PATTERNS AND PROBLEMS:
Analyzing Document Layouts and Improving Mathematical Reasoning of LLMs

by

Mohit Gupta (MT22112)

Under the Supervision of

Dr. Rajiv Ratn Shah

Submitted
in partial fulfilment of the requirements for the degree of
Master of Technology

to

Department of
Computer Science and Engineering

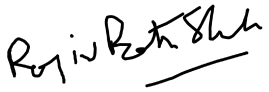
Indraprastha Institute of Information Technology Delhi
NEW DELHI – 110020

May 8, 2024

Certificate

This is to certify that the thesis titled "**PATTERNS and PROBLEMS: Analyzing Document Layouts and Improving Mathematical Reasoning of LLMs**", submitted by Mohit Gupta to the Indraprastha Institute of Information Technology, Delhi for the award of the Master's of Technology, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or whole to any other university or institution for the award of any degree/diploma.



May 8, 2024

Dr. Rajiv Ratn Shah
Associate Professor, Institute Chair Professor
Department of Computer Science & HCD
Indraprastha Institute of Information Technology Delhi
New Delhi, 110020

Acknowledgement

I thank my supervisor **Dr. Rajiv Ratn Shah** for his guidance during the last two years. I barely had any knowledge about the domain when I started, but his guidance, support and patience eased the process for me. Prof. Rajiv has been a great supervisor I could have hoped to have, and working under him taught me a lot about the research field and helped me improve professionally. He also allowed and gave me the opportunity to learn the art of academic research from a research internship at NII, Japan, for which I am deeply grateful for being able to attend. I'd also like to express my gratitude towards **Avinash Anand**, my PhD advisor, and helped me alot in completing and shaping the direction and outcome of my thesis.

I express my gratitude to IIIT Delhi for exposing me to many new experiences, it helped me learn how to learn inside and outside the classroom. It taught me how to be dynamic, to observe and question. Most importantly, it taught me how to efficiently manage my time between work and personal life.

Above all, I am eternally thankful to my family, who supported me during this time. It wouldn't have been possible without them. Handling IIIT Delhi's course work, and thesis was very difficult for me, especially in the first year, when I was adapting to the rigorous schedule of this new environment. They listened to my problems, discussed solutions and guided me on how to manage my workload; by prioritizing the essential things and doing as much as possible in the rest.

Lastly, I truly appreciate my friends for always being there with me in this journey. It couldn't have been fun without them. They were there, when I got stuck in an issue, and helped me resolve the issue. They supported me emotionally, when I needed it the most and helped me get back on track.

The completion of this thesis is a testament to the immense support I have received from my supervisor, lab-mates, friends and family.

Abstract

Documents play a pivotal role in conveying information, serving as integral carriers of knowledge across various domains. Their importance lies in their ability to encapsulate ideas, facts, and insights, thereby facilitating communication and record-keeping. Document analysis, as a field, involves the systematic examination of documents to extract meaningful insights, patterns, or information. While current document analysis tools have made strides, they face challenges such as limited accuracy, scalability issues, and struggles in handling diverse document formats. This highlights the pressing need for more robust and advanced document analysis tools.

This research work attempts to address some of the broad domain difficulties associated with documents and their analysis. This research explains the solutions in the domain of domain adaptation based document layout detection, Detecting and Recognizing Tables within document images.

Secondly, the Large Language Models (LLMs) proved to achieve state-of-the-art results on extensive, and complex NLP related tasks. But sometimes LLMs fails to solve basic mathematical reasoning tasks. Focusing on this, I've worked on proposing a extensive mathematical dataset for training LLMs to enhance their mathematical reasoning capabilities and proposed a efficient approach for solving physics problems using Reinforcement Learning with Human & AI feedback.

As this works focuses on ***patterns*** in the documents and the ***problems*** in the LLMs. That is why the title of this thesis is -

"***PATTERNS and PROBLEMS: Analyzing Document Layouts and Improving Mathematical Reasoning of LLMs***".

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the world of knowledge exchange and data preservation, documents serve as fundamental vessels, whether they manifest as physical pages or digital files. A document, at its core, is designed to communicate thoughts, data, and insights from one entity to another, playing a crucial role in daily communications, professional operations, and even historical record-keeping. The enduring relevance of documents in an increasingly digital landscape highlights their importance across various sectors and disciplines.

The systematic examination of these documents, known as document analysis, employs a range of methodologies to decode the textual and visual information they contain. This field is pivotal for extracting usable data from a sea of information. It supports a wide array of applications 1.1—from processing financial transactions and managing patient records in healthcare to aiding historians and legal professionals in interpreting complex archives. Through document analysis, static content is transformed into actionable insights, making it an invaluable tool in the modern data-driven environment.

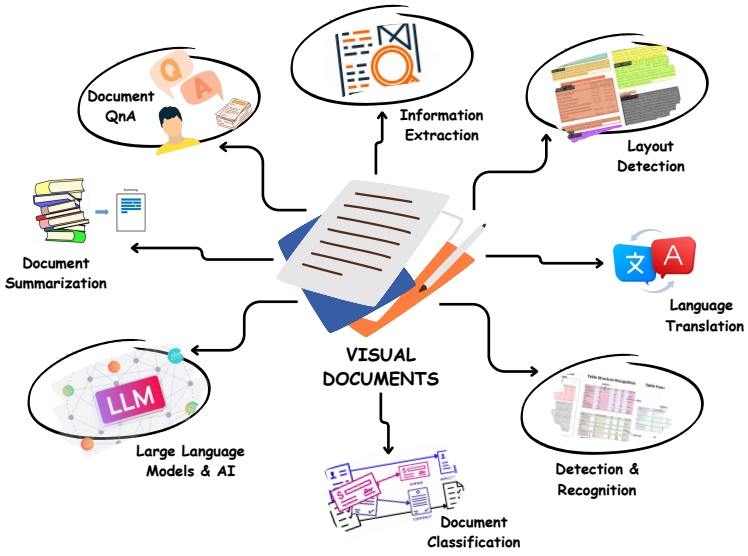


Figure 1.1: Use-Cases of Document Analysis

Despite its extensive applications, the field of document analysis faces significant challenges. Traditional tools often struggle with accuracy when dealing with complex document layouts or diverse formats, such as documents that combine text with graphical elements like tables and images. These tools also frequently lack the scalability needed to process large volumes of data efficiently, hindering their effectiveness and reliability.

Addressing these challenges, this research work introduces advanced methods for improving document layout detection and the recognition of tables within document images. By incorporating techniques from domain adaptation, the research aims to enhance the precision and adaptability of document analysis tools, ensuring robust performance across various document types.

Parallel to the advancements in document analysis, Large Language Models (LLMs) have revolutionized the field of Natural Language Processing (NLP). Models such as GPT and BERT, trained on vast amounts of text, excel in tasks that involve understanding and generating human-like text. They have set new benchmarks in translation, summarization, and sentiment analysis,

demonstrating a deep understanding of language nuances and context.

However, these models often fall short when tasked with mathematical reasoning. Despite their linguistic prowess, LLMs struggle with basic arithmetic and complex problem-solving that requires structured thinking. This gap in capability reveals a crucial limitation in their training and overall function, particularly in scenarios that demand high levels of precision and logical reasoning.

To bridge this gap, this work also proposes the development of a specialized mathematical dataset to train LLMs in mathematical reasoning, complemented by a novel approach to problem-solving using reinforcement learning. This approach, which integrates human and AI feedback, particularly focuses on solving physics problems, aiming to simulate real-world learning processes and enhance the models' application of mathematical concepts in practical situations.

Through these endeavors, this research seeks not only to address and overcome the limitations of current technologies in both document analysis and NLP but also to pave the way for more sophisticated and capable systems. These systems are expected to meet the complex demands of our information-driven world more effectively, marking significant progress in both fields.

Key Contributions:

1. Introduction of the ***RanLayNet*** dataset for Domain Adaptation, aiding in training layout detection models adaptable to different domains [7].

2. Development of ***TC-OCR*** for efficient Table Detection and Recognition of Table Structure & Content [6].

3. Proposing the ***MathQuest*** dataset aimed at improving the problem-solving capabilities of Large Language Models in mathematics, with fine-tuning of LLMs (LLaMA-2 [64], WizardMath [45], MAmmoTH [78]) and achieving strong benchmark results [5].

4. Proposing a approach called ***RLHAIF*** for efficient physics problems solving using LLMs with the help of Reinforcement Learning with Human and AI feedback.
   (Submitted in ACMMM '24)

Chapter 2

Background

> "***The theory, hypothesis, framework or background knowledge held by an investigator can strongly influence what is observed.***"
> ∼ *Norwood Russell Hanson*

In this section, We will be discussing the base concepts, the foundations of the topics related to OCR, Information Extraction, Domain Adaptation, Transformers, and Large Language models. This section will help readers to understand the basic concepts which I have used in this research work to get familiar with the technical stuff.

2.1 Table Detection & Recognition

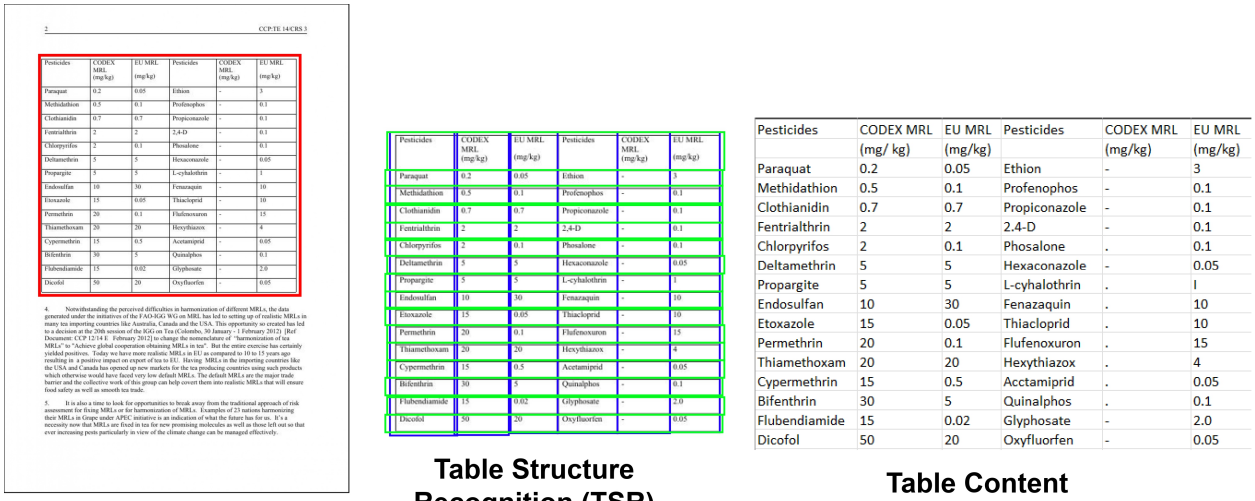


Table Detection (TD)

Table Structure Recognition (TSR)

Pesticides	CODEX MRL (mg/ kg)	EU MRL (mg/kg)	Pesticides	CODEX MRL (mg/kg)	EU MRL (mg/kg)
Paraquat	0.2	0.05	Ethion	-	3
Methidathion	0.5	0.1	Profenophos	-	0.1
Clothianidin	0.7	0.7	Propiconazole	-	0.1
Fentrialthrin	2	2	2.4-D	-	0.1
Chlorpyrifos	2	0.1	Phosalone	.	0.1
Deltamethrin	5	5	Hexaconazole	-	0.05
Propargite	5	5	L-cyhalothrin	.	I
Endosulfan	10	30	Fenazaquin	.	10
Etoxazole	15	0.05	Thiacloprid	.	10
Permethrin	20	0.1	Flufenoxuron	.	15
Thiamethoxam	20	20	Hexythiazox	.	4
Cypermethrin	15	0.5	Acctamiprid	.	0.05
Bifenthrin	30	5	Quinalphos	.	0.1
Flubendiamide	15	0.02	Glyphosate	-	2.0
Dicofol	50	20	Oxyfluorfen	-	0.05

Table Content Recognition (TCR)

Figure 2.1: Table Detection & Text Recognition from Document Image [6]

Table Detection & Recognition is a task of detect and extract information from tables within the images of the documents. It is an important application of OCR, which is a method for recognizing text in visuals of scanned documents and photographs. OCR consists of two main steps: **Text Detection** and **Text Recognition**. Text detection refers to the process of identifying and locating regions that contain text in an image, and text recognition is the process of translating the text in the image into editable, searchable, and storable text.

2.1.1 Text Detection (Bounding Box Detection)

One common method for text detection is bounding box detection, which is a technique to draw rectangular boxes around the text regions in the image. It is a crucial component in OCR systems. It involves identifying the regions of interest (ROI) within an image that contain text. These regions are represented as rectangular bounding boxes that enclose the text elements. Bounding box detection is essential for isolating text from other visual elements in an image, such as graphics, tables, or background noise.

Accurate bounding box detection is achieved through advanced computer vision techniques, including deep learning models. Research papers often focus on improving the precision and recall of bounding box detection algorithms to ensure that text regions are accurately identified. Some examples of bounding box detection algorithms are DBNet [16], EAST [86], and CRAFT [8].

2.1.2 Text Recognition

After detecting the bounding boxes, the next step is to recognize the text inside them. This can be done by using algorithms that can handle various challenges, such as different languages, fonts, sizes, orientations, and noises. Some examples of text recognition algorithms are PP-OCR [18], Tesseract [58], and CRNN [73].

In recent research, PP-OCR [18] has demonstrated advancements in text recognition accuracy. By integrating deep learning models, it achieves SOTA results in recognizing text from bounding boxes, enhancing the overall performance of OCR systems.

