Pramod Gupta - Thesis.pdf

by Pramod Gupta

Submission date: 18-Mar-2024 10:21AM (UTC+0000)
Submission ID: 226973497
File name: Pramod_Gupta_-_Thesis.pdf (1.89M)
Word count: 25433
Character count: 164162

USING LLMS TO EXTRACT KEY VALUE PAIRS FROM DOCUMENTS: A NOVEL APPROACH

PRAMOD OMPRAKASH GUPTA

STUDENT ID: 1096494

Thesis Report for
Master of Science in Machine Learning & Artificial Intelligence

Liverpool John Moores University & UpGrad

March 2024

DEDICATION

In the labyrinth of academic exploration, this thesis stands as a testament to the unwavering support and inspiration I have received from a few exceptional individuals. To my thesis supervisor, Aayushi Verma, I extend my deepest gratitude for being the guiding force behind this academic endeavor. From the inception of the research journey to the culmination of the final thesis, Aayushi's dedication, expertise, and mentorship have been the bedrock upon which this work was built. Her insightful guidance, constructive criticism, and tireless commitment to excellence have shaped not only the outcome of this thesis but also my growth as a scholar.

To my parents, who have been my pillars of strength throughout my academic journey, I dedicate this work with profound appreciation. Your unwavering support, sacrifices, and belief in my abilities have been the driving force behind my pursuit of knowledge. Your sacrifices and encouragement have shaped not only my academic success but also my character.

To my beloved wife, whose unwavering love and understanding have been a constant source of motivation and strength, I dedicate this thesis with heartfelt gratitude. Your encouragement, patience, and sacrifices have been the cornerstone of my ability to focus on this scholarly pursuit. Your belief in my potential has fueled my determination to overcome challenges and persevere in the face of adversity.

This work is not just a culmination of research; it is a reflection of the collective efforts, sacrifices, and love of those who have stood by me. Each page is infused with the lessons learned, the challenges overcome, and the triumphs shared with these incredible individuals. Their influence is etched into the very fabric of this thesis, and I am forever indebted to them for their unwavering support and belief in my academic journey.

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to Aayushi for her invaluable guidance and unwavering support throughout this thesis journey. Her mentorship, insightful feedback, and encouragement have been instrumental in shaping the course of this research endeavor. Without her expertise and dedication, completing this thesis would have been considerably more challenging.

I am also deeply thankful to Ahmed Kaky for his exceptional lectures and invaluable contributions during my master's journey. His profound knowledge, engaging teaching style, and passion for the subject matter have enriched my learning experience significantly. Each lecture was a source of inspiration and motivation, fueling my academic growth and intellectual curiosity.

Furthermore, I extend my sincere appreciation to Liverpool John Moores University for providing me with the opportunity to pursue my master's degree. The university's commitment to academic excellence and research innovation has laid the foundation for my academic and professional development. I am grateful for the outstanding faculty, resources, and learning environment that have enriched my educational experience.

I would also like to acknowledge Updrad for facilitating the platform that enabled me to undertake my master's program from a prestigious UK university while residing in India. Their seamless support and efficient services have made this cross-continental academic journey possible, allowing me to access quality education and expand my horizons without geographical constraints.

In conclusion, I am indebted to everyone mentioned above for their invaluable contributions, guidance, and support throughout this academic endeavor. Their collective efforts have played a pivotal role in shaping my academic journey and achieving this significant milestone in my educational pursuit.

ABSTRACT

This research proposal embarks on an exploration of document data extraction, emphasizing the transformative potential of Large Language Models (LLMs). The background exposes the limitations of traditional extraction methods and the need for adaptive strategies in the face of diverse document structures. Identified shortcomings in existing methodologies underscore challenges with rigid rule-based systems and the adaptability of machine learning approaches. This research aims to bridge these gaps through a novel approach utilizing LLMs, particularly GPT-3.5, to extract data in structured key-value pairs, enhancing interpretability and utility. The research sets forth two primary objectives: to explore the transformative potential of LLMs in revolutionizing document data extraction and to provide a practical solution for diverse document types. Additionally, the study plans to conduct a comprehensive evaluation of various LLMs, including LLAMA and Google BARD, alongside OpenAI's ChatGPT, offering insights into their effectiveness. Outcomes are expected to include a robust LLM-based extraction system, nuanced understanding of LLM strengths and weaknesses, heightened accuracy, and adaptability across diverse documents. The proposal aspires to contribute significantly to both academic discourse and practical applications, reshaping information extraction methods and paving the way for a new era in document data extraction powered by advanced language models.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI: Artificial Intelligence

API: Application Programming Interface

ASG: Automatic Story Generation

BERT: Bidirectional Encoder Representations from Transformers

BLEU: BiLingual Evaluation Understudy

BLOOM: BigScience Large Open-science Open-access Multilingual Language Model

CBT: Children's Book Test

CC: Common Crawl

CIDEr: Consensus-based Image Description Evaluation

CLM: Causal Language Modeling

CTR: Corrupted Text Reconstruction

EACS: Embedding Average Cosine Similarity

FLAN: Finetuned Language Models

FTR: Full Text Reconstruction

GAN: Generative Adversarial Network

GENCI: Grand Équipement National de Calcul Intensif

GMS: Greedy Matching Score

GPT: Generative Pre-trained Transformer

GPU: Graphics Processing Unit

HANNA: Human-ANnotated NArratives

HTML: HyperText Markup Language

IDRIS: Institute for Development and Resources in Intensive Scientific Computing

IML: Instruction Meta-Learning

L2R: Left-to-Right

LAMA: LAnguage Model Analysis

LaMDA: Language Model for Dialogue Applications

LLM: Large Language Model

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

The realm of document data extraction is in constant flux, compelled by the imperative for adaptive solutions in the face of ever-changing document structures and content variations. Traditional extraction methods, rooted in rule-based systems and predefined templates, encounter inherent limitations when confronted with the dynamic nature of documents emanating from diverse sources. This research seeks to make a significant contribution to the field by delving into the transformative potential of Large Language Models (LLMs), with a particular emphasis on the cutting-edge GPT-3.5 developed by OpenAI.

The emergence of sophisticated language models, notably GPT-3.5, represents a paradigm shift in natural language processing. GPT-3.5's autoregressive architecture excels in generating coherent and contextually aware text, positioning it as a formidable tool for grappling with the intricacies of document data extraction. As evidenced by influential works such as "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" (Devlin et al., 2018) and the foundational White Paper on GPT-3.5, these models have redefined our comprehension of bidirectional language understanding and language generation capabilities.

This research not only builds upon previous studies that scrutinized the shift from rule-based extraction to LLM-based extraction but also introduces a nuanced exploration of the distinctive capabilities of GPT-3.5 in the context of document data extraction. Moreover, it incorporates insights from advanced language models such as Gemini Text and Vision, and LLama2. These models, built on parameters ranging from 7 billion to 70 billion, contribute to the multifaceted landscape of large language models and add a layer of complexity to the evolving field of information extraction from documents. As we delve into the intricacies of GPT-3.5 and its counterparts, the goal is to unravel new possibilities for enhancing the adaptability and effectiveness of document data extraction methodologies.

1.2 Problem Statement

The traditional methods of document data extraction encounter formidable challenges when it comes to adapting to the diverse and intricate landscape of document structures and content. Conventional extraction techniques, reliant on rule-based systems or specific patterns, often prove inadequate in the face of the complexity and variability inherent in documents sourced from entities like the IRS, dummy KYC data, emails, and legal agreements such as ISDA documents. The manual creation and maintenance of extraction rules become increasingly impractical as the number and diversity of document types multiply.

The imperative for a robust and versatile document data extraction system becomes even more apparent in scenarios where documents exhibit unique structures or deviate from predefined patterns. Traditional approaches, lacking the flexibility to seamlessly adapt to new document types, inevitably result in suboptimal extraction accuracy.

The current state of document data extraction research emphatically underscores the need for a paradigm shift from traditional methods to more adaptive and context-aware approaches. The challenges in achieving this shift include seamlessly integrating Large Language Models (LLMs) into existing extraction pipelines, addressing the intricate nuances of fine-tuning for specific document domains, and ensuring scalability and efficiency in handling an ever-expanding array of diverse document types. This study is poised to bridge these formidable gaps by not only exploring the vast potential of LLMs for document data extraction but also formulating innovative strategies to enhance their accuracy and adaptability, ushering in a new era of document data extraction methodologies.

1.3 Aim and Objectives

The primary aim of this research is to revolutionize the field of document data extraction by harnessing the transformative capabilities of Large Language Models (LLMs), specifically focusing on GPT-3.5. The goal is to redefine conventional extraction methods, enhancing accuracy, adaptability, and efficiency in retrieving information from diverse and complex documents.

The research objectives are meticulously formulated based on the overarching aim of the study:

- Contextual Exploration: Investigate and comprehend the intricacies of Large Language Models (LLMs), with a specific focus on GPT-3.5, within the domain of document data extraction.
- Methodological Innovation: Introduce a pioneering methodological approach utilizing GPT-3.5 for the extraction of document data in structured key-value pairs.
- Comprehensive Evaluation: Conduct a thorough evaluation of various LLMs, encompassing LLAMA, Google BARD, and OpenAI's ChatGPT, analyzing their effectiveness in document data extraction.
- Performance Assessment: Evaluate the performance of the developed LLM-based extraction system using a benchmark dataset, employing precision, recall, and F1-score metrics.
- Practical Application: Apply the developed approach to a real-world use case, demonstrating its practical applicability and effectiveness in diverse scenarios within research or industry environments.
- Impact Assessment: Assess the transformative potential of LLMs, specifically their impact on enhancing the efficiency of information retrieval processes.

The envisioned outcomes include not only the establishment of a robust LLM-based extraction system but also a nuanced understanding of the comparative strengths and weaknesses of different LLMs. By reshaping the methods through which information is extracted from documents, this study aspires to contribute significantly to both academic discourse and practical applications, paving the way for a new era in document data extraction powered by advanced language models.