These OCR techniques, along with bounding box detection and text recognition algorithms, play a pivotal role in automating the extraction of structured information, particularly in tasks like table detection and recognition within documents. Advances in these technologies contribute to improved document understanding and data extraction capabilities.

2.2 Layout Detection & Domain Adaptation

2.2.1 Layout Detection

Layout detection stands as a pivotal component in the domain of document analysis and comprehension, addressing the intricate challenge of deciphering the structure and arrangement of information within diverse documents. This process involves the identification and categorization of different elements such as text, images, tables, and other visual components within a document, providing a foundation for subsequent data extraction and comprehension.

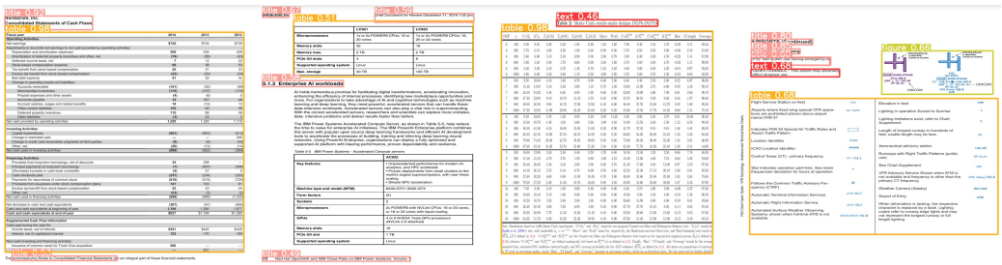


Figure 2.2: Example showing Layout Detection on document Images

The significance of layout detection lies in its role as a precursor to effective information extraction from documents. It is extensively utilized in various domains where structured data is embedded within unstructured documents. In academia and research, layout detection finds applications in digital libraries, archives, and information retrieval systems, enabling efficient categorization and retrieval of relevant content.

Advanced techniques, particularly those rooted in deep learning, have emerged as robust solutions to the challenges of layout detection. Research papers such as LayoutLM [72], Layout-DETR [76] have pioneered the integration of textual and layout information, leveraging pre-trained language models in achieving SOTA performance in document understanding tasks. These models not only discern the textual content but also capture the spatial relationships and hierarchical structures present in documents.

2.2.2 Domain Adaptation

The complexities associated with layout detection arise from the diverse array of document formats, styles, and languages within various domains. Each domain presents distinct challenges, with documents featuring unique layouts characterized by differences in column structures, font sizes, and spatial arrangements. Notably, the structure of financial and medical documents significantly differs from that of research papers or academic documents. This divergence encapsulates what is commonly known as the problem of Domain Shift. Traditional rule-based approaches encounter limitations when confronted with the intricate nature of domain shift, unable to effectively address the complexity and variability inherent in real-world documents.

Domain shift refers to phenomenon where the statistical properties of the original domain, where a model undergoes training, differ significantly from the objective domain, where the trained model is applied. Domain shift arises when the characteristics of the documents used for training differ from those encountered in real-world scenarios or diverse applications. Domain

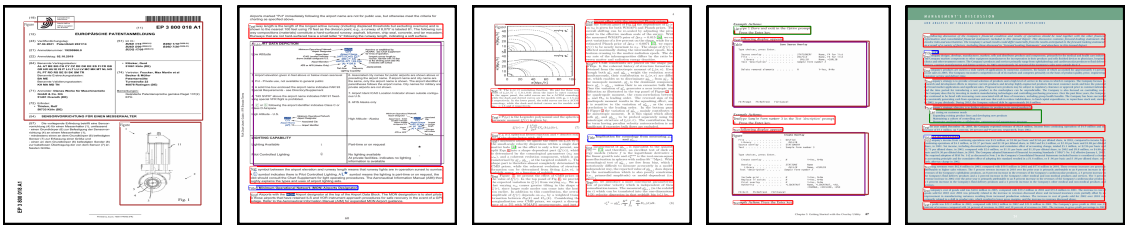


Figure 2.3: Document Images from different domains showing domain shift

Adaptation [19, 43] aims to mitigate the adverse effects of domain shift, enhancing the model's robustness and generalization capabilities across diverse document domains. Domain adaptation techniques are employed to ensure that models trained on a source domain can effectively adapt to the nuances of target domains. Recent research has explored innovative approaches to domain adaptation in document layout detection. Methods like self-training, where the model labels additional unlabeled data during adaptation, and adversarial training, which encourages the model to disregard domain-specific characteristics, have shown promise. Domain adaptation methods are crucial for improving the flexibility and effectiveness of models in various real-world document scenarios.

2.3 Document Parsing

Image-based Text Extraction (OCR)	**Parsing Tools (PyPDF, PDFMiner)**
OCR may struggle with complex document layouts, intricate fonts, or unconventional text arrangements.	Parsing tools heavily relies on the structure of PDF document. If document has a complex layout, non-standard encoding, or uses custom fonts, these tools may not accurately extract text.
OCR systems may struggle with handwritten text or fonts that deviate significantly from standard typefaces.	Maintaining formatting, such as indentation, font styles, or table structures, can be challenging for parsing tools.
Poor-quality images, low resolution, or images with noise can degrade OCR accuracy.	Maintaining formatting, such as indentation, font styles, or table structures, can be challenging for parsing tools.

Table 2.1: Challenges in Image-based OCR tools & PDF Parsing tools

2.3.1 Pdf-to-Text Conversion & Information Extraction

PDF to text converters are tools or software applications designed to convert Portable Document Format files into plain text files. These converters play a crucial role in extracting textual content from PDF documents, making it more accessible and editable. They are utilized in various scenarios, including document management, text analysis, and content extraction.

The process of converting PDF to text typically involves the following steps:

- **Parsing the PDF File**: The converter first parses the PDF file, interpreting its structure and layout. PDFs can contain a mix of text, images, and other elements.
- **Text Extraction**: The converter identifies and extracts text content from the PDF. This includes text from paragraphs, headings, lists, and other textual elements.
- **Maintaining Formatting**: Some converters attempt to preserve the formatting of the original document, such as line breaks, font styles, and indentation. This helps in retaining the document's structure and readability.
- **Handling Images and Graphics**: In cases where the PDF contains images with embedded text, OCR (Optical Character Recognition) may be employed to recognize and extract text from these images.
- **Output as Plain Text**: The final result is a plain text document that mirrors the content of the original PDF. This text file can be easily edited, searched, or used for further analysis.

PDF to text converters find applications in scenarios where access to the raw text content of a PDF document is essential. This includes text mining, content indexing, data extraction for analysis, and making PDF content compatible with systems that require plain text input. Popular PDF to text python-based conversion tools include libraries such as PyPDF2[1], PDFMiner[2], Textract[3].

Model	CPU Time (sec.)	System Time (ms)	Total Time (sec.)
EasyOCR	7.17	454	7.62
pyPDF	0.129	1.87	0.131

Table 2.2: Processing time for text extraction from a 2-page document using EasyOCR and pyPDF.

Text extraction from images using OCR is a process of converting images of text into editable text files. Text extraction using parsing tools like pyPDF or pdfMiner is a process of studying the internal structure and metadata of PDF files and extracting the text elements. Both methods have some advantages and disadvantages. Addressing the challenges discussed in Table 2.1 of image-based text extraction using OCR and parsing tools often requires a combination of OCR technologies with advanced algorithms, ML models, or DL approaches. These methods aim to enhance the robustness of text extraction across various document types, layouts, and languages. Additionally, the development of tools that integrate OCR with parsing capabilities can provide more comprehensive solutions for accurate text extraction from diverse documents.

[1] `https://github.com/py-pdf/pypdf`
[2] `https://github.com/pdfminer/pdfminer.six`
[3] `https://github.com/deanmalmgren/textract`

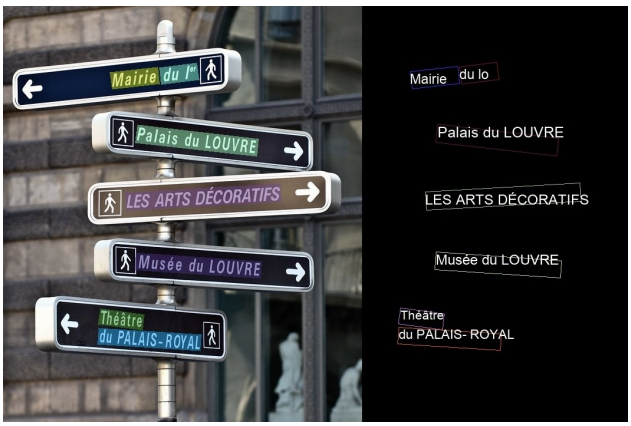


Figure 2.4: Optical Character Recognition from Images

2.4 Large Language Models

Before discussing Large Language Models, we first need to understand the birth of transformers. Natural Language Processing (NLP) tasks predominantly relied on traditional approaches such as rule-based systems, statistical models, and early neural network architectures. Rule-based systems [47, 35, 61, 25] often involved handcrafted linguistic rules, which were limited in their ability to capture the complexity and variability of natural language. Statistical models, including Hidden Markov Models (HMMs) [32, 60] and Conditional Random Fields (CRFs) [28, 12], focused on probabilistic relationships between words but struggled with capturing long-range dependencies and contextual nuances. Early neural network architectures, like feedforward neural networks, faced challenges in handling sequential data and maintaining context across sentences. One

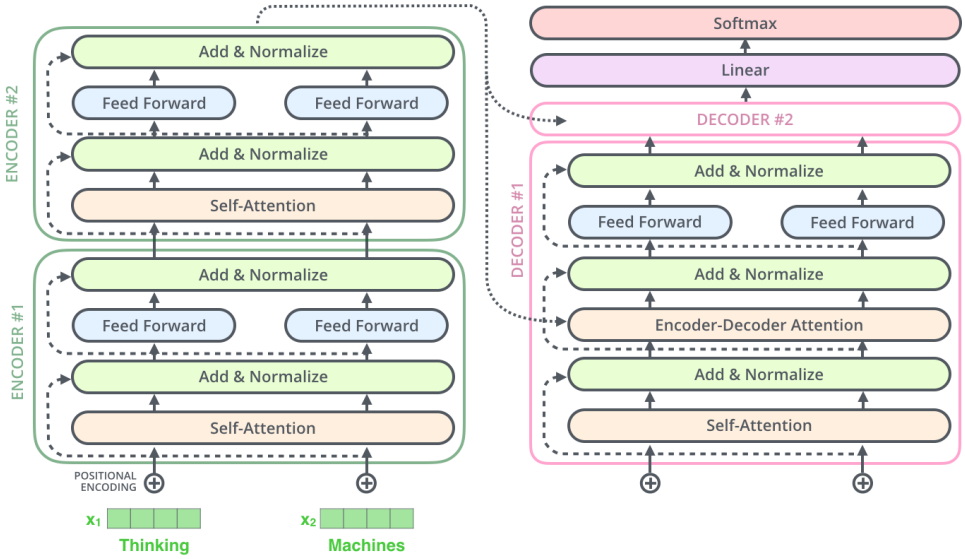


Figure 2.5: Transformer Architecture [2]

notable approach was the use of recurrent neural networks (RNNs) [75], which attempted to address the sequential nature of language by introducing hidden states that could capture contextual information. However, RNNs suffered from vanishing and exploding gradient problems, limiting their effectiveness in modeling long-range dependencies. These challenges led to difficulties in capturing intricate language structures and hindered the overall performance of NLP systems.

The limitations of pre-transformer approaches became more apparent as NLP tasks grew in

complexity, demanding a better understanding of context, semantics, and hierarchical relationships within textual data.

2.4.1 Transformers

Transformers addressed the shortcomings of earlier approaches, leading to significant advancements in the field of NLP. The Transformer architecture was introduced in 2017 by Vaswani et al. in their paper "Attention Is All You Need" [65]. Transformers have revolutionized the field of NLP by introducing attention mechanisms, enabling the efficient processing of long-range dependencies and contextual information, pushing the state-of-the-art for a number of tasks, such as machine translation, text summarization, question answering, and text generation.

Several variants and extensions of the transformer architecture have been proposed, such as BERT [17], GPT [10], BART [36], and T5 [56], which leverage large-scale pre-training and fine-tuning to achieve impressive results on various NLP benchmarks.

Characteristics	**Language Model**	**Large Language Model**
Architecture	Traditional LMs like n-gram models or earlier neural network-based models (e.g RNNs, LSTMs) have simpler architectures. They often focus on capturing local context and may struggle with long-term dependencies.	LLMs, such as GPT series, BERT, and T5 are based on the transformer architecture. This architecture allows them to effectively capture long-range dependencies and understand context over larger spans of text.
Number of Parameters	Traditional LMs have significantly fewer parameters. For instance, earlier RNN-based models might have parameters in the range of millions. The smaller size limits their ability to capture and generalize complex language patterns.	LLMs are characterized by their massive number of parameters. For example, GPT-3 has 175 billion parameters, and GPT-4 goes even beyond that. The large number of parameters allows these models to capture a wide range of language nuances.
Training Process	Traditional LMs are trained on smaller datasets due to their limited capacity. They often require careful feature engineering and domain-specific tailoring.	LLMs are trained on extensive and diverse datasets, often encompassing a large portion of the publicly available internet text. They employ techniques like unsupervised learning, self-supervision, or transfer learning, enabling them to learn a broad range of language patterns.
Model Examples	Hidden Markov Models, or Neural Network models like LSTMs, Bi-LSTMs etc.	OpenAI's GPT series (GPT-2, GPT-3, GPT-4), LLaMA, BERT, and T5 models

Table 2.3: Characteristics Difference between LMs and LLMs

Attention Mechanism

Self-attention, also known as intra-attention, is an attention mechanism that establishes connections between different positions within a single sequence to generate a representation of the sequence. It has proven successful in various tasks such as reading comprehension, abstractive summarization, textual entailment, and learning task-independent sentence representations. Essentially, self-attention is the technique employed by the Transformer to incorporate the "understanding" of other relevant words into the one currently being processed.

The attention function can be defined as a process that maps a query and a set of key-value pairs to an output, where the query (A), keys (B), values, and output are all vectors. The output is computed as a weighted sum of the values, with the weight assigned to each value determined by a compatibility function between the query and the relevant key.

$$Attention\ (A, B, V) = softmax\left(\frac{AB^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

LMs and LLMs

The key difference lies in the scale, which impacts their abilities. Large Language Models, with their vast number of parameters and advanced architectures, are more adept at understanding and generating human-like text, capturing nuances, and performing a wide range of NLP tasks with minimal task-specific tuning. In contrast, traditional Language Models are more constrained in their capabilities but are less resource-intensive and can be more transparent and interpretable. The detailed difference between the characteristics of LMs and LLMs are shown in Table 2.3.

Chapter 3

Literature Review

"***Discoveries are made by pursuing possibilities suggested by existing knowledge***"
~ Michael Polanyi

3.1 Domain Adaptation & Layout Detection

In recent years, tremendous progress has been achieved in the domain adaptation sector [33], tackling the complexities of transferring knowledge from a source domain with distinct data distribution to a target domain. A particularly notable advancement is presented in the study by Zhang et al. [82], where they proposed a progressive self-training framework designed for unsupervised domain adaptation. This method systematically uses both labelled data from the original domain and unlabeled information from the desired domain, refining the model's predictions and integrating samples from the target domain through iterative steps.

Initial approaches to document layout analysis depended on rule-based systems and heuristic approaches [1]. However, there has been a paradigm shift towards addressing these challenges through the adoption of deep learning techniques. This involves leveraging object detection models [23, 22, 57, 15, 62], which have shown significant improvements in both accuracy and speed over the last decade. Importantly, contemporary object detection methods are user-friendly, allowing for easy training and implementation with minimal effort, Standardised ground-truth formats for data [42] and popular deep learning frameworks [70]. Reference datasets like PubLayNet [84] and DocBank [39] conveniently provide their data in the widely accepted COCO format [42].

3.2 Table Detection & Text Recognition

Recognizing table structures has emerged as a complex and unresolved challenge in the document-analysis community. Addressing this difficulty, several open challenges have been organized [20, 24, 31]. The intricacy of this issue can be attributed to various elements.

Initially, tables present a diverse array of shapes and sizes, requiring an adaptable strategy to properly manage their fluctuation. This becomes especially important when confronted with intricate and demanding features of complex column and row headers. Secondly, a significant challenge emerges due to restricted availability of data specifically designed for table structure analysis. Nevertheless, substantial progress has been made in recent years, addressing the data deficiency issue with the introduction of valuable datasets such as PubTabNet [83], FinTabNet [81], and TableBank [38].

Nguyen et al. [51] introduced TableSegNet, a compact convolutional network that can perform table separation and detection simultaneously. Zhang et al. [79] presented a YOLO-based table detection method, enhancing spatial arrangement learning involves improving efficiency by incorporating an involution into the network's core & utilizing a basic feature pyramid network.

In the initial phases of table structure recognition, methods strongly relied on hand-crafted features and heuristic rules [29, 68]. These approaches were effective in handling basic table structures or established data types. However, recent advancements, driven by the substantial success of deep

learning in various computer vision tasks like object detection and semantic segmentation, have given rise to several innovative deep learning-based methods for table structure recognition.

TableMASTER, as introduced by [26, 40], represents a Transformer-based model meticulously crafted for the purpose of table structure recognition. This approach ingeniously merges the Transformer model with a text line detector to discern text lines within each table cell. Additionally, the method incorporates a text line recognizer, drawing inspiration from the work of [44], extract textual material from the identified lines.

Zhang et al. [80] put forward an end-to-end deep learning model designed for text reading and information extraction within document understanding. Their approach involves a hierarchical architecture, comprising a feature extraction module, a text reading module, and an information extraction module. The authors rigorously evaluate their model using diverse public datasets. They utilize SpaCy, for example, to extract private information from unstructured or loosely organized data.

3.3 Educational AI & LLMs

Recent research has emphasized the potential of Large Language Models in the field of education. They show promise in automating question generation and facilitating direct interactions within the learning environment [30]. Additionally, investigations have delved into few-shot prompting techniques applied to LLMs, specifically for addressing mathematical word problems [69, 85, 21]. The "chain-of-thought" prompting approach [69] strategically employs explicit intermediate reasoning steps to enhance the LLM's overall reasoning abilities. In response to arithmetic errors often observed in LLMs [37, 27], prior studies [14] have explored the use of external calculators to execute operations generated by LLMs.

Several approaches aim to enhance mathematical problem-solving using Large Language Models (LLMs). Wang et al.'s self-consistency method [67], based on the CoT framework, evaluates multiple potential reasoning paths and selects answers through a plurality of votes. Li et al. [41] extend self-consistency by instructing a verifier to validate each step, while Madaan et al. [46] leverage recent LLMs like GPT-3.5 to generate an output, provide feedback, and prompt the model for improvements. Wang et al. [66] assess pretrained language models on basic arithmetic expressions, including addition $(+)$ and subtraction $(-)$, and Muffo et al. [49] expand the evaluation to include multiplication $(*)$ operations within the language model's scope.

It's crucial to recognize that, despite the effectiveness of Large Language Models (LLMs), approaches such as CoT, ToT [69, 74] and knowledge graphs [53] may still encounter limitations in delivering responses that mimic human-like understanding to complex problems [9].

The constraints associated with the aforementioned approaches have led to the emergence of a new method called Reinforcement Learning from Human Feedback (RLHF). This method aims to improve language models by aligning their outputs more closely with human preferences [71]. Initially, RLHF was utilized to enhance language models for specific tasks such as text summarization [59] and question answering [50]. Over time, RLHF techniques have gained widespread acceptance for more versatile, general-purpose language models.

Chapter 4

Contributions & Evaluations

"***Research is formalized curiosity.***"
∼ Zora Neale Hurston

In this section, I'll delve into the research contributions and publications I've been involved in during this period. Let's start with the **"PATTERNS"** part of the thesis, which encompasses two published research works. The first paper tackles Document Layout Detection, proposing a diverse noisy dataset. This dataset aids in training a model to detect various layout labels like text, headings, figures, tables, and charts in documents, along with providing bounding box annotations. Published in **ACM MMAsia '23**, this work significantly enhances the robustness of document layout detection performance, facilitating seamless adaptation to different domains. More details on this work are covered in Section 4.1.

While the aforementioned work lays the groundwork for understanding document structure, merely detecting it is not sufficient. To maximize its utility, a subsequent research project, published in **ACM MMIR '23**, addressed this gap. Consider scientific documents containing images, tables, graphs, and text. Language models efficiently reason through text, and vision-based models handle images. However, tables present a unique challenge due to their diverse structures—some have borders, others don't; some have merged rows and columns, while others contain missing values. Extracting data from tables without any loss of information or structure is a daunting task. Previous approaches often focused on singular aspects like Table Detection, Recognition, or Text Extraction. However, there lacked a unified pipeline capable of handling all these tasks within a single framework. Building on this, we introduced TC-OCR, a comprehensive framework for detecting, recognizing table structures, and extracting content from document images, saved in CSV format. This pioneering work outperforms previous state-of-the-art models like Table Transformer and effectively analyzes table structures within documents. Refer to Section 4.2 for detailed insights into the results.

In the **"PATTERNS"** segment of the thesis, we introduced the RanLayNet dataset aimed at enhancing robust document layout detection by addressing domain shift challenges. Additionally, we presented a unified pipeline for detecting and recognizing table content from document images. This innovative approach surpasses previous benchmark results for this task, paving the way for further research in the realm of table understanding and reasoning.

Transitioning to the **"PROBLEMS"** segment of the thesis, we shift our focus slightly to a different challenge. Large Language Models (LLMs) have gained prominence in the NLP domain for their exceptional long-sequence understanding and reasoning capabilities. While LLMs excel in general tasks, their performance in solving basic mathematical problems falls short. As LLMs become integral to smart education, bridging this gap becomes imperative. The **"PROBLEMS"** part of the thesis aims to enhance the mathematical reasoning abilities of open-source LLMs. Introducing MathQuest published in **NeurIPS' 23**, an Indian High-School Mathematics Dataset, this research initiative seeks to provide a comprehensive collection of math problems covering various topics and difficulty levels. By proposing this dataset alongside benchmark results utilizing the top-performing LLM, **MAmmoTH-13B**, we pave the way for further advancements in mathematical problem-solving. For a detailed discussion, refer to Section 4.3.

As it is said that:

"***In the realm of problem-solving, mathematics is the trusted compass, steering us through the vast terrain of physics.***"
$\sim$ *ChatGPT*

In physics problem-solving, a comprehensive understanding of context and precise arithmetic operations is essential, posing a challenge for conventional Large Language Models (LLMs). Our focus is enhancing LLMs to tackle physics-related questions extracted from the PhyQA [3, 4] dataset, which encompasses problem sets from Indian NCERT textbooks for 11th and 12th grades. We employ a methodology called Reinforcement Learning with Human Feedback (RLHF) to boost the efficacy and accuracy of LLMs in arithmetic reasoning. Various reinforcement learning techniques, including Policy Optimization (PPO), Direct Preference Optimization (DPO), and ReMax optimization, are examined to assess their performance in solving physics problems across diverse scenarios. A pivotal aspect of our approach involves integrating human and artificial intelligence feedback. This innovative strategy aids in training our models to produce more logical and reasonable solutions to physics problems. This work is submitted for review in **ACMMM '24**. A more detailed discussion is mentioned in Section 4.4.

4.1 RanLayNet: A Dataset for Document Layout Detection used for Domain Adaptation and Generalization

Citation: *Avinash Anand, Raj Jaiswal, **Mohit Gupta**, Siddhesh S Bangar, Pijush Bhuyan, Naman Lal, Rajeev Singh, Ritika Jha, Rajiv Ratn Shah, and Shin'Ichi Satoh. 2024. RanLayNet: A Dataset for Document Layout Detection used for Domain Adaptation and Generalization. In Proceedings of the 5th ACM International Conference on Multimedia in Asia (MMAsia '23). Association for Computing Machinery, New York, NY, USA, Article 74, 1–6.*

4.1.1 Abstract

The availability of large ground-truth datasets and recent advancements in deep learning techniques have proven beneficial for layout detection. However, due to the limited diversity of layouts in these datasets, training models on them necessitates a substantial number of annotated instances, which is costly and time-consuming. Consequently, differences between the source and target domains can significantly impact the effectiveness of these models. To address this issue, domain adaptation methods have been developed to adjust models to the target domain using a small amount of labelled data. In this study, we introduce a synthetic document dataset named RanLayNet, which is enriched with automatically assigned labels indicating spatial positions, ranges, and types of layout elements. Our objective is to create a versatile dataset capable of training models with resilience and adaptability to various document formats. Through empirical experimentation, we demonstrate that a deep layout identification model trained on our dataset outperforms a model trained solely on authentic documents. Additionally, we conduct a comparative analysis by fine-tuning inference models using both the PubLayNet and IIIT-AR-13K datasets on the Doclaynet dataset. Our results underscore that models enriched with our dataset excel in tasks such as achieving 0.398 and 0.588 mAP95 scores in the scientific document domain for the TABLE class.

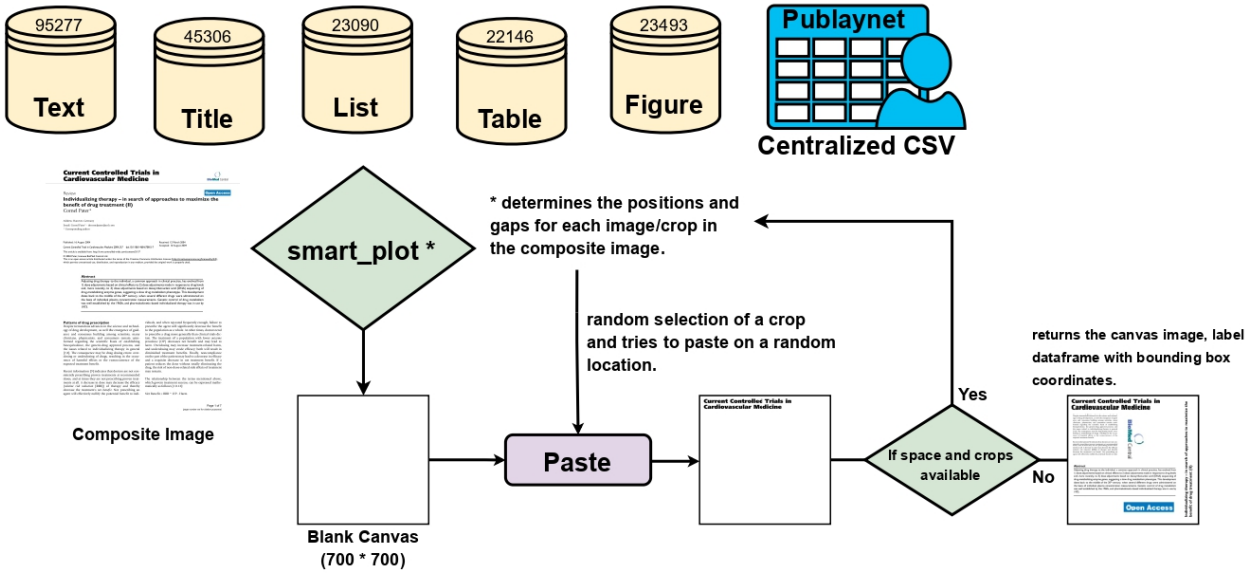


Figure 4.1: RanLayNet generation workflow

4.1.2 Methodology

Dataset

In this paper, we introduce the RanLayNet dataset. RanLayNet contributes to the field by introducing greater variability in document layouts, exceeding the scope of existing datasets such as Publaynet [84], Doclaynet [54], and IIIT-AR-13K [48]. It specifically presents complex structures, covering a wide range of layout classes to offer a balanced representation of document layout components. This diverse set equips models to handle unpredictable real-world layouts. Deep learning models trained on the The RanLayNet dataset provides insights into the spatial positions, categories of elements, and extents within document layouts. Exposure to such diverse layouts assists models in acclimating to real-world variations. The dataset's generality enables the training of models for a wide range of layouts, effectively addressing domain shift issues.