1.4 Research Questions

While the research objectives provide a comprehensive roadmap, the study is guided by specific research questions to further refine the investigation:

- How can the contextual understanding and language generation capabilities of GPT-3.5 be leveraged to enhance document data extraction?

- What innovative methodological approaches can be introduced to optimize the extraction of document data in structured key-value pairs using GPT-3.5?
- How do various LLMs, including LLAMA, Google BARD, and OpenAI's ChatGPT, compare in terms of effectiveness in document data extraction?
- What metrics, such as precision, recall, and F1-score, best capture the performance of the developed LLM-based extraction system when evaluated on a benchmark dataset?
- In what real-world scenarios can the developed approach demonstrate practical applicability and effectiveness?
- What is the transformative impact of LLMs, particularly GPT-3.5, on the efficiency of information retrieval processes in document data extraction?

These research questions serve as guiding pillars, directing the investigation towards a comprehensive understanding of the potential and challenges associated with LLM-based document data extraction.

1.5 Scope of the Study

The scope of this study extends into the multifaceted domain of document data extraction methodologies, with a pronounced emphasis on the utilization of Large Language Models (LLMs) and the paradigm-shifting concept of zero-shot predictions. This research delves comprehensively into the diverse landscape of document types, encompassing but not limited to IRS documents, dummy KYC data, emails, and legal agreements such as ISDA documents. By incorporating these diverse sources, the study aspires to develop a model that transcends the constraints associated with specific document types, showcasing unparalleled adaptability and effectiveness across a broad spectrum.

The investigation further extends to the creation and meticulous analysis of structure templates and instruct documents within the curated datasets. This involves not only defining robust frameworks but also providing illustrative examples to guide the LLM in achieving accurate extraction. This forms a critical aspect of the study's scope, ensuring a nuanced exploration of LLM capabilities in handling various document structures.

Additionally, the scope embraces the intricacies of the evaluation process post zero-shot predictions. This involves a meticulous assessment of accuracy, contextual relevance, and scalability of the developed LLM-based extraction system. The study aims to identify not only the inherent strengths but also the limitations of the system, thereby providing crucial insights into areas of improvement and refinement.

In essence, the expansive scope of this study reaches into the realms of document analysis, natural language processing, and machine learning, offering a holistic exploration of the capabilities and potential applications of LLMs in document data extraction. The study's scope is not confined to a singular facet but rather spans the comprehensive landscape of challenges and opportunities in the realm of document data extraction methodologies.

1.6 Significance of the Study

The significance of this study lies in its capacity to advance the field of document data extraction, particularly within the nuanced realm of Large Language Models (LLMs) and their application in zero-shot predictions. By meticulously exploring and refining the capabilities of LLMs in extracting information from diverse documents, this research contributes significantly to the evolving landscape of natural language processing and document analysis.

The outcomes of this study hold profound implications for industries reliant on accurate and efficient document data extraction, including but not limited to the legal, financial, and administrative sectors. The focus on custom datasets from varied sources, such as the IRS, dummy KYC data, emails, and legal agreements, adds an additional layer of versatility to the findings. The significance of this research is not only in developing a model that adapts to different document types but also in setting a precedent for more effective and adaptable document data extraction systems.

Moreover, the insights generated by this study can inform practitioners, researchers, and developers in refining their approaches to document data extraction. By providing practical solutions to challenges in information extraction from a diverse range of documents, the study paves the way for improved efficiency, accuracy, and applicability across a spectrum of real-world scenarios. In essence, the significance of this research extends to its potential to shape

the future of document analysis, offering tangible solutions to challenges in information extraction.

1.7 Structure of the Study

The structure of this study is meticulously organized to provide a comprehensive exploration of document data extraction methodologies, leveraging the transformative capabilities of Large Language Models (LLMs) and zero-shot predictions. The study unfolds in distinct chapters, each contributing to the overarching narrative and aims of the research.

- Chapter 1: Introduction

  This introductory chapter sets the stage for the entire research endeavor. It begins by establishing the background of the study, shedding light on the challenges within traditional document data extraction methods and the transformative potential offered by LLMs. The problem statement delineates the shortcomings of existing approaches, emphasizing the need for a paradigm shift. The chapter culminates in the delineation of the study's aims and objectives, research questions, and the scope of the investigation.

- Chapter 2: Related Works

  The second chapter delves into the landscape of prior research, offering insights into traditional extraction methods, machine learning approaches, and the transition to LLM-based extraction. It critically examines the problem statement and challenges associated with traditional methods. Drawing upon seminal works, the chapter presents a thorough exploration of the state-of-the-art in document data extraction methodologies.

- Chapter 3: Methodology

  This pivotal chapter details the methodological approach adopted in the study. It provides a comprehensive overview of the experimental design, dataset curation, and the utilization of GPT-3.5. The chapter outlines the steps taken to explore contextual nuances, innovate extraction methods, and evaluate the performance of LLMs. It also elucidates the process of applying the developed approach to real-world use cases.

- Chapter 4: Analysis

  Our focus shifts to a meticulous analysis of the gathered data and the application of the proposed methodology. Section 4.1 initiates the analysis by presenting a comprehensive overview of the datasets utilized, including their inherent fields and volume. This section serves as a contextual backdrop, providing a clear understanding of the input variables crucial to subsequent assessments.

  Following this, Section 4.2 scrutinizes the pre-processing steps undertaken to refine and prepare the data for analysis. By detailing the various steps involved in data cleaning, transformation, and feature extraction, this section lays the groundwork for a robust analytical framework. Each pre-processing step is meticulously explained to ensure transparency and reproducibility.

  Section 4.3 delves into the architectural intricacies of the language model utilized for generating embeddings. This analysis encompasses an in-depth exploration of the model's design, parameters, and underlying mechanisms, elucidating the rationale behind its selection and its relevance to the research objectives.

  In Section 4.4, we turn our attention to the evaluation methods and metrics employed to assess the performance and fairness of the language model. Each metric is explained in detail, emphasizing its significance in gauging the effectiveness of the model in achieving the predefined aims. Comparative analyses and visual representations are included to provide a comprehensive and accessible overview.

  The synthesis of these analyses forms the basis for the subsequent chapters, guiding the reader through the interpretation of results and facilitating a nuanced discussion of the implications derived from the research.

- Chapter 5: Results and Discussion

  The fourth chapter presents the findings and outcomes of the study. It offers a detailed analysis of the performance of the developed LLM-based extraction system, including precision, recall, and F1-score metrics. The chapter provides insights into the adaptability and effectiveness of the approach in real-world scenarios. The results are juxtaposed with other state-of-the-art extraction methods for a comprehensive comparative analysis.

- Chapter 6: Conclution and Recommandation

In Chapter 6, we embark on the final phase of our research journey, culminating in a comprehensive conclusion and offering insightful recommendations for future work. Section 6.1 encapsulates the key findings derived from our analyses, presenting a synthesis of results in alignment with the predefined research objectives. This section aims to provide readers with a lucid understanding of the implications drawn from our study.

Section 6.2 extends beyond summarization, delving into the broader implications of our research within the existing scholarly discourse. By contextualizing our findings within the broader landscape of gender bias in Machine Translation, we contribute to the evolving narrative in this domain. The discussion encompasses the strengths and limitations of our approach, fostering a nuanced comprehension of the significance of our contributions.

In the final subsection, 6.3, we offer recommendations for future research endeavors, identifying avenues for further exploration and refinement. These recommendations are grounded in the insights gained during the course of our study, providing a roadmap for researchers seeking to build upon our work. Through this conclusive chapter, we not only draw a definitive endpoint to our current investigation but also pave the way for continued advancements in the understanding and mitigation of gender bias in Machine Translation.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Named Entity Recognition (NER), the fundamental task of identifying and classifying key entities within text, has undergone a significant evolution over the past three decades. Early research in NER relied heavily on hand-crafted rules and linear models, often tailored to highly specific text domains. The drive for standardization through initiatives like MUC, CONLL, and others pushed the development of supervised learning techniques. Methods such as Hidden Markov Models (HMMs), Maximum Entropy (ME), Support Vector Machines (SVMs), and crucially, Conditional Random Fields (CRFs), emerged as powerful tools for NER. CRFs, with their ability to effectively model dependencies within sequences, became a particularly dominant approach. Developments in CRF models, including the semi-CRF and the incorporation of phrase embeddings, further refined NER accuracy. Moreover, the recognition that interrelated NLP tasks could benefit from joint learning led to the development of multi-task models that outperformed those trained solely on NER. However, the limitations of supervised learning, particularly the reliance on structured annotated data, spurred the development of semi-supervised approaches. These techniques leveraged vast amounts of unlabeled text to address the scarcity of labeled data.

The recent deep learning revolution has profoundly transformed the field of NER and sequence tagging. The ability of deep neural networks to learn complex representations from data has led to significant breakthroughs. This literature review explores the historical trajectory of NER research, with a particular emphasis on the advancements brought about by deep learning methods in sequence tagging.

Additionally, the rise of large language models (LLMs) has introduced the concept of prompt-based extraction. LLMs, with their vast knowledge store and ability to understand complex instructions, can be prompted to reveal the very training data they were built upon. This extraction technique raises important considerations regarding data privacy and the responsible use of LLMs, which will be a noteworthy point of discussion within this review.

2.2 the Evolution of Named Entity Recognition (NER)

Named Entity Recognition (NER) has undergone significant advancements over the past three decades, with researchers continually refining techniques and models to address the challenges of identifying and classifying named entities in unstructured text. The journey begins with early rule-based linear models and progresses through various supervised learning approaches to the emergence of semi-supervised methods.

2.2.1 Rule-Based Linear Models (1980s - Early 1990s):

- The earliest attempts at NER were characterized by hand-crafted rule-based linear models. Rau (1991) made notable contributions by solving part of the NER task. Other efforts in Information Extraction (Besemer and Jacobs, 1987; DeJong et al., 1979; Dyer and Zernik, 1986) laid the foundation for later developments.

2.2.2 Standardization and Annotated Corpora (Mid-1990s - Early 2000s):

- The need for standardization prompted the introduction of benchmarks such as MUC-6 (Grishman and Sundheim, 1996), HUB-4 (Chinchor et al., 1998), and subsequent challenges like MUC-7, MET-2, IREX, CONLL, ACE, and HAREM. These challenges aimed to evaluate and compare NER systems using annotated corpora.

2.2.3 Supervised Learning Techniques (Late 1990s - Early 2000s):

- Supervised learning techniques, trained on large annotated corpora, became dominant. Notable methods include Hidden Markov Models (HMM), Decision Trees, Maximum Entropy Models (ME), Support Vector Machines (SVM), and Conditional Random Fields (CRF). CRF emerged as one of the most effective NER algorithms, addressing issues like label bias present in ME models.

2.2.4 Semi-CRF and Lexicon-Infused Skip-Gram Models (Mid-2000s - 2010s):

- Sarawagi and Cohen (2004) improved the CRF model by introducing the semi-CRF model, assigning labels to subsequences rather than individual entities. Passos et al. (2014) incorporated a lexicon-infused skip-gram model into a log-linear CRF system to enhance performance.

2.2.5 Joint Models and Multitask Learning (2010s):

- Researchers explored joint models of entity analysis, combining tasks such as coreference resolution, named entity recognition, and entity linking. Durrett and Klein (2014) introduced a structured CRF model for multitask learning, demonstrating improved performance on various entity analysis tasks.

2.2.6 JERL Model and Dependency Modeling (2010s):

- Luo et al. (2015) proposed the Joint Entity Recognition and Linking (JERL) model, an extension of the semi-CRF model capturing dependencies between NER and entity linking. This approach aimed at enhancing the contextual understanding of named entities.

2.2.7 Semi-Supervised Learning and Web-Based Data (2010s):

- As structured text limitations for supervised learning became apparent, researchers turned to semi-supervised methods. Suzuki et al. (2011) introduced an unsupervised method leveraging large-scale unlabelled data to create condensed feature representations, addressing the challenges of limited annotated corpora.

The evolution of NER has seen a progression from rule-based models to sophisticated supervised and semi-supervised approaches, with a focus on joint modeling, multitask learning, and leveraging web-based data for contextual information. Each stage of development builds upon the shortcomings of previous methods, leading to more robust and efficient NER systems.

2.3 The Evolution of NER: From Rule-Based to Deep Learning

The rise of deep learning revolutionized NER. Neural networks, like Recurrent Neural Networks (RNNs), demonstrated the power of learning complex features automatically from data. This shift minimized the need for manual rule creation.

The Transformer architecture brought further breakthroughs. Its attention mechanism excels at capturing long-term dependencies in text, crucial for understanding context in NER. Pre-trained language models like BERT provide rich contextualized word representations, leading to substantial improvements in NER accuracy.

Deep learning has made NER systems more accurate than ever before. Ongoing research focuses on further enhancing performance, addressing challenges like nested entities (entities within entities), and making NER more adaptable to new domains and languages.

2.3.1 The Rise of Deep Learning: Neural Networks Take Charge

The advent of deep learning transformed NER. Neural network models could learn complex representations of words and phrases directly from the data, minimizing manual feature engineering. Early deep learning NER models used architectures like Recurrent Neural Networks (RNNs), particularly LSTMs and BiLSTMs, to capture sequential relationships in text.

2.3.1.1 Recurrent Neural Networks (RNNs): The Foundation

A Recurrent Neural Network (RNN) is a type of artificial neural network that differs from traditional feedforward networks. While feedforward networks process inputs and produce outputs without considering previous outputs, RNNs utilize feedback loops, allowing them to learn from past outputs. This enables RNNs to capture temporal dependencies and context in sequential data. Notable applications of RNNs include Apple's Siri and Google's voice search algorithm.

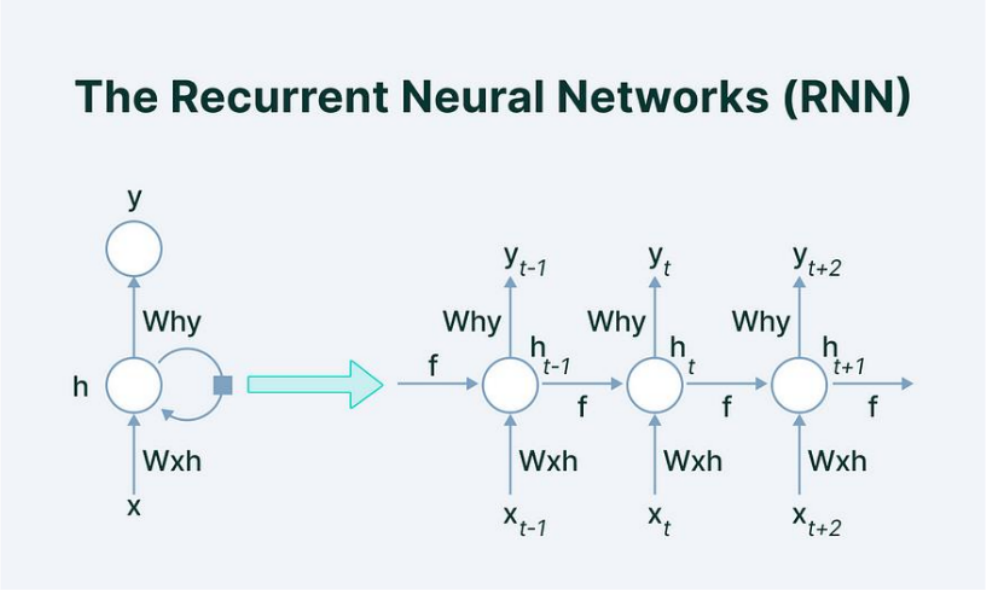


Figure 2.1 RNN Architecture

RNNs have recurrent connections, where the output is fed back into the network, creating a loop. Each node in an RNN functions as a memory cell, retaining information over time. RNNs self-learn through backpropagation, adjusting predictions based on past errors.

- **Core Idea:** RNNs process input sequences (e.g., words in a sentence) one element at a time. They maintain a "hidden state" that acts as a memory of information seen so far.
- **Processing Loop:** At each step, the RNN takes the current input and the previous hidden state. It passes them through a neural network layer to produce a new hidden state, which summarizes the current word and its context.
- **Extraction Use:** For NER, an output layer can be added on top of the hidden state at each time step to predict the entity label for the current word.
- **The Downside: Vanishing Gradients** Simple RNNs struggle to retain information from much earlier in the sequence due to a problem called "vanishing gradients," making them less effective at modeling long-term dependencies in text.

Types of RNNs:

- One to One: Typical feedforward model with one input and one output.
- One to Many: Single input generating multiple outputs, useful in applications like music generation or image captioning.
- Many to One: Multiple inputs producing a single output, employed in sentiment analysis or rating models.
- Many to Many: Multiple inputs and outputs, often used in machine translation.

RNNs, with their ability to incorporate past outputs for improved predictions, and LSTMs, addressing issues like vanishing gradients and long-term dependence, play crucial roles in various machine learning applications, especially those involving sequential data.

2.3.1.2 Long Short-Term Memory Networks (LSTMs): The Memory Upgrade

To address challenges like vanishing gradients and long-term dependence, Long Short-Term Memory (LSTM) networks were introduced. LSTMs enhance memory capabilities, allowing nodes to remember information for an extended period efficiently. LSTMs employ three gates forget gate, input gate, and output gate - to update and regulate cell states. The forget gate decides which information in the cell state should be retained or discarded.

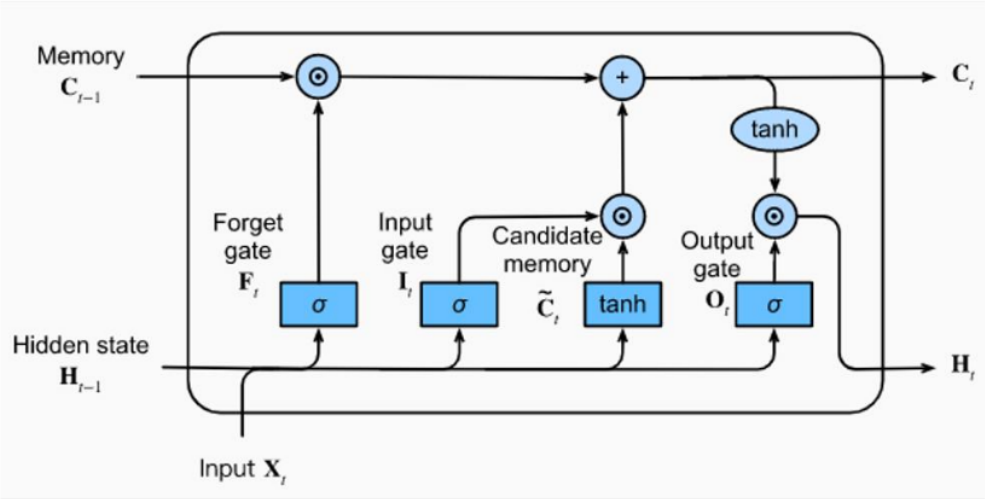


Figure 2.2 LSTM Architecture

- **Gates to the Rescue:** LSTMs address the vanishing gradient issue through special "gates." These gates control when information is added, removed, or retained in the cell state (the "long-term memory"). These include:
    - Forget Gate: Decides what information to remove from the previous cell state.
    - Input Gate: Determines what new information from the current input to store in the cell state.
    - Output Gate: Controls what information from the cell state should make it to the hidden state that becomes input for the next step.
- **Better Recall:** This architecture enables LSTMs to effectively access and utilize past information even if it occurred much earlier in the sequence.

2.3.1.3 Bidirectional LSTMs (BiLSTMs): Understanding Context from Both Sides

A **Bidirectional LSTM**, or **biLSTM**, is a sequence processing model that consists of two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. BiLSTMs effectively increase the amount of information available to the network, improving the context available to the algorithm (e.g. knowing what words immediately follow *and* precede a word in a sentence).