Label	Count	Percentage (%)
Text	95,227	45.52
Title	45,306	21.65
List	23,090	11.03
Table	22,146	10.58
Figure	23,493	11.22

Table 4.1: Distribution of different class labels in RanLayNet

The diverse structures within RanLayNet serve as a defense against overfitting, promoting adaptable representations that mitigate domain disparities. Models trained on RanLayNet demonstrate superior performance compared to those trained on PublayNet, highlighting robustness and adaptability to various layouts, thus reinforcing domain adaptation. The count and distribution of class labels in RanLayNet are presented in Table. 4.1.

This work aims to construct a dataset conducive to developing robust and versatile models. A model trained on this dataset can process data in diverse formats efficiently. Our approach commences by utilizing the source dataset Publaynet [84], which includes fixed labels such as Text, Title, List, Figure, and Table. This dataset is employed for the initial training of our model. Following this, a target dataset is carefully curated, featuring an expanded class structure and diverse domains. The primary challenge in domain adaptation is mitigating the disparities between the source and target datasets. An essential aspect of this method involves systematically expanding the RanLayNet dataset. We meticulously incorporate diverse layout structures and distinct data distributions as patches onto the dataset canvas. This augmentation enhances the dataset's complexity and variance, exposing the model to a larger spectrum of scenarios.

RanLayNet Generation: Generating the RanLayNet dataset, comprising packed images and related label information entails multiple steps. Initially, a .csv file containing details about the images and their associated bounding boxes is supplied from Publaynet [84]. Utilizing the 'smart_plot' function, positions and gaps are determined to effectively accommodate each image or crop within a composite image. Upon successful position and gap calculation, a blank canvas is generated as the base for the composite image.

Subsequently, the function iterates through each determined position. Based on the provided data, it selectively crops specific regions from the images or pastes entire images onto the composite canvas at the designated positions. Throughout this process, the coordinates of bounding boxes are updated, and these new coordinates are stored in a label file. Iterating through all positions, the

function orchestrates the arrangement and placement of images or crops onto the composite canvas, creating a comprehensive representation.

Ultimately, the output comprises the composite image itself, along with a label file containing adjusted bounding box information. The entire workflow is visually represented in Figure. 4.1.

4.1.3 Results & Analysis

In this study, we conducted fine-tuning of YOLOv8 across various datasets for 40 epochs, utilizing a batch size of 16 and setting num_workers to 4. The optimization was performed using the SGD optimizer with a learning rate of 1e-3. In our context, Noise Labeling[1] is employed for a document layout identification model with the goal of enhancing the model's capability to accurately detect various elements within diverse types of documents spanning different domains. Our experimentation reveals that the "Table" class exhibits the highest mean average precision, indicating its significance and potential impact across various applications.

Domain	**IIIT-AR-13K**				**IIIT-AR-13K + RanLayNet**			
	Precision	**Recall**	**mAP50**	**mAP95**	**Precision**	**Recall**	**mAP50**	**mAP95**
Manuals	0.194	0.0353	0.0587	0.0262	**0.446**	**0.356**	**0.336**	**0.226**
Financial Documents	0.00405	0.013	0.00028	0.00088	**0.374**	**0.334**	**0.298**	**0.187**
Laws and Regulations	0.194	0.353	0.0587	0.0262	**0.619**	**0.263**	**0.293**	**0.222**
Scientific Documents	0.0112	0.0042	0.00582	0.00219	**0.187**	**0.474**	**0.334**	**0.398**

Table 4.2: Inference of the "TABLE" class on Doclaynet Documents with models fine-tuned on IIIT-AR-13K and IIIT-AR-13K with RanLayNet.

Building on this observation, we performed "Table" class detection using two fine-tuned models derived from source datasets: IIIT-AR-13K and IIIT-AR-13K with RanLayNet. The objective was to detect "Tables" in diverse domain documents, including Manuals, Financial Documents, Laws & Regulations, and Scientific Documents. The results, as presented in Table. 4.2 and 4.3, illustrate that fine-tuning the model on the source dataset with our noisy dataset significantly enhances model performance across various metrics in different target domains. This outcome provides additional evidence supporting the effectiveness of our approach in enhancing adaptability and generalization capabilities.

Domain	**PubLayNet**				**PubLayNet + RanLayNet**			
	Precision	**Recall**	**mAP50**	**mAP95**	**Precision**	**Recall**	**mAP50**	**mAP95**
Manuals	0.488	0.582	0.421	0.220	**0.562**	**0.807**	**0.761**	**0.588**
Financial Documents	0.350	0.452	0.288	0.163	**0.427**	**0.555**	**0.465**	**0.293**
Laws and Regulations	0.039	0.356	0.337	0.194	**0.294**	**0.572**	**0.350**	**0.282**
Scientific Documents	0.366	0.621	0.562	0.376	**0.562**	**0.807**	**0.761**	**0.588**

Table 4.3: Inference of the "TABLE" class on Doclaynet Documents with models fine-tuned on PubLayNet and PubLayNet with RanLayNet.

[1]**Noise labeling** refers to the intentional addition of noisy labels to the training data for the purpose of studying the impact of noisy labels on the performance of models. This practice is commonly employed to assess the robustness of machine learning algorithms against various types and levels of label noise.

4.1.4 Paper Conclusion

In summary, our proposed approach highlights the challenges in bridging the gap between source and target datasets with distinct label structures. The model, trained on a source dataset with a specific set of five labels, faced limitations when applied to a target dataset encompassing additional classes. This disparity hindered the model's generalization, prompting the exploration of innovative solutions. In response, we introduced the concept of "RanLayNet," a dynamic and unbiased approach to dataset creation. This noisy dataset, devoid of a predetermined layout structure, exhibited inherent adaptability to diverse layout configurations. This adaptability is crucial for mitigating bias and promoting versatility, enabling the model to navigate the intricate domain landscape effectively. By leveraging the power of RanLayNet and adopting a generality paradigm, our approach has the potential to revolutionize domain adaptation strategies. Through the fusion of innovative methodologies, we aim to empower models with enhanced adaptability, ensuring they transcend the constraints of source-target alignment paradigms.

4.1.5 Future Scope

The current implementation, featuring five labels, demonstrates promising adaptability in the target domain. However, our trajectory envisions a progressive evolution, with plans to introduce diverse patches on our canvas. This expansion is anticipated to facilitate seamless generalization across a broader spectrum of domains, even without explicit label generation for both the source and target datasets.

4.2 TC-OCR: TableCraft OCR for Efficient Detection & Recognition of Table Structure & Content

Citation: *Avinash Anand, Raj Jaiswal, Pijush Bhuyan,* ***Mohit Gupta****, Siddhesh Bangar, Md. Modassir Imam, Rajiv Ratn Shah, and Shin'ichi Satoh. 2023. TC-OCR: TableCraft OCR for Efficient Detection & Recognition of Table Structure & Content. In Proceedings of the 1st International Workshop on Deep Multimodal Learning for Information Retrieval (MMIR '23). Association for Computing Machinery, New York, NY, USA, 11–18.*

4.2.1 Abstract

The automatic recognition of tabular data in document images presents a significant challenge due to the diverse range of table styles and complex structures. Tables offer valuable content representation, enhancing the predictive capabilities of various systems such as search engines and Knowledge Graphs. Addressing the two main problems, namely table detection (TD) and table structure recognition (TSR), has traditionally been approached independently. In this research, we propose an end-to-end pipeline that integrates deep learning models, including DETR, Cascade TabNet, and PP OCR v2, to achieve comprehensive image-based table recognition. This integrated approach effectively handles diverse table styles, complex structures, and image distortions, resulting in improved accuracy and efficiency compared to existing methods like Table Transformer. Our system achieves simultaneous table detection, table structure recognition, and table content recognition (TCR), preserving table structures and accurately extracting tabular data from document images. The integration of multiple models addresses the intricacies of table recognition, making our approach a promising solution for image-based table understanding, data extraction, and information retrieval applications. Our proposed approach achieves an IOU of 0.96 and an OCR Accuracy of 78%, showcasing a remarkable improvement of approximately 25% in the OCR Accuracy compared to the previous Table Transformer approach.

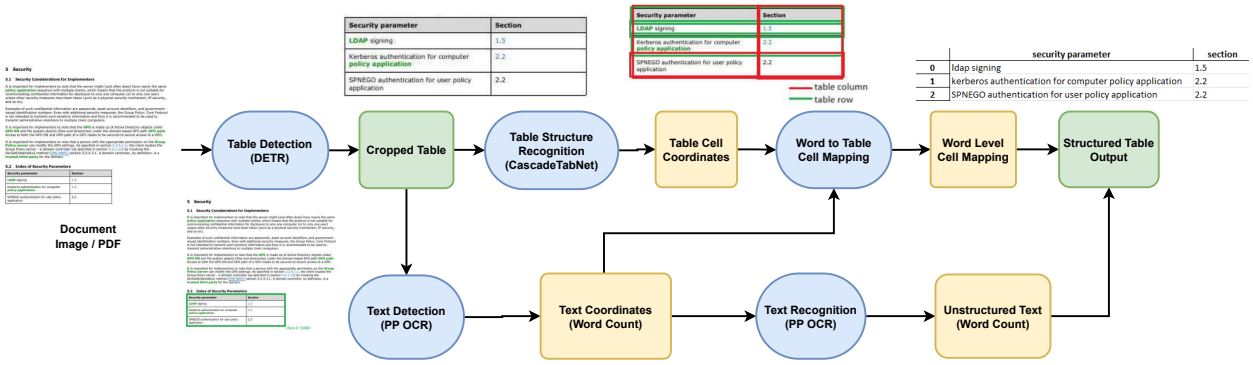


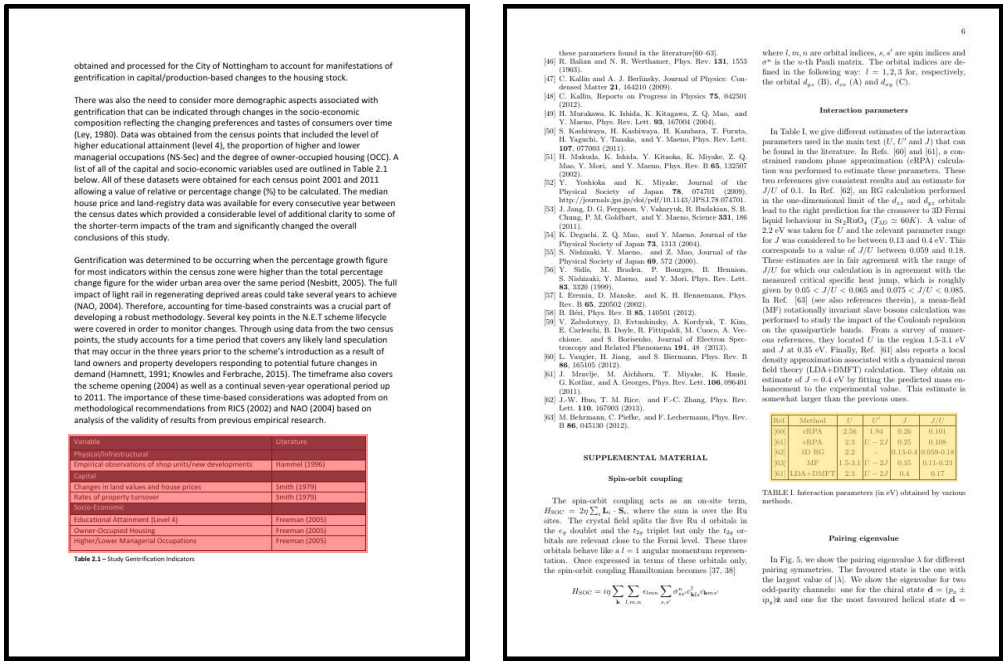
Figure 4.2: Overall Architecture of TC-OCR

4.2.2 Methodology

Dataset

In this research, we utilized the TableBank [38] dataset, an extensive standardized open-domain table benchmark dataset, to fulfill the requirement for large-scale table analysis across various domains. This dataset outnumbers previous human-labeled datasets in terms of size, comprising

417,234 tables, each associated with its original document. The dataset is generated by manipulating markup tags for tables found in electronic documents, such as Word (.docx) and LaTeX(.tex) files. To offer high-quality labelled data, bounding boxes are added with the markup language. Figure 4.3 depicts instances of data samples from TableBank from two distinct sources.



Word Tables (.docx) **LateX Tables (.tex)**

Figure 4.3: Different document images highlighting Tables from two different sources.

In this work, we present an extensive pipeline that combines three unique models to meet the many issues associated with various table designs, complicated structures, and visual distortions often found in document graphics.

1. **DETR - Object Detection Model:** Nicolas Carion et al. [11] introduced was a methodology for addressing object detection by framing it as a direct set prediction problem. The approach utilizes an encoder-decoder architecture based on transformers, known for their efficacy in sequence prediction tasks. This is harnessed as an end-to-end, transformer-based solution for object detection, directly generating sets of bounding boxes and class labels. This approach ensures clear and distinct predictions, effectively handling challenges associated with duplicate predictions.