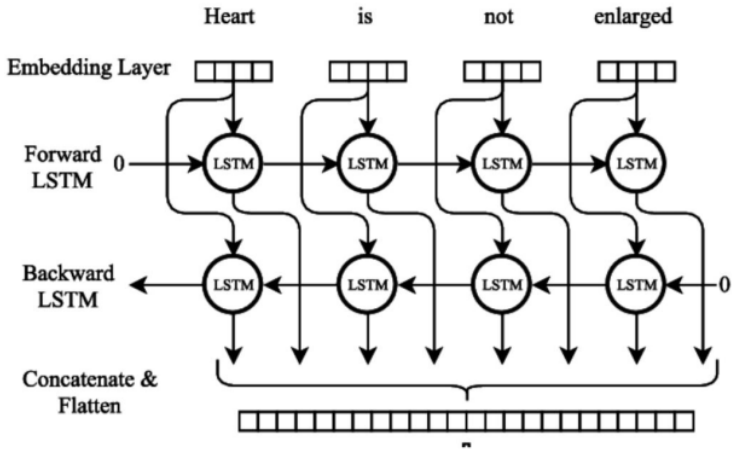


Figure 2.3 Bidirectional LSTM

- **Two LSTMs for the Price of One:** BiLSTMs combine two LSTMs: one processes the sequence in the forward direction, the other in the reverse direction.

- **Seeing the Whole Picture:** This enables the model to consider both past and future contexts when making predictions about a given word
- **NER Advantage:** In NER, BiLSTMs are extremely valuable as they can, for example, use the upcoming word "Ltd." to help infer that the current word "Google" is part of a company name.

In Summary:

- **RNNs** form the basis for sequence processing but can lack long-term memory.
- **LSTMs** enhance memory for long-range dependencies, making them powerful for sequence tasks.
- **BiLSTMs** add the benefit of bidirectional context, further improving performance in tasks like NER.

2.3.2 The Transformer Era: Attention-Based Revolution

The Transformer architecture unleashed a wave of advancements. Its superior ability to model long-range dependencies and its parallelizable nature were significant upgrades over RNNs. Pre-trained language models like BERT, trained on massive text corpora, have become cornerstones of modern NER systems. They provide rich contextual embeddings, significantly improving accuracy. The paper "Attention Is All You Need" (Vaswani et al., 2017) introduced the Transformer architecture, dispensing with recurrence and relying solely on attention mechanisms. This laid the foundation for an explosion of innovation in large language models (LLMs).

- **GPT Series (OpenAI):** OpenAI's Generative Pre-trained Transformer models (GPT, GPT-2, GPT-3) demonstrated remarkable language abilities. GPT-3, with 175 billion parameters, marked a significant leap in scale. However, this series remained closed-source until recently.

  The GPT model series has demonstrated consistent improvement over time:

  - **GPT-1:** Introduced the core Transformer-based architecture.
  - **GPT-2:** Increased both scale and dataset size, leading to better text generation capabilities.

- **GPT-3:** Marked a significant leap in scale with 175 billion parameters, showcasing even more versatile and adaptable language skills.

- **ChatGPT (OpenAI):** Released in 2022, ChatGPT quickly gained popularity due to its exceptional conversational abilities and depth of knowledge. It's based on the GPT-3.5 architecture, specifically fine-tuned for dialogue generation.
    - Optimized for conversational AI, utilizing Reinforcement Learning from Human Feedback (RLHF) for engaging, informative dialogue.
    - Struggles with factual consistency and can be prone to incorrect information.

- **Gemini (Google AI):** Google AI's Gemini family of models stands out for its rigorous training and careful evaluation. Gemini models aim to combine the strengths of prior work, focusing on reliability, informativeness, and safety.
    - Focus on reliability, factual consistency, and reducing harmful biases.
    - Employs human feedback during the fine-tuning process.

- **Gemini Vision:** A notable extension of the Gemini family is Gemini Vision, a multimodal model integrating both text and images. This enables more comprehensive understanding and generation across different forms of content.
    - Multi-modal capabilities allow it to process and reason about text and images simultaneously.
    - Has applications in image-based question answering and generating descriptions for images.

- **LLaMA (Meta AI):** Meta AI's LLaMA series (7B, 13B, 65B, 70B) contributed to the field by making smaller LLMs publicly available. While less powerful than commercial offerings, LLaMA models are an excellent research resource. These models focus on efficient scaling and multilingual capabilities.
    - Open-source and accessible to the research community.
    - Smaller models enable broader experimentation and customization.
    - Demonstrates strong performance at lower parameter counts than commercially dominant models.

2.3.2.1 The Future of Transformer-based LLMs

The evolution of LLMs is far from over. Research is ongoing to address:

- **Factuality and Bias:** Improving reliable knowledge retrieval and grounding for better accuracy and minimizing the generation of harmful content.
- **Multimodality:** Extending capabilities to combine text, speech, code, images, and other modalities into unified models.
- **Efficiency:** Developing models that are more computationally efficient while maintaining or exceeding performance levels.

2.3.3 Prompt Engineering: The Art of Input

Prompt engineering involves designing carefully crafted input text (prompts) that coax the desired behavior from large language models. In the case of JSON extraction, this translates to:

- **Natural Language Instructions:** Providing clear descriptions of the JSON structure and the data you want to extract.
- **Examples:** Giving the model a few input-output pairs to quickly grasp the extraction task.
- **Clear Formatting:** Potentially structuring prompts in ways that make it easier for the model to understand the relationship between the JSON and the desired output (e.g., using tables or delimiters).

2.3.3.1 Key Research Areas

- **Instruction-following in LLMs:** Research aims to improve how LLMs follow complex instructions provided in natural language. This directly impacts the effectiveness of JSON extraction prompts.
- **Few-Shot and Zero-Shot Learning:** Exploring how LLMs can learn new tasks (like specific JSON extraction) with limited or no labeled examples.
- **Prompt Design Techniques:** Developing systematic methods for constructing effective prompts. This area is still evolving, especially regarding JSON extractors.

- **Core Prompt Engineering Techniques**

    1. **Clear and Explicit Instructions:**
        - Describe the task to the language model as precisely as possible. Avoid ambiguity.
        - **Example:** "Extract the customer's first name, last name, and email address from the following JSON data:"

    2. **Input-Output Examples:**
        - Provide a few examples of JSON snippets and their corresponding desired outputs. This helps the model quickly grasp the pattern.
        - **Example:** Include multiple JSON variations to demonstrate edge cases.

    3. **Zero-Shot vs. Few-Shot:**
        - **Zero-Shot:** Rely solely on task description with no examples
        - **Few-Shot:** Provide one or more input-output examples.
        - Choose the method based on the task complexity and the availability of labelled data.

    4. **Template-Based Prompts:**
        - Create prompts with slots to be filled by specific data from the JSON. This can help the model generalize better.
        - **Example:** "Extract the value of the [field_name] field from the JSON input."

    5. **Iterative Refinement:**
        - Start with a basic prompt and analyze the outputs. Tweak the wording, add more examples, or adjust the format based on the model's responses.

2.3.3.2 Key Developments in Deep Learning NER

- **Joint Entity and Relation Extraction:** Models that can simultaneously identify entities and the relationships between them, improving the overall understanding of complex text.
- **Domain-Specific NER:** Fine-tuning pre-trained models on domain-specific text (medical, scientific, technical) for better performance in specialized areas.
- **Multilingual NER:** Models that can handle multiple languages, broadening their applicability.
- **Zero-shot or Few-shot NER:** Methods that can identify entities with little or even no labeled training examples, making NER more adaptable.

2.3.3.3 Benefits of Deep Learning for NER

- **Improved Accuracy:** Deep learning models consistently outperform traditional approaches, leading to more precise entity identification.
- **Reduced Feature Engineering:** These models learn features directly from the data, decreasing reliance on human-designed input.
- **Generalizability:** Pre-trained models can be adapted to new domains with relative ease.
- **Handling Ambiguities:** Contextual representations from deep learning models help understand ambiguous terms better than rule-based systems.

2.3.4 Challenges in Prompt Engineering

While prompt-based learning offers immense promise, it faces several hurdles researchers are actively addressing. Let's outline some key areas where improvement is needed:

2.3.4.1 Designing Effective Prompts

- **Task Complexity:** Prompt design becomes particularly complex for tasks like information extraction and text analysis, in contrast to its relative success in text classification or generation. Strategies for reformulating tasks or novel output engineering may be required (Aghajanyan et al., 2021).

- **Structured Data:** Expressing structured information (tables, graphs, etc.) within prompts is a challenge. Initial work using HTML-like structures exists (Aghajanyan et al., 2021), but more research is needed.
- **Template and Answer Optimization:** Performance hinges on both prompt templates and answers. Finding the optimal combination through search or simultaneous learning needs refinement (Gao et al., 2020c; Shin et al., 2020; Hambardzumyan et al., 2021).

2.3.4.2 Answer Engineering

- **Classification Tasks:** Large numbers of classes complicate answer space selection. Multiple token answers introduce difficulties in language model decoding (Jiang et al., 2020b).
- **Text Generation:** Semantically equivalent outputs may differ syntactically. Guiding the learning process with multiple references is an active research area.

2.3.4.3 Tuning Strategy Selection

- **Prompt vs. Model Parameters:** Tuning prompts, language model parameters, or both remains an area of research. Understanding the pros and cons of each approach is crucial.

2.3.4.4 Harnessing Multiple Prompts

- **Ensembling:** Using multiple prompts can improve outcomes but increases complexity. Techniques to distill knowledge from them are needed (Schick & Schütze, 2020a , 2020b, 2021).
- **Composition and Decomposition:** Breaking complex tasks into sub-prompts is an area of study. When to choose composition vs. decomposition requires further understanding.
- **Augmentation:** Effective sample selection and ordering for prompt augmentation, particularly with input length constraints, is a research focus.