2. **CascadeTabNet:** CascadeTabNet [55], introduced is an advanced end-to-end deep learning framework that adeptly addresses both table recognition sub-problems through a unified model. This method achieves pixel-level table segmentation, precisely identifying each table instance inside an input image. Furthermore, it conducts table cell segmentation, predicting segmented regions corresponding to individual cells, thereby facilitating the extraction of the table's structural information.

   Figure 4.4 illustrates the architecture of CascadeTabNet. The primary components in its design include Cascade RCNN, a multi-stage model tailored to tackle the challenges of high-quality object detection in CNNs. Also, a customised version of HRNet is integrated, offering dependable high-resolution representations and multi-level features, which prove advantageous for the semantic segmentation tasks associated with table recognition.

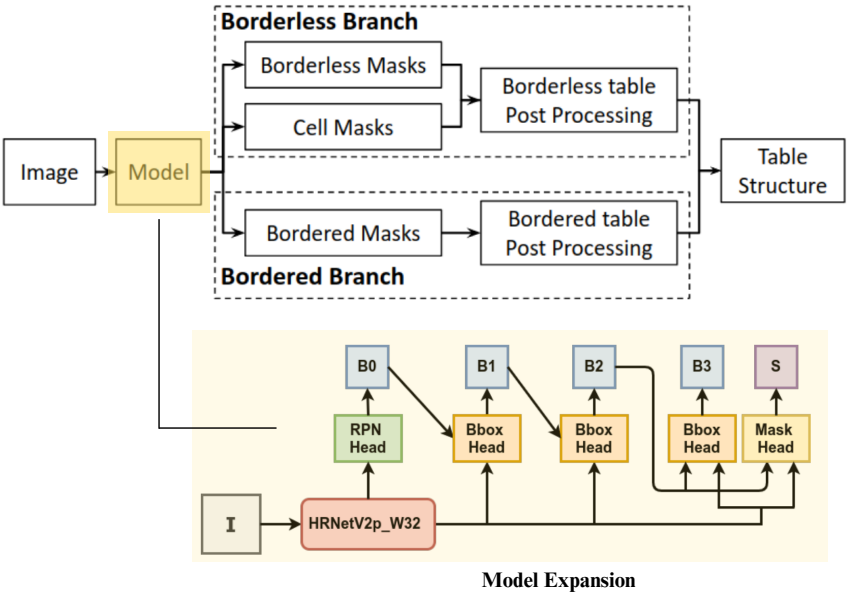


Figure 4.4: CascadeTabNet Model Architecture [55]

3. **PP OCRv2:** For text detection and recognition, we employ PP OCRv2 [18]. The text cells identified by PP OCRv2 are then compared with those detected by CascadeTabNet. Upon establishing correspondence between the detected text and the cells, we calculate their centroids. By finding the shortest possible distance between any two cells, we accurately identify the structure or placement of the text within the rows and columns.

Our proposed methodology for image-based table recognition integrates three distinct models: **DETR** for table detection, **CascadeTabNet** for table structure recognition, and **PPOCR** for text detection and recognition, as illustrated in Figure 4.2. The aforementioned pipeline is specifically developed to overcome issues caused by diverse table types, complicated structures, and visual distortions often found in document images.

At first, the input, whether in image or PDF format, undergoes pre-processing in order to offer standardized data for subsequent analysis. The document image is then fed into the DETR model, which uses object detection to precisely localize tables by creating a fixed-size collection of S predictions. After identifying table regions, they are isolated from the original image and employed as input for the CascadeTabNet model. This specialized deep-learning architecture is specifically designed for precise recognition of table structures, accurately determining the number of rows and columns as well as their corresponding spatial coordinates. Following that, we utilize the PPOCR (Pixel-level Patch-wise Object Character Recognition) method to achieve accurate text detection and recognition within the identified table cells. PPOCR extracts the spatial coordinates of the detected text, and we establish a mapping process using the nearest neighbor approach to align this text with the original coordinates of the table cells obtained from CascadeTabNet. To address the drawbacks of current table-structure recognition models, we tested the Table Transformer (TATR), which incorporates a powerful table-structure deconstruction technique.

4.2.3 Results & Analysis

We performed a comparative assessment of the inference time between our proposed model and Table Transformer (TATR) using the TableBank dataset, which includes 47,053 table images. The findings of this evaluation are illustrated in Table. 4.4. Notably, our model exhibits superior efficiency, showcasing quicker inference times across all measured parameters. Specifically, our

model attains a maximum inference time of 12.7 seconds, a minimum of 5.42 seconds, and an average of 8.23 seconds.

Model	Inference Time (sec.)	
TATR	Max	15.48
	Min	4.95
	Avg	12.43
TC-OCR	Max	**12.7**
	Min	**5.42**
	Avg	**8.23**

Table 4.4: On the TableBank dataset of 47,053 images, the inference time of TC-OCR is compared with TATR.

As indicated in Table. 4.5, our model surpasses the Table Transformer (TATR) concerning Intersection over Union (IOU) and Optical Character Recognition (OCR) accuracy metrics, underscoring its superior performance. Notably, our model attains an impressive IOU of 0.96, signifying its effectiveness in precisely delineating and localizing table elements. Furthermore, our model showcases a noteworthy improvement in OCR accuracy, achieving an impressive 78%, excelling in the pivotal task of accurately recognizing and comprehending the textual content within tables.

Model	IOU	OCR Accuracy
Table Transformer	0.94	62 %
TC-OCR	**0.96**	**78 %**

Table 4.5: IOU and OCR Accuracy of TC-OCR and TATR.

A thorough comparison between the Table Transformer (TATR) model and our proposed method is presented in Table. 4.6. This table illustrates the performance evaluation on various columns within a dataset comprising 240 images and 2,785 rows. Across all columns, our method exhibits superior accuracy, surpassing TATR significantly.

Column No.	No. of Images	Rows	TATR	TC-OCR	TATR Accuracy (%)	TC-OCR Accuracy (%)	Improvement (TC-OCR - TATR)
Total	240	2785	1818	**2485**	65	**89**	24
2	100	1130	838	**1075**	74	**95**	21
3	100	1085	760	**1010**	70	**91**	21
4	40	570	220	**400**	39	**70**	31

Table 4.6: Comparison of outcomes between the TC-OCR and the TATR.

4.2.4 Paper Conclusion

This study proposes an integrated workflow for end-to-end image-based table recognition, harnessing the capabilities of three cutting-edge models: DETR, CascadeTabNet, and PP OCR v2. Through the amalgamation of these models, we successfully address challenges presented by diverse table styles and complex structures in document images. Our method allows for precise restoration

of table layouts and the extraction of cell information from PDF or OCR using bounding boxes. Finally, **our work helps advance data extraction and interpretation in digitized documents, supporting development in the discipline of document analysis.**

4.2.5 Limitations & Future Scope

A significant drawback of this proposed method is its incapacity to accurately identify intricate tables featuring merged cells, nested tables, or irregular structures. Managing such complex layouts presents difficulties in understanding the intricate connections between cells and headers. Consequently, our current approach may not be appropriate for managing these specialized instances, necessitating further research and improvements to effectively tackle these complexities.

For future research, we can delve into integrating this pipeline with Large Language Models (LLMs) to tackle more advanced tasks, such as reasoning and question-answering directly from complex tabular document images. This integration involves extracting the structure and information from the pipeline and feeding it into the prompt of the LLM. This will guide the model on the structure of the tabular data, enabling it to learn how to understand and reason from the tabular structure to address user queries effectively.

4.3 Mathify: Evaluating Large Language Models on Mathematical Problem Solving Tasks

Citation: *Avinash Anand, **Mohit Gupta**, Kritarth Prasad, Navya Singla, Sanjana Sanjeev, Jatin Kumar, Adarsh Raj Shivam, and Rajiv Ratn Shah. Mathify: Evaluating Large Language Models on Mathematical Problem Solving Tasks. NeurIPS'23 Workshop on Generative AI for Education (GAIED), 2023.*

4.3.1 Abstract

The rapid evolution of natural language processing systems and the accumulation of large language models (LLMs) have created numerous possibilities in education and instructional methodologies. These developments present opportunities for personalized learning experiences and prompt feedback, all delivered through accessible and cost-effective means. One notable application of this technological progress lies in mathematical problem-solving. Solving mathematical problems demands the ability to comprehend complex problem statements and the proficiency to execute accurate arithmetic computations at each stage of the problem-solving process. However, assessing large language models' arithmetic prowess remains an area that has received relatively scant attention. In response to this gap, we introduce a comprehensive mathematics dataset named "MathQuest," derived from 11th and 12th standard Mathematics NCERT textbooks. This dataset encompasses mathematical challenges of varying complexity, covering a broad spectrum of mathematical concepts. Leveraging this dataset, we conduct fine-tuning experiments with three prominent LLMs: LLaMA2, WizardMath, and MAmmoTH. These fine-tuned models serve as benchmarks for evaluating their performance on our dataset. Our experiments reveal that MAmmoTH-13B emerges as the most proficient among the three models, demonstrating the highest level of competence in solving the presented mathematical problems. Consequently, MAmmoTH-13B establishes itself as a robust and reliable benchmark for addressing NCERT mathematics problems.

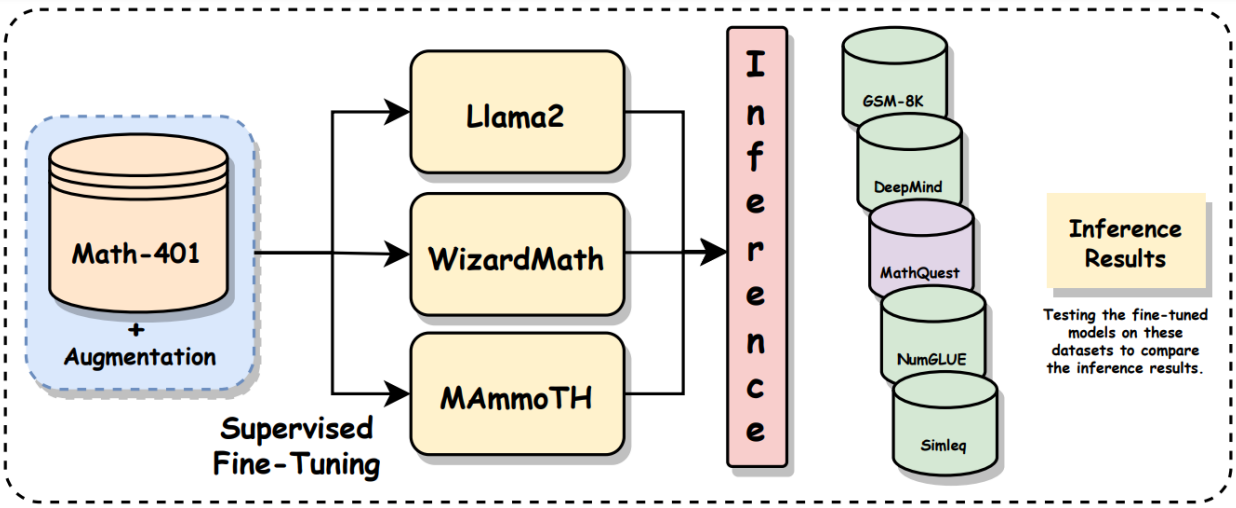


Figure 4.5: Fine-Tuning and Inference flow of LLMs

4.3.2 Methodology

The process of solving mathematical problems involves a diverse range of cognitive skills. It includes understanding problem statements, identifying relevant concepts and formulas, applying

appropriate algorithms and strategies, performing precise calculations, and verifying the validity and reasonableness of solutions. Traditionally, the teaching and assessment of mathematical problem-solving have relied on conventional methods such as textbooks, worksheets, and examinations. However, these methods often provide limited feedback and guidance to learners. With advances in artificial intelligence and natural language processing, LLMs have grown as strong tools for producing natural language text across a broad spectrum of areas and applications. Existing LLMs face significant challenges in solving math word problems that require multi-step arithmetic calculations, complex reasoning, or domain-specific knowledge.

Dataset

In our research experiments, we utilized the Math-401 dataset [77], comprising 401 samples of mathematical problems. This dataset encompasses a diverse range of mathematical operations, including $(+, -, *, /)$, exponentiation, trigonometric, logarithmic functions $(\sin, \cos, \tan, \log, \ln)$, and incorporates integers, decimals, and irrational numbers (π, e). Acknowledging the restricted sample size of Math-401 for effective learning by large language models, we expanded it through augmentation, yielding a dataset size of 302,000 samples. To create our augmented dataset, we utilized the **SymPy**[2] Python library. This library enabled us to generate arithmetic mathematical equations along with their corresponding ground truth values. Table 4.7 offers a detailed breakdown of the question types employed in crafting our augmented dataset.

Type	Range	Decimal Places (1 - 4)	Variables	Count
Small Integer	[-20, 20]	×	(x, y)	65,000
Small Decimal	[-20, 20]	✓	(x, y)	35,000
Small Decimal + Integer	[-20, 20]	✓	(x, y)	39,000
Large Integer	[-1000, 1000]	×	(x, y)	39,000
Large Decimal	[-1000, 1000]	✓	(x, y)	25,000
Large Decimal + Integer	[-1000, 1000]	✓	(x, y)	25,000
3 Terms	[-100, 100]	✓	(x, y, z)	25,000
4 Terms	[-100, 100]	✓	(w, x, y, z)	49,000
Total	-	-	-	302,000

Table 4.7: The distribution of types of question in our augmented Math-401 dataset

MathQuest: In this research work, we have also carefully curated our proprietary dataset, known as MathQuest, by extracting problems from high school mathematics NCERT books. MathQuest serves as a diverse resource, incorporating word problems of various complexities and covering a wide range of mathematical concepts. Our dataset encompasses a total of 14 comprehensive mathematical domains, including sets, trigonometry, binomial theorem, and more. The distribution of samples across these concepts is visually illustrated in Figure 4.6. Our dataset comprises a total of 223 samples, with the "Sequence and Series" category notably having the highest number of problems, as indicated in the charts.