- **Prompt Sharing:** Extending prompt learning across domains, languages, or tasks calls for the development of strategies to customize prompts or manage their interactions.

2.3.4.5 Pre-trained Model Selection

- **Optimal Choice:** Guidance on selecting the most suitable pre-trained language model (PLM) for a given prompt-based task is lacking. Systematic comparative studies are needed.

2.3.4.6 Prompt Transferability

- **Model-Specificity:** Understanding the degree of a prompt's model-specificity will enhance transferability. Research shows prompts transfer effectively across similar-sized models with sufficient validation data, but not in true few-shot settings or with vast model size disparities (Perez et al., 2021).

2.3.4.7 Combining Pre-training Paradigms

- **Synergy:** The success of prompt-based learning partially stems from PLMs trained via the pre-train and fine-tune paradigm. Whether this should be adapted for prompt-based learning or if entirely new pre-training methods are needed requires investigation.

2.3.4.8 Calibration

- **Reliable Probabilities:** PLMs may generate biased responses due to recency bias, majority label bias, or common token bias (Zhao et al., 2021). Techniques to calibrate prompt-based methods for accurate prediction probabilities are crucial.

The Future: The evolution of NER with deep learning is far from over. Research pushes the boundaries to handle nested entities, incorporate knowledge graphs, and address challenges in noisy text.

2.4 Summary

Named Entity Recognition (NER), the task of identifying and classifying key information within text, has undergone a remarkable evolution. Early approaches relied heavily on hand-crafted rules and linear models. Initiatives like MUC, CONLL, and others drove standardization and the development of supervised learning techniques such as Hidden Markov Models, Support Vector Machines, and notably, Conditional Random Fields (CRFs). CRFs excelled at modeling sequential dependencies, becoming a dominant approach for NER. Further refinements included the semi-CRF and the incorporation of lexicon-infused models for performance boosts.

To address the limitations of supervised learning, semi-supervised methods were developed, leveraging vast amounts of unlabeled data. Researchers also explored joint models and multi-task learning, demonstrating that interrelated NLP tasks could benefit from being trained together.

The deep learning revolution has transformed NER. Recurrent Neural Networks (RNNs), including LSTMs and BiLSTMs, were introduced to capture long-range dependencies in text. However, the Transformer architecture has been a game-changer, dispensing with recurrence and utilizing attention mechanisms to surpass RNNs. Pre-trained language models (LLMs) like BERT and GPT-3 provide rich contextual representations, significantly improving NER systems.

Prompt engineering, where carefully designed inputs guide LLMs towards desired outputs, has become a vital area of research. Techniques like providing natural language instructions, examples, and template-based prompts are key to effective prompt design for tasks like JSON extraction. Ongoing research addresses challenges in prompt design, transferability, model selection, and calibration to ensure reliable and accurate results.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

In its entirety, our research pipeline meticulously navigates through five crucial stages, each contributing to the seamless extraction of structured information from unstructured data. These stages encompass Optical Character Recognition (OCR), chunking, document classification, prompt generation, and Large Language Model (LLM) inference and decoding. This comprehensive approach is designed to optimize the efficiency and accuracy of the information extraction process.

Our research adopts a strategic approach, beginning with the creation of a well-crafted prompt to guide the LLM. Leveraging the flexibility and user-friendly nature of LLMs, we forego the need for extensive model training. Instead, we cleverly prompt the LLM with unstructured text, allowing it to generate structured outputs. Our emphasis lies in enforcing the LLM to produce valid JSON, facilitating easy integration with Python through tools like Function cal. This ensures that the output aligns with our predefined data structure, enhancing interpretability and usability.

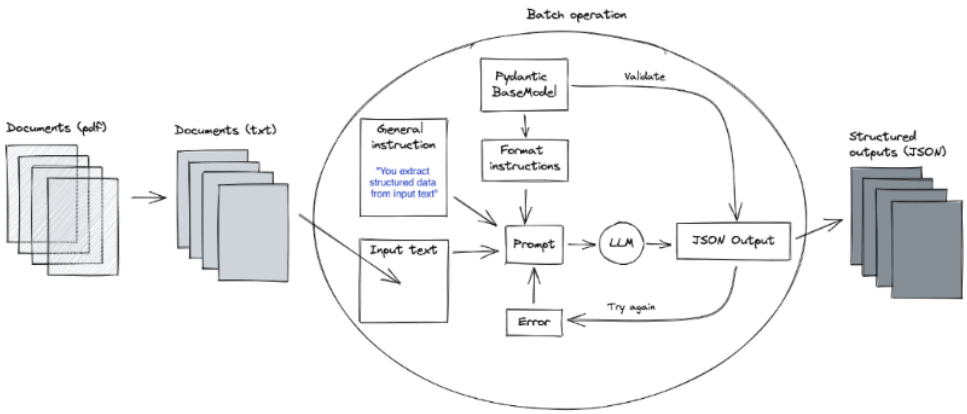


Figure 3.1 Request flow of LLM based Extraction (Yke Rusticus 2023).

Going beyond the immediate application, our research proposes a generalized solution applicable to diverse use cases requiring structured information extraction from unstructured data. We extend the functionality to accommodate PDF documents, a common starting point for text-related tasks. The solution presented here offers adaptability to various data sizes, efficiently processing both small and large documents. To enhance the usability of the approach, we recommend additional processing steps for more extensive PDFs. This broader perspective on generalization showcases the scalability and versatility of our proposed methodology.

3.2 Methodology

3.2.1 Data Selection

Custom datasets derived from diverse sources, including IRS forms such as W-9, W-8BEN, and W-2, will be integral to this research. These datasets will be categorized specifically into structured documents, focusing on standardized forms. The structured documents will be further divided into two primary components: structure templates and instances.

Structure templates will define the expected format and organization of data within each document category, emphasizing the standardized layout of forms like W-9, W-8BEN, and W-2. These templates serve as blueprints for the model, providing a structured framework for information extraction.

Instances, on the other hand, refer to specific examples or instances derived from diverse sources of the structured documents. These instances play a crucial role in guiding the model on how to interpret and extract information according to the predefined structure templates. The use of instances ensures that the model learns from real-world examples, enhancing its ability to generalize and adapt to a wide range of structured document types and formats.

By focusing on structured documents like W-9, W-8BEN, and W-2 forms, this research aims to create a comprehensive and specialized training set. This approach ensures that the model becomes proficient in extracting information from standardized forms commonly used in contexts such as tax reporting and financial transactions.

3.2.2 Data Preparation

In the realm of document data extraction, the significance of structured forms such as W-2, W-8, and W-9 cannot be overstated. These forms, commonly used for wage and tax reporting, verification of non-US taxpayers, and confirmation of taxpayer information, respectively, play a crucial role in various financial and administrative processes. However, due to the confidential nature of these forms, obtaining real-world datasets for research purposes presents a challenge. To overcome this hurdle, this thesis employs a meticulous approach to generate a proprietary dataset using Python packages, particularly the "fillpdf" and "pypdf" libraries.

3.2.2.1 Utilizing the fillpdf Package

The "fillpdf" package stands as a valuable tool in simplifying the process of handling and manipulating PDFs, especially when it comes to filling and flattening forms programmatically. Recognizing the need for an efficient solution to work with PDFs in Python, the author developed this library to address the challenges of writing, flattening, and editing PDFs. The package leverages the capabilities of pdfrw2, a forked version of pdfrw, and offers several functionalities crucial for this thesis:

- Filling PDFs: The package facilitates the automation of filling PDF forms, a key requirement for generating varied datasets.
- Listing Fields in PDFs: It allows for the extraction of field information within PDF documents, providing insights into the structure of the forms.
- Flattening PDFs: By converting editable PDFs into non-editable ones, the package ensures that the generated datasets maintain the intended structure.
- Inserting Images and Text: This feature is essential for enhancing the realism of the generated datasets, mimicking real-world scenarios.
- Rotating PDFs and Placing Images: These capabilities contribute to the versatility of the package, enabling the creation of diverse datasets.

3.2.2.2 Leveraging pypdf Package:

The "pypdf" package, a free and open-source pure-Python PDF library, complements the efforts of the "fillpdf" package. It offers a range of functionalities that are pivotal for document manipulation and analysis:

- Splitting, Merging, and Cropping PDFs: These operations provide flexibility in handling the pages of PDF files, allowing for the creation of diverse datasets with varied structures.

- Transforming Pages: The package facilitates the transformation of PDF pages, supporting the adaptation of forms to different layouts.

- Adding Custom Data and Passwords: Customizing PDFs with additional data and security features is crucial for simulating real-world scenarios in the generated datasets.

- Retrieving Text and Metadata: Extracting textual content and metadata from PDFs ensures that the generated datasets are rich in information.

3.2.3 OCR

Optical Character Recognition (OCR) is a crucial step in converting non-machine-readable documents like PDFs or images into text, enabling further analysis by computer systems. In our research methodology, OCR plays a vital role, acting as a bridge between visual content and machine-interpretable data.

Initiating the process, we utilize an off-the-shelf OCR service, a technology designed to recognize and extract textual information from document images. This service comprehensively analyzes the document image, identifying individual words and line segments while preserving their spatial context through bounding boxes. The spatial information within these bounding boxes is essential for maintaining the layout and structure of the original document. A detailed example of the output from this OCR stage for a specific document is illustrated in Appendix A.6, demonstrating the effectiveness of the process.

In our research, we specifically leverage the Tesseract open-source OCR engine for its accuracy and efficiency in recognizing text from images. However, in the broader landscape of OCR tools, various alternatives cater to diverse needs. Adobe Acrobat OCR excels in

handling complex documents, ABBYY FineReader is known for advanced text recognition, and Google Cloud Vision OCR, Azure OCR, and AWS OCR offer robust cloud-based solutions.

The selection of an OCR tool depends on project requirements, such as document complexity, language diversity, and processing speed. In our research, we consider factors like cost, speed, ease of use, and adaptability. Tesseract, being open-source, aligns with our research objectives, but other tools like Google OCR, Azure OCR, and AWS OCR provide competitive alternatives. Our final choice will be based on a balance of factors, ensuring the selected OCR tool is not only proficient in document analysis but also aligns with our research priorities, such as being cost-effective, fast, and user-friendly.

3.2.4 Chunking

To address the challenge of processing arbitrarily long documents within the constraints of LLMs' limited input token length, we employ a chunking strategy. Initially, the document is segmented into individual pages. Subsequently, we iteratively trim the last line segments, originating from OCR, until the prompt associated with each chunk is within the LLM's maximum input token limit. The removed lines are then grouped to form a new document page, and this process is repeated until all chunks adhere to the LLM's input token limit. This approach yields N chunks. The choice to initially partition the document by page is informed by the observation that entities seldom span page boundaries, ensuring minimal impact on the final extraction quality.

3.2.5 LLM Function

3.2.5.1 Introduction to Function Calling

To enhance the interpretability and functionality of Large Language Models (LLMs), a novel feature known as Function Calling is incorporated into the research methodology. This innovative approach involves the creation of custom functions tailored to specific tasks. The example below demonstrates the utilization of this feature with OpenAI's GPT-3.5 Turbo

model. Notably, this concept is applicable to various LLMs such as Google Gemini, LLama2, and others.

3.2.5.2 Custom Function Definition

A crucial step in the Function Calling process is the definition of custom functions. These functions encapsulate specific functionalities and parameters, providing a structured interface for LLMs to comprehend and execute tasks effectively. Below is a sample Python function designed for extracting student information from text:

```python
student_custom_functions = [
    {
        'name': 'extract_student_info',
        'description': 'Get the student information from the body of the input text',
        'parameters': {
            'type': 'object',
            'properties': {
                'name': {
                    'type': 'string',
                    'description': 'Name of the person'
                },
                'school': {
                    'type': 'string',
                    'description': 'The university name.'
                },
                'grades': {
                    'type': 'integer',
                    'description': 'GPA of the student.'
                },
                'club': {
```

```
                'type': 'string',
                'description': 'School club for extracurricular activities. '
            }
        }
    }
}
]```

The provided JSON represents a function definition in a format that is commonly used for specifying custom functions within a system or application, particularly in the context of interacting with Large Language Models (LLMs) like ChatGPT. Let's break down each key component:

1. `name`: This field specifies the name of the function, in this case, 'extract_student_info'. This is a unique identifier for the function, allowing the system to reference and invoke it.

2. `description`: The description field provides a brief explanation of the functionality of the function. In this example, the function is designed to "Get the student information from the body of the input text."

3. `parameters`: This section defines the input parameters expected by the function.

- `type`: It specifies the data type of the parameter, and in this case, it is an 'object'. This implies that the function expects an input structured as an object, which is a collection of key-value pairs.

- `properties`: It defines the specific properties (or fields) that the object should contain.

- `name`: This property represents the name of the person and is of type 'string', indicating that it expects a textual value. The 'description' field provides additional information about the purpose of this property.

- `school`: Similar to 'name', 'school' is another property of type 'string', expecting the name of the university. The 'description' field provides context about its purpose.

- `grades`: This property is of type 'integer', indicating that it expects a numerical value representing the GPA of the student. The 'description' field clarifies the nature of this property.
```

- `club`: The 'club' property is again of type 'string' and expects information about the school club for extracurricular activities.

In summary, this JSON defines a function named 'extract_student_info' that takes an object as input with specific properties (name, school, grades, and club). This function is designed to process textual input and extract information related to a student, including their name, university, GPA, and involvement in extracurricular activities. Such a function could be used as part of a system that interacts with an LLM to extract structured data from unstructured text.

3.2.5.3 Function Integration in LLM Inference

Once the custom function is defined, it is integrated into the LLM inference process. The following code snippet illustrates how to use the function with OpenAI's GPT-3.5 Turbo model to generate responses for two student descriptions:

```python
student_description = [student_1_description, student_2_description]
for i in student_description:
    response = client.chat.completions.create(
        model='gpt-3.5-turbo',
        messages=[{'role': 'user', 'content': i}],
        functions=student_custom_functions,
        function_call='auto'
    )
    # Loading the response as a JSON object
    json_response =json.loads(response.choices[0].message.function_call.arguments)
    print(json_response)
```

In this high-level explanation, a Python script is presented to interact with the GPT-3.5 Turbo model using OpenAI's Chat API. The script involves a loop over a list of student descriptions, where each description represents information about a student. For each student description, a

request is made to the GPT-3.5 Turbo model through the OpenAI Chat API. The request includes the user's message, represented by the student description, and a set of custom functions named 'student_custom_functions.' These functions are predefined and assist in structuring the information extracted by the model.

Once the response is received from the GPT-3.5 Turbo model, it is loaded as a JSON object. This JSON object represents the function call made by the model and contains information extracted from the student description. Specifically, the extracted information includes details such as the student's name, major, university, GPA, and club affiliation. This structured data is then printed for analysis or further processing. Overall, the script demonstrates a practical application of leveraging advanced language models to extract specific information from unstructured text, offering a glimpse into the potential of AI-assisted data extraction in real-world scenarios.

3.2.5.4 Output Analysis

The generated JSON output demonstrates the uniformity and consistency achieved through the Function Calling approach. Each response contains structured information about the extracted student details, including name, major, school, grades, and club affiliation. Notably, numeric grades are represented consistently as integers, ensuring a standardized and reliable output. This consistency is paramount for the development of bug-free AI applications and contributes to the overall effectiveness of document data extraction methodologies powered by LLMs.

Sample Output JSON:

```json
{'name': 'David Nguyen', 'major': 'computer science', 'school': 'Stanford University', 'grades': 3.8, 'club': 'Robotics Club'}

{'name': 'Ravi Patel', 'major': 'computer science', 'school': 'University of Michigan', 'grades': 3.7, 'club': 'Chess Club'}
```

This section of the research methodology outlines the integration of Function Calling as a pivotal step in leveraging LLMs for document data extraction, ensuring consistency and reliability in the generated outputs.

3.3 Proposed Method

Document data extraction, especially in unstructured environments, has witnessed a paradigm shift with the advent of Large Language Models (LLMs). In this proposed method, we aim to leverage the transformative capabilities of LLMs for efficient and accurate extraction of information from diverse documents. Our focus is on achieving structured outputs in JSON format, enhancing interpretability and facilitating seamless integration with downstream applications.

3.3.1 LLM-Based Extraction and JSON Output

The foundation of our proposed method lies in harnessing LLMs to process unstructured text and generate structured outputs in JSON format. JSON (JavaScript Object Notation) offers a versatile and standardized way to represent structured information. For our target field output, we define a Pydantic BaseModel, specifying the desired structure. This includes key topics, such as "quality," "price," and "shipping," each represented as literals in the JSON. The aim is to ensure that the LLM outputs align with this predefined structure, providing a standardized and easily interpretable format for extracted information.

3.3.2 Exploration of Different LLM Models

In the dynamic landscape of Large Language Models (LLMs), our research places a particular emphasis on evaluating the efficacy of diverse models, each distinguished by its unique architecture and specialized capabilities.

3.3.2.1 GPT-3.5 (Generative Pre-trained Transformer 3.5)

Developed by OpenAI, GPT-3.5 stands as a groundbreaking autoregressive language model. Its exceptional prowess stems from extensive pre-training on a rich tapestry of diverse datasets, enabling it to capture intricate contextual relationships and nuances in language. GPT-3.5's remarkable proficiency in contextual understanding and language generation positions it as a formidable candidate for document data extraction tasks. Its autoregressive architecture empowers it to generate coherent and contextually aware text, making it a powerhouse in processing unstructured information within various document structures.

3.3.2.2 Google Gemini Pro 1.0 and Gemini Vision

Google's foray into the realm of Large Language Models manifests in the Gemini series, which seamlessly integrates advanced language understanding and visionary capabilities. Gemini Pro 1.0 and Gemini Vision represent a dual-pronged approach to tackling text-based and vision-related tasks. While Gemini Pro 1.0 focuses on text-based challenges, showcasing its proficiency in language-related tasks, Gemini Vision extends its ambit to encompass vision-related information extraction. This duo reflects Google's commitment to a holistic approach, recognizing that language models need to comprehend both textual and visual content for comprehensive understanding and extraction.

3.3.2.3 LLAMA2 Models (7b, 13b, 70b)

The Large Language Model Archive (LLAMA) contributes significantly to our exploration by offering models of varying sizes, ranging from 7b to 70b in parameters. Developed by Meta, these models cater to a spectrum of requirements, with the size of the model impacting its ability to discern nuanced patterns and context. Smaller models, such as 7b, may offer computational efficiency for specific tasks, while larger counterparts, like the 70b model, exhibit potential for capturing intricate details in diverse documents. However, the trade-off involves increased computational requirements for larger models. This selection of LLAMA2 models provides our research with the flexibility to adapt to specific extraction needs, considering both performance and resource constraints.

3.3.3 Advantages of Different Models

Each LLM considered in our research presents distinct advantages, influencing their suitability for specific tasks.

- GPT-3.5: Known for its extensive pre-training on diverse datasets, GPT-3.5 excels in capturing complex contextual relationships. Its autoregressive architecture allows it to generate coherent and contextually aware text, making it particularly effective for document data extraction tasks with varied structures.
- Google Gemini Pro 1.0 and Gemini Vision: The Gemini models from Google integrate both language understanding and vision capabilities. This dual functionality broadens their applicability, enabling comprehensive extraction tasks that involve textual and visual information.
- LLAMA2 (7b, 13b, 70b): The LLAMA models, with varying sizes, offer scalability in terms of performance. Smaller models may be computationally more efficient for certain tasks, while larger models can capture intricate details in diverse documents. The choice depends on the specific requirements and available computational resources.