This study aims to improve the problem-solving capacities of LLMs within the field of mathematics. Initially, we noted that established publicly available models, including LLaMA [63] and Vicuna [13], struggled with basic mathematical tasks like subtraction and addition. This insight became the catalyst behind our study, motivating us to improve LLMs ability to grasp and solve mathematical problems.

[2] `https://www.sympy.org/en/index.html`

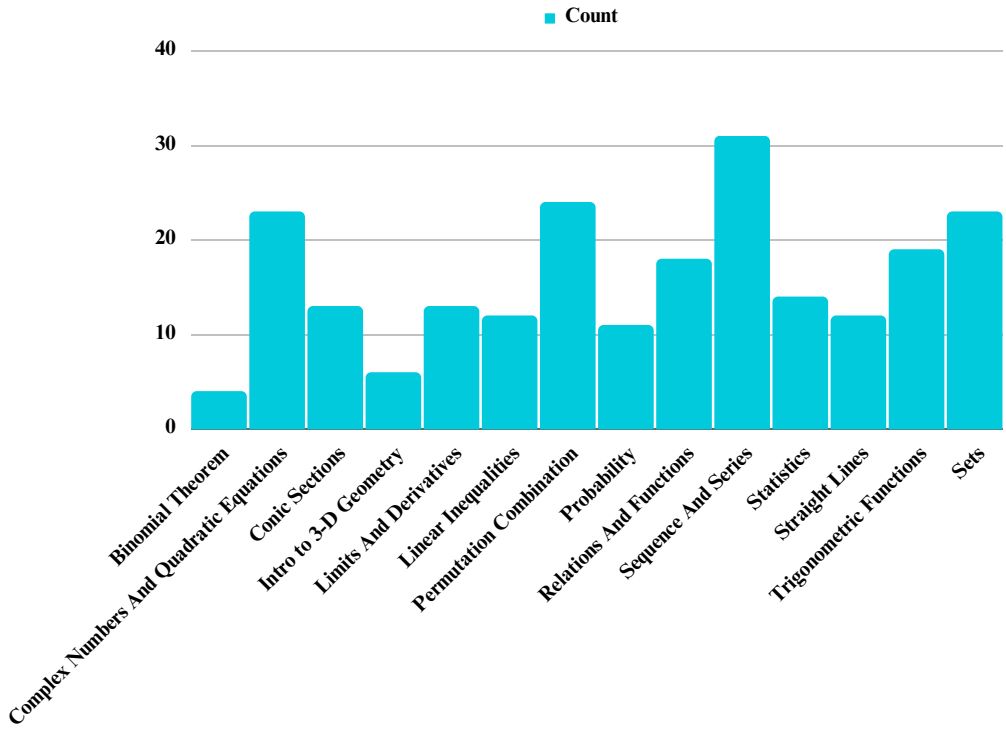


Figure 4.6: Distribution of the number of questions for each mathematical topic.

To achieve this goal, we used an instructional strategy similar to how children are taught mathematics. We started with basic operators like $+, -, *$ and $/$ before moving on to more sophisticated operators and expressions. In a same spirit, we wanted to familiarize LLMs with the meanings of numerous mathematical operators and expressions. To help with this procedure, we used the Math-401 dataset [77], which is a good resource that contains 401 data samples that include fundamental mathematics questions and their solutions. Given the dataset's limited size, we expanded it to add more diversity and complexity, guaranteeing that the model could learn and grasp advanced mathematical ideas throughout training. To fine-tune, we used three popular LLMs: MAmmoTH [78], WizardMath [45], and LLaMA-2 [64].

4.3.3 Results & Analysis

In this section, we delve into the specifics of our conducted experiments, providing an overview of the experimental setup. We ran trials using three popular large language models: MAmmoTH, WizardMath, and LLaMA2. We tested both the 7B and 13B versions of these LLMs. Our experiments were conducted in two stages. In the first stage, we loaded the original model weights and performed inference on our test set. In the second stage, we fine-tuned the LLMs using the Math-401, which we have augmented in this research work.

The dataset was partitioned into 2.41K train, 30K test and 30K validation samples. We used QLora for fine-tuning, which optimizes memory and reduces computing costs in a pretrained language model by 4-bit quantization. Each model is fine-tuned for #10 epochs at a step size of 3e-4.

To evaluate performance, we measured accuracy by assessing the match between generated answers and the actual solutions for five open-source datasets: GSM-8K, DeepMind, SimulEq, NumGLUE, and Math-401. These datasets offer ground truth answers, enabling the calculation of exact match accuracy.

Table. 4.8 shows the exact match accuracy of three models (7B and 13B variants) before fine-

Model	# of Params	Accuracy					
		GSM-8K	DeepMind	NumGLUE	SimulEq	Math-401*	MathQuest
LLaMA-2	7B	16.0	46.0	37.0	11.0	10.0	10.4
LLaMA-2	13B	22.0	50.0	42.0	15.0	10.0	14.1
WizardMath	7B	61.0	51.0	54.0	27.0	6.0	14.6
WizardMath	13B	65.0	55.0	70.0	36.0	8.0	14.3
MAmmoTH	7B	43.0	49.0	54.0	23.0	11.0	12.2
MAmmoTH	13B	44.0	48.0	56.0	26.0	14.0	18.1

Table 4.8: Before fine-tuning, results for 100 samples from five datasets and our MathQuest dataset. (*) denotes the augmented subset of Math-401.

tuning on five datasets and our MathQuest dataset. Overall, performance is notably lower on the SimulEq dataset and our augmented Math-401 dataset. This is likely due to the presence of complex problems that require additional knowledge, such as questions like ”Number of red color cards in a deck of 52 cards.”

Model	# of Params	Accuracy					
		GSM-8K	DeepMind	NumGLUE	SimulEq	Math-401*	MathQuest
LLaMA-2	7B	30.0	46.0	45.0	15.0	17.0	10.6
LLaMA-2	13B	42.0	51.0	54.0	16.0	24.0	20.3
WizardMath	7B	64.0	55.0	52.0	29.0	15.0	16.01
WizardMath	13B	68.0	56.0	70.0	38.0	10.0	20.1
MAmmoTH	7B	56.0	50.0	62.0	24.0	16.0	18.5
MAmmoTH	13B	67.0	51.0	64.0	34.0	18.0	**24.0**

Table 4.9: After fine-tuning, results for 100 samples from five datasets and our MathQuest dataset. (*) denotes the augmented subset of Math-401.

Table. 4.9 provides an in-depth analysis of the accuracy outcomes following the fine-tuning process. In summary, all models improved significantly in accuracy post fine-tuning on our heterogeneous question-and-answer dataset. We can also see that, models with 13B parameters were more accurate than those with 7B parameters.

The major findings from Tables 4.8 and 4.9 show that MAmmoTH-13B is the best-performing model for our MathQuest dataset, with the highest accuracy after fine-tuning (24.0%). It’s worth noting that both MAmmoTH-7B and 13B produced results with precision up to two decimal places, demonstrating their accuracy. Table. 4.9 shows that MathQuest is a tougher difficulty due to its complexity and diversity, resulting in lesser accuracy when compared to other datasets.

4.3.4 Paper Conclusion

In conclusion, this research work provides Large Language Models (LLMs) with critical reasoning abilities for exact mathematical problem solution. The MathQuest dataset includes customizable question-and-answer pairs that address one or more mathematical operators as well as expressions. These challenges direct the model’s approach to incremental problem resolution, with the goal of improving solution clarity and precision. Our findings show considerable gains in

solution correctness and comprehensibility, which will be useful for educators and students looking to improve their problem-solving mathematical ability.

While this study provides a solid foundation for using Generative LLMs to advance mathematical problem-solving, further adjustments and optimizations are required to broaden its application to a wider range of contexts. Finally, our research helps to increase conceptual comprehension and numerical problem-solving abilities in high school-level mathematical question-answering, providing essential aid to pupils as well as professionals dealing with challenging questions via LLMs.

4.3.5 Limitations & Future Scope

While the proposed solution effectively handles simple mathematical problems, it occasionally faces difficulties when confronted with complex mathematical scenarios that require retaining variable values for subsequent equations. Additionally, our work exhibits a limitation concerning the partial enhancement of LLMs' reasoning abilities in solving mathematical problems. However, it struggles to address complex expressions containing nested brackets within equations.

In our future work endeavors, we target to address these drawbacks by expanding our training dataset. Given the rapid pace of advancements in LLM research, with new techniques, models, and prompting strategies emerging daily, we plan to integrate more advanced techniques to enhance LLM reasoning capabilities. This includes leveraging prompting techniques such as Recall, CoT, and Self-Consistency CoT, as well as advanced techniques like RLHF. By incorporating these methodologies, we seek to further refine LLMs' reasoning abilities and effectively address the challenges posed by complex mathematical problems.

4.4 Empowering Large Language Models for Physics: Insights from Reinforcement Learning with Human and AI Feedback

Status: *Pending*

4.4.1 Abstract

Currently, Large Language Models (LLMs) have shown great performance in general tasks like text summarization. However, they often struggle with complex arithmetic questions and mathematical reasoning tasks. While simple approaches, such as fine-tuning LLMs for specific mathematical problem-solving tasks, have improved their reasoning abilities, they still face challenges when encountering new questions or variations in problem-solving approaches. This study aims to enhance LLMs to effectively handle physics-related questions extracted from the PhyQA dataset, which consists of problem sets from Indian NCERT textbooks for grades 11 and 12. We employ Reinforcement Learning to improve the efficacy and accuracy of our models in arithmetic reasoning for the PhyQA dataset. Various reinforcement learning methods, including DPO, ReMax, and PPO optimization, are explored to assess their performance in physics problem-solving across different scenarios. A crucial aspect of our approach involves integrating human and artificial intelligence feedback, referred to as Reinforcement Learning with Human and Artificial Intelligence Feedback. This innovative approach helps train our models to generate more logical and reasonable solutions to physics problems. In evaluations, the MISTRAL-PPO model stands out for its ability to produce reasonable solutions, achieving commendable scores such as a 58.67% METEOR Score, an 80.39% Reasoning Score, and 38.0% accuracy in a manual evaluation of 100 random samples, quantitatively measuring the model's reasoning abilities.

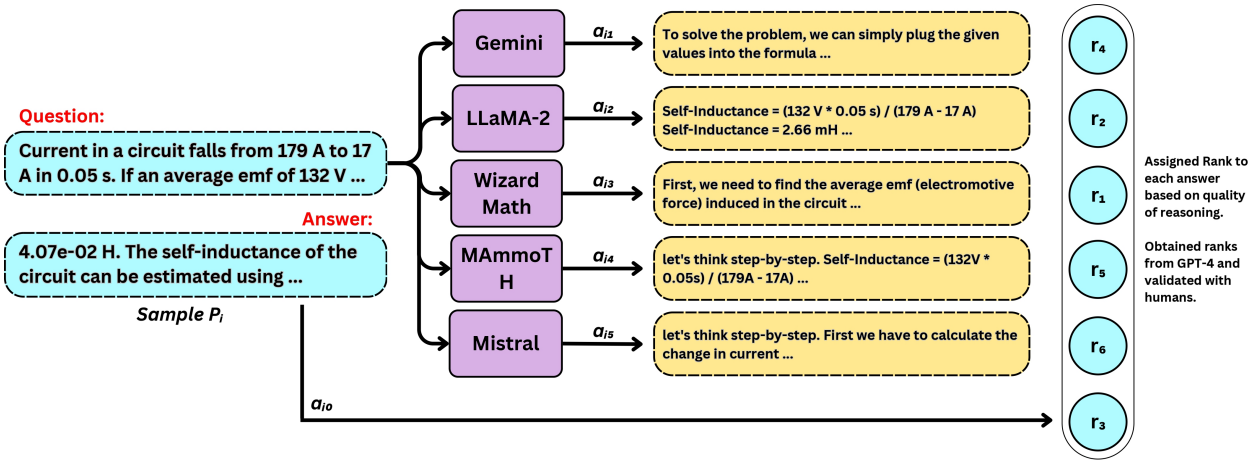


Figure 4.7: Our innovative method for prioritizing responses for the Preference Dataset

In the quest to improve a language model's ability to generate responses that better match human preferences, [52] introduced Reinforcement Learning from Human Feedback (RLHF). RLHF is a machine learning approach that integrates reinforcement learning techniques, including rewards and comparisons, with human guidance to train an artificial intelligence agent. The RLHF process unfolds in three distinct phases: collecting human feedback, training the reward model, and refining the policy. At the heart of RLHF lies preference data, which involves rating and comparing various responses generated in response to the same prompt. However, gathering human feedback to

construct preference data poses a significant challenge. Obtaining high-quality feedback from humans and accounting for the potential sub-optimal nature of human input can be quite complex.

To address this challenge, we introduced RLHAIF, which combines both human feedback and AI feedback to incorporate diverse preference datasets. The preference data generated through modern LLMs have the potential to enhance generalization abilities and improve robustness to various response patterns.

4.4.2 Methodology

Dataset

The dataset used for experimentation, known as **PhyQA**, is an extension of SCIMAT's science problems [3, 4]. It contains 9.5K high school physics questions and answers, covering topics taught to students aged 15-19, such as Alternating Current, Atoms and Nuclei, and more. Figure 4.8 shows the distribution of problems across topics.