3.3.4 Model

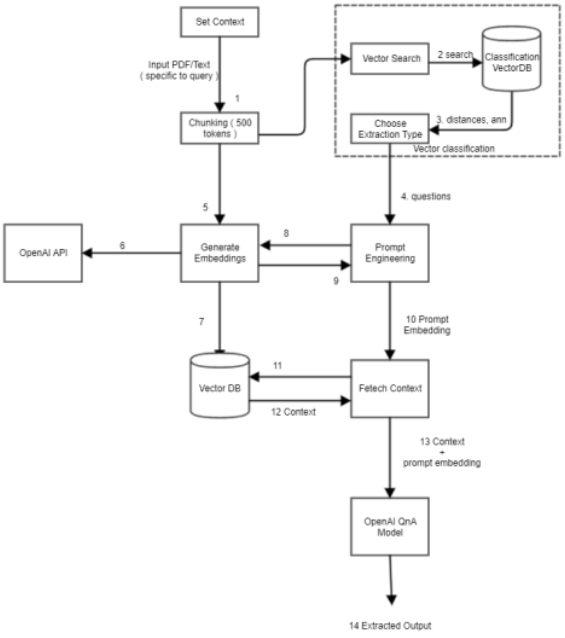


Figure 3.2 Architecture of LLM based extraction

After categorizing the document, we retrieve a corresponding prompt template containing essential elements like the document context, a standardized task description, and a structured schema representation. This template becomes the basis for constructing a specific prompt tailored for the Large Language Model (LLM). The crafted prompt is meticulously designed to incorporate the document context, a detailed task description outlining the entities for extraction, and a schema representation defining the extraction structure. Subsequently, this carefully constructed prompt is sent to the LLM through the chatbot API. In our research, we've opted to utilize OpenAI's ChatGPT for both generating embeddings and executing the extraction process.The LLM then processes the prompt, leveraging its extensive knowledge to comprehend and execute the extraction task. The resulting output undergoes evaluation against the predefined schema representation, ensuring alignment with the specified structure and criteria. This methodology establishes a systematic and personalized interaction with the LLM, ensuring precise and contextually relevant document data extraction within the scope of our research.

Additionally, in this research endeavor, we plan to evaluate the performance of other LLMs such as LLAMA or Google BARD alongside OpenAI's ChatGPT. This comparative analysis aims to provide insights into the effectiveness and suitability of different LLMs for the document data extraction task.

3.3.5 Evaluation:

After conducting zero-shot predictions, an effective evaluation approach is crucial to assess the model's performance. Here is a general framework you can consider:

3.3.5.1 Ground Truth Comparison:

- Maintain a ground truth dataset with manually annotated correct extractions for a subset of documents. Conduct a comparison between the model's predictions and the ground truth to evaluate accuracy and pinpoint any discrepancies. This approach aligns with the guidance provided in a paper by (Garg et al. 2022), which offers a comprehensive overview of entity extraction techniques and applications, encompassing diverse methods for evaluating the performance of entity extraction models.

3.3.5.2 Metric Selection

- Select evaluation metrics tailored to the nature of your task, with precision, recall, and F1-score being common choices for entity extraction tasks. This approach aligns with the insights provided in a referenced paper, which discusses the relevance of these metrics for named entity recognition (NER) tasks and their applicability to zero-shot extraction tasks involving entity recognition. Additionally, the paper explores various evaluation metrics in the broader context of natural language processing (NLP), emphasizing the importance of choosing metrics based on the specific task and desired information.

3.3.5.3 Entity-level Evaluation:

- Conduct a thorough assessment of the model's performance on individual extracted entity types, calculating precision, recall, and F1-score for each entity type to discern specific strengths and weaknesses. This paper delivers an in-depth exploration of entity recognition and classification techniques, incorporating diverse evaluation metrics and their practical applications. The evaluation process, focused on each extracted entity type, facilitates the identification of specific areas of strength and weakness in the model's performance (Ward et al., 2011).

3.3.5.4 Error Analysis:

- Conduct a thorough error analysis to understand common mistakes made by the model.
- Identify patterns in misclassifications and consider refining the training data or adjusting the model architecture accordingly.

3.3.5.5 Scalability Testing:

- Evaluate the model's performance on varying document lengths and complexities to ensure scalability in real-world scenarios.

3.3.5.6 Continuous Monitoring:

- Implement continuous monitoring of model performance over time. This can include retraining the model with new data to adapt to evolving patterns and challenges.

CHAPTER 4

ANALYSIS

4.1 Introduction

The heart of our thesis lies within Chapter 4, where we embark on a comprehensive analysis of the automated form-filling methodology powered by Large Language Models (LLMs) and the innovative use of imaginary Personally Identifiable Information (PII) data. This pivotal chapter delves into the intricacies of our approach, aiming to unravel the multifaceted impact and effectiveness of our novel document generation system.

To begin, the chapter meticulously dissects the results of our automated form-filling methodology, examining the generated samples across various document titles such as W-2, W-8, and W-9. Each category of forms undergoes a detailed examination to assess the system's proficiency in populating editable PDFs with diverse and representative samples. We explore the nuances of the PyPDF Python package in conjunction with LLMs, shedding light on the synergistic relationship that streamlines the generation of 50 samples for each form category.

Furthermore, the analysis extends beyond the technical intricacies to encompass the broader implications and advantages of our methodology. We scrutinize the efficiency gains achieved through the automation of traditionally labor-intensive processes, emphasizing the reduction in manual effort associated with form completion. The chapter elucidates how the integration of LLMs and imaginary PII data not only enhances efficiency but also upholds ethical standards, ensuring the non-confidential nature of the generated samples.

Additionally, the chapter is a platform for exploring the adaptability and flexibility of LLMs in the context of document generation. Through a careful examination of the diverse outputs and variations in the generated samples, we highlight the capacity of LLMs, particularly exemplified by ChatGPT, to dynamically respond to prompts and contextual nuances. This adaptive nature contributes to the system's robustness and positions it as a valuable tool for nuanced document processing.

Analysis is a journey through the intricate details of our innovative methodology, offering a panoramic view of its technical efficacy, efficiency gains, and ethical considerations. As we navigate through the analysis, we unravel the potential impact of our research on the landscape of document generation and pave the way for future advancements in this dynamic field.

4.2 Dataset Description

Our research hinges on the utilization of custom datasets meticulously curated from diverse sources, with a particular emphasis on structured documents originating from IRS forms like W-9, W-8BEN, and W-2. This dataset forms the bedrock of our analysis, providing the model with a rich and varied training ground for information extraction.

4.2.1 Categories and Sources:

The datasets are strategically categorized into structured documents, primarily focusing on standardized forms, to ensure a targeted and specialized training environment. These structured documents are drawn from an array of sources, including IRS forms, renowned for their prevalence in tax-related contexts and financial transactions.

4.2.2 Components of the Dataset:

Within the realm of structured documents, our dataset comprises two pivotal components: structure templates and instances. Structure templates define the expected format and organization of data within each document category, elucidating the standardized layout found in W-9, W-8BEN, and W-2 forms. These templates serve as architectural blueprints, guiding the model in its extraction endeavors.

Conversely, instances represent specific examples derived from diverse sources of the structured documents. These instances play a crucial role in shaping the model's understanding, showcasing real-world scenarios and variations encountered in forms like W-9, W-8BEN, and W-2. The inclusion of instances is instrumental in augmenting the model's

adaptability, allowing it to generalize its learning to accommodate a myriad of document types and formats.

4.2.3 Form-Specific Considerations:

Each IRS form, namely W-9, W-8BEN, and W-2, holds distinct significance in the dataset. The W-9 form, commonly used for tax reporting, contributes insights into information extraction from entities providing services. W-8BEN, with its focus on foreign entities, enriches the dataset with examples from an international context, broadening the model's scope. Finally, the W-2 form, pivotal in employee wage reporting, imparts nuances relevant to financial transactions and employment-related data.

4.2.4 Document Form Fields

The dataset encapsulates various form fields inherent to W-9, W-8BEN, and W-2 forms. These fields, ranging from personal details to financial information, constitute the foundation of our model's learning. For instance, the W-9 includes fields such as name, address, and taxpayer identification number (TIN). W-8BEN focuses on fields related to foreign status and beneficial ownership. Simultaneously, the W-2 encompasses fields crucial for wage reporting, such as income and tax withholdings.

4.2.4.1 W-9 Form

The W-9 form is a critical document for tax reporting in the United States, particularly in business transactions. It is typically provided by individuals or entities that offer services, such as independent contractors or freelancers.

The form includes several key fields:

1. Name: The legal name of the individual or entity providing services.
2. Business Name: If applicable, the name of the business entity associated with the services.
3. Address: The mailing address associated with the individual or business.
4. Taxpayer Identification Number (TIN): This can be a Social Security Number (SSN) for individuals or an Employer Identification Number (EIN) for businesses.
5. Exemptions: Any exemptions claimed by the individual or business to reduce the amount of tax withheld.

Figure 4.1 W9 Form

4.2.4.2 W-8BEN Form:

The W-8BEN form is utilized for individuals who are not U.S. citizens but earn income from U.S. sources. It is crucial for establishing foreign status and beneficial ownership for tax purposes.