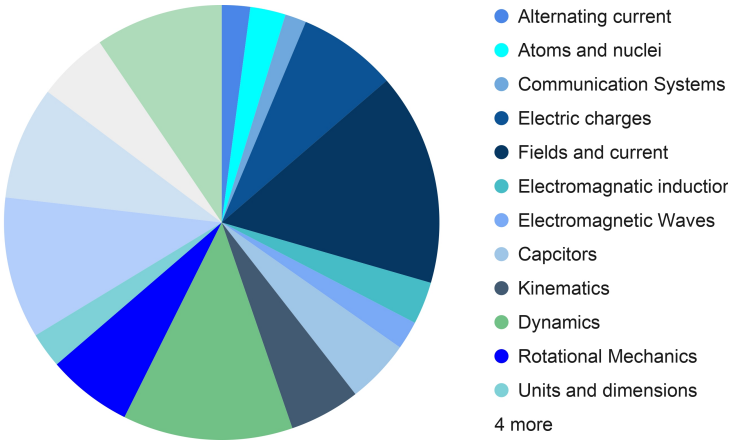


Figure 4.8: PhyQA Topic Distribution

Analyzing PhyQA provides insights into question and solution characteristics, aiding Large Language Models (LLMs) performance. Questions average 35.74 words, while solutions average 54.95 words, with maximum lengths of 75 and 220 words, respectively. Concise and precise solutions in PhyQA help LLMs better understand and address questions. The PhyQA dataset P comprises 8100 samples. Each sample includes a question q_i and its corresponding answer a_{i0}. To augment the dataset, we expanded our investigation by generating answers using four open-source large language models: LLaMA2-7B, WizardMath-7B, Mistral-7B, and MAmmoTH-7B, alongside Gemini, a closed-source model. This expansion yields six answers for each question, offering a broad and diverse array of responses to bolster our research efforts.

We proceed to rank these answers on a scale from 1 to 6 based on the quality of their reasoning, using prompts similar in detail to [34]. Lower ranks indicate higher-quality reasoning in the answers. To establish these rankings, we initially employ GPT-4 to generate rankings, which are then evaluated and re-ranked by human evaluators. This process is carried out to address any inaccuracies in the rankings generated by GPT-4. Following this, we form pairs of answers, designating one to be accepted and the other to be rejected. For each data sample P_i, we generate three distinct pairs of answers based on the rankings.

This modified dataset is then employed for training the Reward Model for which we have used **LLaMA-2 13B** model. The visual representation of our preference data creation process is shown in Figure. 4.7. For the experiments, we have used the 7B variants of the following LLMs, LLaMA2, WizardMath, MetaMath, LLeMMA, and Mistral. Additionally, we have divided the explanation of

our approach and experimental setup into three parts for clarity. Firstly, we delve into different RL algorithms like DPO, PPO, ReMax and their setup. Secondly, we explore Chain of Thought (CoT) prompting techniques, and for this we have experimented with both zero-shot and few-shot CoT. Lastly, we discuss the Recall Prompting technique.

Parameter	**PPO**	**DPO**	**ReMax**
KL Coefficient	0.2	0.2	0.2
Epochs	3	3	1
Batch Size	4	2	1
Gradient Accumulation	2	8	1
Learning rate	3e-5	3e-5	1e-6

Table 4.10: Hyper-parameter configuration used in training RL models with different RL Policy Optimization Methods.

4.4.3 Results & Analysis

We have conducted extensive evaluations to assess the performance of the models and the various approaches. The evaluation comprises thorough error analysis, accuracy assessments, and reasoning scoring, providing a comprehensive understanding of each model's strengths and weaknesses.

Model	**Setting**	**METEOR**	**BLUE-1**	**BLUE-2**	**BLUE-3**	**BLUE-4**	**ROUGE1**	**ROUGE2**	**ROUGEL**	**ROUGELSUM**	**BERTScore**
LLaMA2-7B	0-Shot	36.65	18.73	13.55	10.64	8.35	31.97	17.07	22.01	27.11	79.07
	3-Shot	25.28	20.90	13.61	9.78	7.18	29.07	12.57	20.55	24.18	76.71
	SFT	28.09	8.37	5.54	4.08	2.97	20.65	9.69	13.56	16.37	76.87
	PPO	39.32	7.10	6.27	5.90	5.33	31.34	24.58	28.23	28.98	82.48
	DPO	35.64	18.74	13.24	9.99	7.49	33.58	17.78	24.07	27.78	80.19
	Remax	37.85	25.69	19.49	16.04	13.08	37.72	22.59	29.46	31.70	81.26
	Recall	23.82	21.00	13.64	9.91	7.20	26.72	12.15	18.78	21.19	74.16
Mistral	0-Shot	28.59	15.42	10.54	7.93	5.95	26.25	13.03	18.77	22.37	75.76
	3-Shot	17.59	13.51	9.26	7.0	5.24	18.79	8.36	14.43	16.2	72.93
	SFT	25.53	6.58	4.62	3.6	2.74	19.97	9.98	13.53	16.18	77.08
	PPO	**58.67**	**40.04**	**35.87**	**34.5**	**32.81**	**57.94**	**51.55**	**56.32**	**56.53**	**87.49**
	DPO	29.94	13.79	8.69	6.08	4.15	29.68	13.3	19.59	23.56	77.42
	Remax	-	-	-	-	-	-	-	-	-	-
	Recall	20.19	10.06	6.71	4.95	3.59	21.35	9.11	15.24	17.56	73.1

Table 4.11: Evaluation of LLaMA-2 and Mistral model with different settings (0-shot, 3-shot, SFT, PPO, DPO, ReMax, Recall) using various scoring metrics (BLUE, ROUGE, METEOR, BERT).

In Table 4.11, I've only shared the results of top two models, the results of other models are presented in the paper. In a analysis of various model settings, as presented in Table 4.11, Mistral-PPO achieved the highest overall scores, consistently scoring approximately 35.0 across BLEU-1 to BLEU-4 metrics, with a METEOR score of 58.67. This consistency indicates a strong alignment between the predicted and target words. Additionally, the LLaMA2-7B model demonstrated impressive performance, especially in aligning with preferred answers. However, it slightly lagged behind Mistral-PPO in accurately matching specific words and displayed limitations in semantic understanding and solving arithmetic problems.

Although Mistral showcases strong logical and mathematical reasoning skills in certain contexts, it occasionally commits notable errors in insignificant stages. These inaccuracies in problem analysis, concept recall, and application underscore further avenues for exploration.

To assess the precision of our models, we examined a 100 random samples from the dataset. Table 4.12 displays the comparative outcomes as follows:

- GPT-4 leads with a 72% accuracy rate, followed by GPT-3.5 at 40%.
- Mistral, employing the PPO policy, achieves a 38.0% accuracy, while LLaMA2-7B with the PPO policy registers an 18% accuracy.

These findings illustrate that although GPT-3.5 outperforms our suggested top model, it still encounters computational errors, and occasionally our proposed model outperforms it. With scalability, we have the potential to surpass GPT-3.5, as the results are not significantly different from those of the Mistral-PPO 7B model.

Model	Setting	Correct	Wrong	Total
LLaMA2-7B	SFT	9	91	100
	PPO	18	82	100
	DPO	10	90	100
	Recall	14	86	100
Mistral	SFT	21	79	100
	PPO	**38**	62	100
	DPO	22	78	100
	Recall	16	84	100
GPT-3.5	—	40	60	100
GPT-4	—	72	28	100

Table 4.12: Model's output performance with Human Evaluations

For a detailed analysis of the reasoning evaluation of each response, we have formulated a six-step reasoning assessment. Each skill point is assessed according to the LLM's proficiency in executing specific problem-solving aspects, including identifying the **correct context (CA)**, interpreting **physics concepts (PD)**, **performing calculations (AC)**, maintaining **logical coherence (LR)**, demonstrating an **understanding of concepts (CU)**, and identifying or **rectifying errors (ED)**.

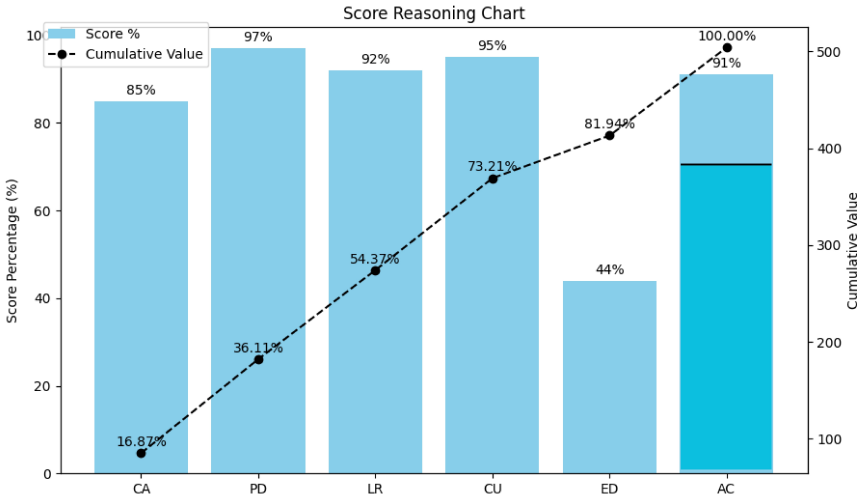


Figure 4.9: Reasoning Score Distribution on Mistral-PPO's 100 random sample responses

Figure. 4.9 also demonstrates that LLM faces challenges in arithmetic calculations in approximately 91.0% of cases, as determined by assessments from Human Annotators across 100 responses

from the Mistral-PPO model. Out of this total, the model accurately follows the sequence of steps and formulas to solve the solution 76.92% of the time. However, in approximately 23.08% of cases, the model fails to execute accurate arithmetic calculations. This leads to an overall score of 62.0 for incorrect answers, as shown in Table. 4.12.

4.4.4 Paper Conclusion

This study presents RLHAIF, an effective and efficient strategy for improving the physics problem-solving capabilities of large language models (LLMs) and aligning their responses more closely with human preferences. The revolutionary RLHAIF methodology ranks answers using human and AI-generated input, resulting in a more diversified and resilient model training process. Experiments with various LLMs consistently demonstrate that the Mistral-PPO created from this approach beats its equivalents, establishing itself as a reliable benchmark for the PhyQA dataset. By bridging the gap between human intuition and LLM replies, RLHAIF has the potential to advance natural language understanding and generation significantly.

4.4.5 Limitations & Future Scope

The possibility of students misusing the model for cheating in assignments or quizzes raises ethical concerns. To tackle this issue, future research could explore shifting focus from providing direct answers to offering hints and structured reasoning, with the aim of enhancing students' conceptual understanding and problem-solving skills.

Our investigation has been constrained to a 7B model due to computational costs and resource efficiency. Future directions could involve exploring the performance spectrum of larger models with increased parameters, such as 13B or 70B.

Chapter 5

Conclusion

5.1 Discussion

The presented research makes substantial contributions to the area of document analysis and information extraction, addressing challenges across various domains. The comprehensive approach to document analysis, particularly in visually rich documents (VrDs), is crucial for extracting information from diverse sources such as receipts, invoices, and forms. The integration of deep learning models, including PP OCR v2, DETR, and Cascade TabNet, for table detection and recognition showcases a remarkable improvement in accuracy and efficiency, surpassing existing methods. The introduction of RanLayNet, a synthetic document dataset for domain adaptation in layout detection, proves to be a versatile tool for training models with resilience and versatility to handle various document formats.

Moreover, the research ventures into the realm of educational AI, specifically focusing on mathematical and physics problem-solving. The creation of the MathQuest dataset and subsequent fine-tuning experiments with prominent Large Language Models provide a benchmark for evaluating their competence in solving complex mathematical problems drawn from 11th and 12th standard NCERT textbooks. A new method for preference data labeling for Reinforcement learning to solve physics problems using LLMs provides a efficient mechanism which reduces extensive human-annotation for providing feedback to model responses, and best optimized RL approach for this task of solving physics problems. This contributes to the evolving landscape of educational technology, offering tailored learning experiences and immediate feedback through advanced NLP systems.

5.2 Conclusion

In conclusion, this thesis successfully addresses the challenges (PATTERNS) in document analysis and information extraction & issues (PROBLEMS) in the mathematical reasoning of LLMs, collectively advance the capabilities of Large Language Models (LLMs) in handling complex tasks across diverse domains (MATHS, and PHYSICS). This work represents a significant stride towards achieving more robust, accurate, and scalable solutions, contributing to the broader landscape of document understanding, NLP and LLM research. The research outcomes not only enhance accuracy and efficiency but also provide practical solutions applicable in real-world scenarios.

5.3 Future Recommendations

As the field of Large Language Models continues to progress rapidly, there is a promising avenue for incorporating these models into this research to address problems with greater precision and accuracy. While our current work utilizes OCR-based models for tabular data extraction and layout detection, the emergence of very large models with over 7 billion parameters, or LLMs equipped with vision capabilities like LLaVA, UReader, and DocLLM, offers a more sophisticated approach to understanding document layouts. These OCR-Free Document Analysis methods present an exciting opportunity for future exploration, particularly in conjunction with Multimodal Large Language Models.

Reinforcement learning plays a central role in improving the generated response quality of models like ChatGPT, making them more aligned with human preferences. To enhance the mathematical reasoning abilities of LLMs, future research can focus on leveraging more advance Reinforcement

Learning with Human Feedback (RLHF) techniques with different policy optimizations, and more better preference labelling mechanisms. As the combination of RLHF with LLMs proven to be a promising approach for refining LLMs reasoning capabilities not specific to mathematics. Moreover, this approach can be extended to other educational domains such as chemistry, computer science etc. By developing Open-Source LLMs tailored for subject matter experts, we possess the capability to transform the landscape of educational AI and enrich the learning journey for students.

"***Our Intelligence is what makes us HUMAN, and AI is an extension of that Quality.***"
$\sim$ *Yann LeCun*

Chapter 6

List of Publications

S.No	Paper Title	Venue	Link
1.	**RanLayNet**: A Dataset for Document Layout Detection used for Domain Adaptation and Generalization.	MMAsia' 23	🔗
2.	**TC-OCR**: TableCraft OCR for Efficient Detection & Recognition of Table Structure & Content.	MMIR' 23	🔗
3.	**Mathify**: Evaluating Large Language Models on Mathematical Problem Solving Tasks.	GAIED' 23	🔗
4.	Empowering Large Language Models for Physics: Insights from Reinforcement Learning with Human and AI Feedback.	ACMMM' 24	-

Chapter 7

Appendix

7.1 Prompt Templates

7.1.1 Model Inference Prompt

```
 Below is an instruction that describes a task.  Write a response
that appropriately completes the request.  Beware of wrong
calculation and do not repeat it.

### Instruction:
sample["instruction"]

### Input:
sample["input"]

### Response:
```

7.1.2 Model Fine-Tuning Prompt for Mathify, and Physics Paper

```
 Below is an instruction that describes a task.  Write a response
that appropriately completes the request.  Beware of wrong
calculation and do not repeat it.

### Instruction:
sample["Question"]

### Input:
sample["Input"]

### Response:
sample["ground-truth"]
```

7.1.3 Prompt for GPT-4 to rank the model responses

In the realm of mathematical word problems, finding the correct and reasonable solutions is imperative. Various models may generate different answers to the same problem. The evaluation of these solutions is critical to determine which model generates the most promising results. Below we define three essential evaluation axes: correctness, clarity, and overall reasonableness.

Correctness: This axis corresponds to the question, "Is the provided solution mathematically correct?" The response will be marked as correct if there are no mathematical errors, and it accurately addresses all aspects of the problem.

Clarity: This axis corresponds to the query, "How comprehensible is the solution?" A clear response is one that can be easily followed and understood. It must use the correct mathematical notation and language, and justify each step transparently.

Overall Reasonableness: This axis pertains to the question, "How reasonable is the solution in its entirety?" This evaluates the correctness and clarity, and further considers factors such as the appropriateness of the technique used and the practicality of the solution in the given real-world context.

You are an expert evaluator, tasked with evaluating multiple solutions, each presented by a different model, to a given mathematical word problem. After careful consideration of the correctness, clarity, and overall reasonableness of each solution, your task is to rank them from the best to the worst. This ranking should mirror your judgement on which solution represents the most clear, correct, and overall reasonable approach to the problem.

Question: sample["question"]

Solution 1- sample["GEMINI"]
Solution 2- sample["LLAMA2-7B"]
Solution 3- sample["WIZARD7B"]
Solution 4- sample["MAMMOTH7B"]
Solution 5- sample["MISTRAL7B"]
Solution 6- sample["Target"]

Consider the correctness, clarity and overall reasonableness of each solution and to rank them from best to the worst.

Rationale:

Bibliography

[1] Riaz Ahmad, Muhammad Tanvir Afzal, and Muhammad Abdul Qadir. Information extraction from pdf sources based on rule-based system using integrated formats. In *Semantic Web Challenges: Third SemWebEval Challenge at ESWC 2016, Heraklion, Crete, Greece, May 29-June 2, 2016, Revised Selected Papers 3*, pages 293–308. Springer, 2016.

[2] Alammar, J (2018). The Illustrated Transformer, https://jalammar.github.io/illustrated-transformer/.

[3] Avinash Anand, Krishnasai Addala, Kabir Baghel, Arnav Goel, Medha Hira, Rushali Gupta, and Rajiv Ratn Shah. Revolutionizing high school physics education: A novel dataset. In *Big Data and Artificial Intelligence: 11th International Conference, BDA 2023, Delhi, India, December 7–9, 2023, Proceedings*, page 64–79, Berlin, Heidelberg, 2023. Springer-Verlag.

[4] Avinash Anand, Arnav Goel, Medha Hira, Snehal Buldeo, Jatin Kumar, Astha Verma, Rushali Gupta, and Rajiv Ratn Shah. Sciphyrag - retrieval augmentation to improve llms on physics q&a. In *Big Data and Artificial Intelligence: 11th International Conference, BDA 2023, Delhi, India, December 7–9, 2023, Proceedings*, page 50–63, Berlin, Heidelberg, 2023. Springer-Verlag.

[5] Avinash Anand, Mohit Gupta, Kritarth Prasad, Navya Singla, Sanjana Sanjeev, Jatin Kumar, Adarsh Raj Shivam, and Rajiv Ratn Shah. Mathify: Evaluating large language models on mathematical problem solving tasks, 2023.

[6] Avinash Anand, Raj Jaiswal, Pijush Bhuyan, Mohit Gupta, Siddhesh Bangar, Md. Modassir Imam, Rajiv Ratn Shah, and Shin'ichi Satoh. Tc-ocr: Tablecraft ocr for efficient detection & recognition of table structure & content. In *Proceedings of the 1st International Workshop on Deep Multimodal Learning for Information Retrieval*, page 11–18, New York, NY, USA, 2023. Association for Computing Machinery.

[7] Avinash Anand, Raj Jaiswal, Mohit Gupta, Siddhesh S Bangar, Pijush Bhuyan, Naman Lal, Rajeev Singh, Ritika Jha, Rajiv Ratn Shah, and Shin'Ichi Satoh. Ranlaynet: A dataset for document layout detection used for domain adaptation and generalization. In *Proceedings of the 5th ACM International Conference on Multimedia in Asia*, New York, NY, USA, 2024. Association for Computing Machinery.

[8] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9357–9366, 2019.

[9] Maciej Besta1, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michał Podstawski, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of thoughts: Solving elaborate problems with large language models, 2023.

[10] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler,

Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[11] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.

[12] Raghavendra Chalapathy, Ehsan Zare Borzeshi, and Massimo Piccardi. Bidirectional lstm-crf for clinical concept extraction, 2016.

[13] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[14] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[15] Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic detr: End-to-end object detection with dynamic attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2988–2997, 2021.

[16] Yonghao Dang, Fuxing Yang, Baiquan Su, Jianqin Yin, and Jun Liu. Dbnet: A new generalized structure efficient for classification. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1–6, 2019.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[18] Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, and Haoshuang Wang. Pp-ocr: A practical ultra lightweight ocr system, 2020.

[19] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation, 2015.

[20] Liangcai Gao, Yilun Huang, Herve Dejean, Jean-Luc Meunier, Qinqin Yan, Yu Fang, Florian Kleber, and Eva Lang. Icdar 2019 competition on table detection and recognition (ctdar). In *ICDAR 2019*, pages 1510–1515, 09 2019.

[21] Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models, 2023.

[22] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[23] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[24] Max C. Göbel, Tamir Hassan, Ermelinda Oro, and Giorgio Orsi. Icdar 2013 table competition. *2013 12th International Conference on Document Analysis and Recognition*, pages 1449–1453, 2013.

[25] Philip John Gorinski, Honghan Wu, Claire Grover, Richard Tobin, Conn Talbot, Heather Whalley, Cathie Sudlow, William Whiteley, and Beatrice Alex. Named entity recognition for electronic health records: A comparison of rule-based and machine learning approaches, 2019.

[26] Yelin He, Xianbiao Qi, Jiaquan Ye, Peng Gao, Yihao Chen, Bingcong Li, Xin Tang, and Rong Xiao. Pingan-vcgroup's solution for icdar 2021 competition on scientific table image recognition to latex, 2021.

[27] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021.

[28] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging, 2015.

[29] K. Itonori. Table structure recognition based on textblock arrangement and ruled line position. In *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93)*, pages 765–768, 1993.

[30] Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274, 2023.

[31] Pratik Kayal, Mrinal Anand, Harsh Desai, and Mayank Singh. Icdar 2021 competition on scientific table image recognition to latex, 2021.

[32] Rahul Khanna and Mariette Awad. *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers, 04 2015.

[33] Mehran Khodabandeh, Arash Vahdat, Mani Ranjbar, and William G Macready. A robust learning approach to domain adaptive object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 480–490, 2019.

[34] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback, 2023.

[35] Jee Hee Lee, June seong Yi, and JeongWook Son. Development of automatic-extraction model of poisonous clauses in international construction contracts using rule-based nlp. *J. Comput. Civ. Eng.*, 33, 2019.

[36] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[37] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022.

[38] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. Tablebank: A benchmark dataset for table detection and recognition, 2020.

[39] Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. Docbank: A benchmark dataset for document layout analysis. *arXiv preprint arXiv:2006.01038*, 2020.

[40] Xiang Li, Wenhai Wang, Wenbo Hou, Ruo-Ze Liu, Tong Lu, and Jian Yang. Shape robust text detection with progressive scale expansion network, 2018.

[41] Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making large language models better reasoners with step-aware verifier, 2023.

[42] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[43] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks, 2015.

[44] Ning Lu, Wenwen Yu, Xianbiao Qi, Yihao Chen, Ping Gong, Rong Xiao, and Xiang Bai. Master: Multi-aspect non-local network for scene text recognition. *Pattern Recognition*, 117:107980, September 2021.

[45] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct, 2023.

[46] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023.

[47] Tanzim Mahmud, K. M. Azharul Hasan, Mahtab Ahmed, and Thwoi Hla Ching Chak. A rule based approach for nlp based query processing. *2015 2nd International Conference on Electrical Information and Communication Technologies (EICT)*, pages 78–82, 2015.

[48] Ajoy Mondal, Peter Lipps, and C. V. Jawahar. Iiit-ar-13k: A new dataset for graphical object detection in documents, 2020.

[49] Matteo Muffo, Aldo Cocco, and Enrico Bertino. Evaluating transformer language models on arithmetic operations using number decomposition, 2023.

[50] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, and William Saunders. Webgpt: Browser-assisted question-answering with human feedback, 2023.

[51] Duc-Dung Nguyen. Tablesegnet: a fully convolutional network for table detection and segmentation in document images. *Int. J. Doc. Anal. Recognit.*, 25(1):1–14, mar 2022.

[52] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

[53] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap, 2023.

[54] Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter Staar. Doclaynet: A large human-annotated dataset for document-layout segmentation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22. ACM, aug 2022.

[55] Devashish Prasad, Ayan Gadpal, Kshitij Kapadni, Manish Visave, and Kavita Sultanpure. Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents, 2020.

[56] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

[57] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[58] R. Smith. An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633, 2007.

[59] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, and Alec Radford Dario Amodei Paul Christiano. Learning to summarize from human feedback, 2023.

[60] Dima Suleiman, Arafat A. Awajan, and Wael Al Etaiwi. The use of hidden markov model in natural arabic language processing: a survey. In *EUSPN/ICTH*, 2017.

[61] Li Im Tan, Wai San Phang, Kim On Chin, and Anthony Patricia. Rule-based sentiment analysis for financial news. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1601–1606, 2015.

[62] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[63] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[64] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[66] Cunxiang Wang, Boyuan Zheng, Yuchen Niu, and Yue Zhang. Exploring generalization ability of pretrained language models on arithmetic and logical reasoning, 2021.

[67] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023.

[68] Yalin Wang, Ihsin T. Phillips, and Robert M. Haralick. Table structure understanding and its performance evaluation. *Pattern Recognit.*, 37:1479–1497, 2004.

[69] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[70] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. *Facebook Research*, 2019.

[71] Yuanzhen Xie, Tao Xie, Mingxiong Lin, WenTao Wei, Chenglin Li, Beibei Kong, Lei Chen, Chengxiang Zhuo, Bo Hu, and Zang Li. Olagpt: Empowering llms with human-like problem-solving abilities, 2023.

[72] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 1192–1200, New York, NY, USA, 2020. Association for Computing Machinery.

[73] Aditya Yadav, Shauryan Singh, Muzzamil Siddique, Nileshkumar Mehta, and Archana Kotangale. Ocr using crnn: A deep learning approach for text recognition. In *2023 4th International Conference for Emerging Technology (INCET)*, pages 1–6, 2023.

[74] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023.

[75] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing, 2017.

[76] Ning Yu, Chia-Chih Chen, Zeyuan Chen, Rui Meng, Gang Wu, Paul Josel, Juan Carlos Niebles, Caiming Xiong, and Ran Xu. Layoutdetr: Detection transformer is a good multimodal layout designer, 2023.

[77] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, and Songfang Huang. How well do large language models perform in arithmetic tasks?, 2023.

[78] Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning, 2023.

[79] Daqian Zhang, Ruibin Mao, Runting Guo, Yang Jiang, and Jing Zhu. Yolo-table: disclosure document table detection with involution. *Int. J. Doc. Anal. Recognit.*, 26(1):1–14, may 2022.

[80] Peng Zhang, Yunlu Xu, Zhanzhan Cheng, Shiliang Pu, Jing Lu, Liang Qiao, Yi Niu, and Fei Wu. Trie: end-to-end text reading and information extraction for document understanding. *ACM*, pages 1413–1422, 2020.

[81] Xinyi Zheng, Doug Burdick, Lucian Popa, Xu Zhong, and Nancy Xin Ru Wang. Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context, 2020.

[82] Li Zhong, Zhen Fang, Feng Liu, Jie Lu, Bo Yuan, and Guangquan Zhang. How does the combined risk affect the performance of unsupervised domain adaptation approaches? In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11079–11087, 2021.

[83] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. Image-based table recognition: data, model, and evaluation, 2020.

[84] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE, 2019.

[85] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models, 2023.

[86] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: An efficient and accurate scene text detector. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2642–2651, 2017.