Key fields in the W-8BEN form include:

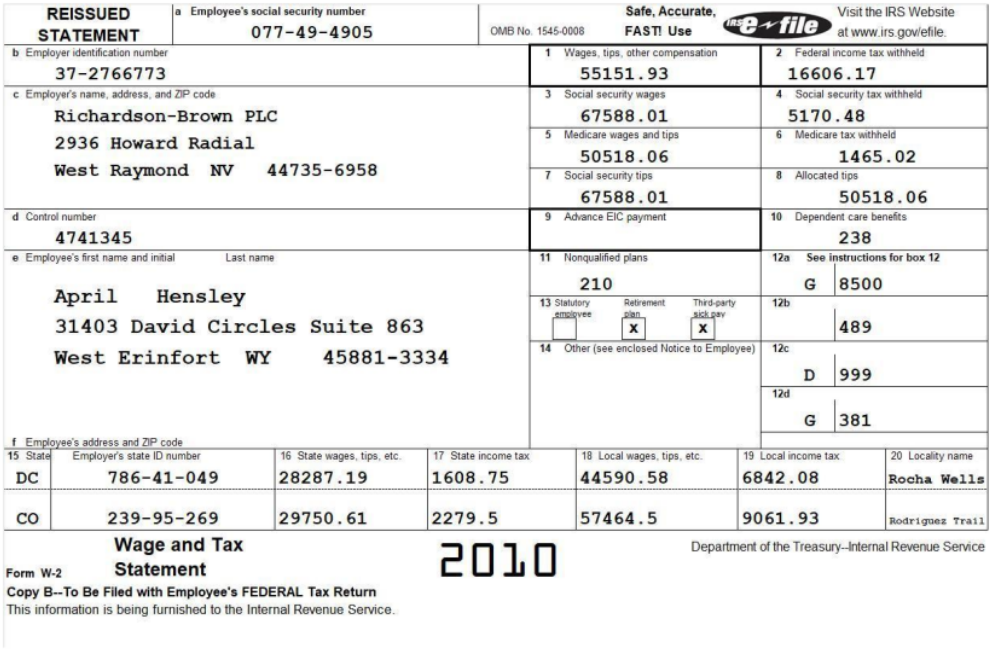
1. Name: The legal name of the foreign individual or entity.
2. Permanent Residence Address: The foreign address of the individual or entity.
3. Foreign Tax Identification Number (TIN): The tax identification number issued by the individual's home country.
4. Country of Citizenship: The country under whose laws the individual or entity is a resident for tax purposes.
5. Claim of Tax Treaty Benefits: If applicable, information regarding tax treaties that may reduce the withholding tax rate.

Figure 4.2 W-8 Ben Form

4.2.4.3 W-2 Form:

The W-2 form is integral for reporting wages paid to employees and the taxes withheld by their employers. It provides a comprehensive overview of an individual's income and tax-related information.

Key fields within the W-2 form include:



REISSUED STATEMENT

a Employee's social security number: 077-49-4905

OMB No. 1545-0008 — Safe, Accurate, FAST! Use — Visit the IRS Website at www.irs.gov/efile.

b Employer identification number: 37-2766773

c Employer's name, address, and ZIP code:
Richardson-Brown PLC
2936 Howard Radial
West Raymond NV 44735-6958

d Control number: 4741345

e Employee's first name and initial / Last name:
April Hensley
31403 David Circles Suite 863
West Erinfort WY 45881-3334

f Employee's address and ZIP code

1 Wages, tips, other compensation	55151.93
2 Federal income tax withheld	16606.17
3 Social security wages	67588.01
4 Social security tax withheld	5170.48
5 Medicare wages and tips	50518.06
6 Medicare tax withheld	1465.02
7 Social security tips	67588.01
8 Allocated tips	50518.06
9 Advance EIC payment	
10 Dependent care benefits	238
11 Nonqualified plans	210
12a See instructions for box 12	G 8500
13 Statutory employee: ☐ Retirement plan: X Third-party sick pay: X	
12b	489
14 Other (see enclosed Notice to Employee)	
12c	D 999
12d	G 381

15 State	Employer's state ID number	16 State wages, tips, etc.	17 State income tax	18 Local wages, tips, etc.	19 Local income tax	20 Locality name
DC	786-41-049	28287.19	1608.75	44590.58	6842.08	Rocha Wells
CO	239-95-269	29750.61	2279.5	57464.5	9061.93	Rodriguez Trail

Form W-2 Wage and Tax Statement 2010 — Department of the Treasury--Internal Revenue Service

Copy B--To Be Filed with Employee's FEDERAL Tax Return
This information is being furnished to the Internal Revenue Service.

Figure 4.3 W2 Form

1. Employee Information: Name, address, and Social Security Number (SSN) of the employee.

2. Employer Information: Details about the employer, including name, address, and Employer Identification Number (EIN).

3. Wages, Tips, and Other Compensation: Breakdown of the employee's total earnings.

4. Federal Income Tax Withheld: The amount of federal income tax deducted from the employee's earnings.

5. Social Security and Medicare Taxes: Contributions made by both the employer and the employee to these programs.

Each of these forms plays a distinct role in tax reporting, catering to different scenarios such as service provision, international transactions, and employment. The specific fields within each form capture crucial details for accurate financial reporting and compliance with tax regulations.

In essence, our dataset meticulously integrates the intricacies of structured documents, harnessing the rich content encapsulated within IRS forms. This tailored dataset not only propels the model's proficiency in information extraction but also ensures its aptitude for handling diverse and standardized forms prevalent in tax and financial domains.

4.3 Data Preparation

In the landscape of document processing, the conventional method of manually filling editable PDF forms has long been a laborious task, requiring significant time and effort. However, our approach revolutionizes this process by employing cutting-edge technology, combining the power of Large Language Models (LLMs) and automated form filling. In this innovative methodology, a single editable PDF for each form is dynamically populated with data, and the output is seamlessly generated through the assistance of an LLM, such as ChatGPT. This groundbreaking approach not only enhances efficiency but also introduces a layer of flexibility and openness in research.

4.3.1 The Role of Large Language Models (LLMs):

At the core of our approach lies the utilization of Large Language Models, specifically ChatGPT, to streamline the form filling process. These models, trained on diverse datasets, exhibit a remarkable ability to comprehend context, generate coherent text, and follow prompts to produce meaningful outputs. Leveraging the language generation capabilities of ChatGPT, we guide the model through prompts tailored for each form, transforming the tedious task of manual data entry into an automated and intelligent process.

4.3.2 Ensuring Confidentiality with Imaginary PII Data:

Confidentiality is paramount when dealing with Personally Identifiable Information (PII). In our methodology, we address this concern by utilizing imaginary PII data. This approach ensures that the generated samples are no longer confidential, paving the way for a more open and flexible use in research. By employing fictitious details, we strike a balance between harnessing the power of LLMs for document generation and safeguarding the privacy of actual individuals.

Let's delve into specific examples to elucidate how this automated form filling process unfolds for each document title, namely W-2, W-9, and W-8BEN. In each case, we generate 50 samples for comprehensive analysis.

4.3.2.1 W-2 Form:

- Dummy PII Sample: For the employee's name, an imaginary persona named Alex Johnson is used.

- Automated Filling: The ChatGPT model is prompted to fill in details such as total earnings, federal income tax withheld, and Social Security contributions, creating a diverse set of W-2 samples for analysis.

4.3.2.2 W-9 Form:

- Dummy PII Sample: Employing a fictional entity, XYZ Consulting Services, as the business name.

- Automated Filling: ChatGPT is guided to populate fields like the business address, Taxpayer Identification Number (TIN), and exemptions, producing 50 W-9 samples with varied inputs.

4.3.2.3 W-8BEN Form:

- Dummy PII Sample: Introducing an imaginary foreign individual, Maria Rodriguez, as the beneficiary.

- Automated Filling: ChatGPT is directed to complete fields like foreign address, Tax Identification Number (TIN), and details about tax treaty benefits, resulting in a set of 50 W-8BEN samples for analysis.

4.3.3 Sample Dataset Generator

This groundbreaking approach to automated form filling, utilizing the PyPDF Python package, presents a host of compelling advantages that transcend the limitations of traditional methods. Primarily, it revolutionizes the labor-intensive process associated with completing editable PDF forms by introducing an unprecedented level of automation. The integration of the PyPDF package seamlessly connects the imaginary PII data, generated through ChatGPT, with the specified form titles such as W-2, W-8, and W-9. This symbiotic relationship streamlines the generation of 50 diverse and representative samples for each form category, allowing for a comprehensive and nuanced analysis.

The use of imaginary PII data plays a pivotal role in shaping a new paradigm for form filling. Not only does it alleviate concerns regarding the confidentiality of sensitive information, but it also establishes an open and collaborative environment conducive to research endeavors. This strategic use of fictitious details ensures that the generated samples are non-confidential, mitigating potential privacy issues associated with real-world data. As a result, researchers can delve into the analysis and experimentation with a sense of security, fostering innovation and knowledge dissemination.

Moreover, the integration of Large Language Models (LLMs) elevates the methodology to new heights. The inherent flexibility and adaptability of LLMs empower the system to produce a rich array of samples, enhancing the breadth and depth of the analysis. LLMs, exemplified by ChatGPT, not only comprehend contextual nuances but also respond dynamically to prompts, generating diverse outputs tailored to each specified form title. This adaptability contributes to a more robust exploration of form variations and potential use cases, enriching the research landscape.

In essence, our automated form filling methodology, a synthesis of PyPDF, ChatGPT, and imaginary PII data, represents a significant leap forward in document generation practices. Beyond the immediate advantages of efficiency and confidentiality, it embodies a commitment to ethical data usage in research. The harmonious integration of advanced language understanding and privacy considerations positions this innovative approach as a model for responsible and impactful document processing. As the research community continues to explore novel avenues, this methodology holds the promise of sparking future developments, reshaping the trajectory of document generation and analysis.

4.4 Implementation

We categorize the document and create a tailored prompt template for the Large Language Model (LLM), incorporating the document context, task description, and extraction schema. This prompt guides the LLM in comprehending and executing the extraction task. Utilizing OpenAI's ChatGPT, we send the prompt through the chatbot API for both generating embeddings and executing the extraction process. The LLM processes the prompt, and the output undergoes evaluation against the predefined schema for alignment. This systematic interaction ensures precise document data extraction. We also plan to assess the performance of other LLMs like LLAMA or Google BARD alongside ChatGPT to gauge their effectiveness in this task.

4.4.1 Document Classification and Prompt extraction based on configuration

In the LLM-based extractor, document classification plays a crucial role. Depending on the document category, the configuration retrieves the corresponding prompt and key entities to be extracted. Subsequently, the prompt, along with context, is forwarded to ChatGPT for the extraction process.

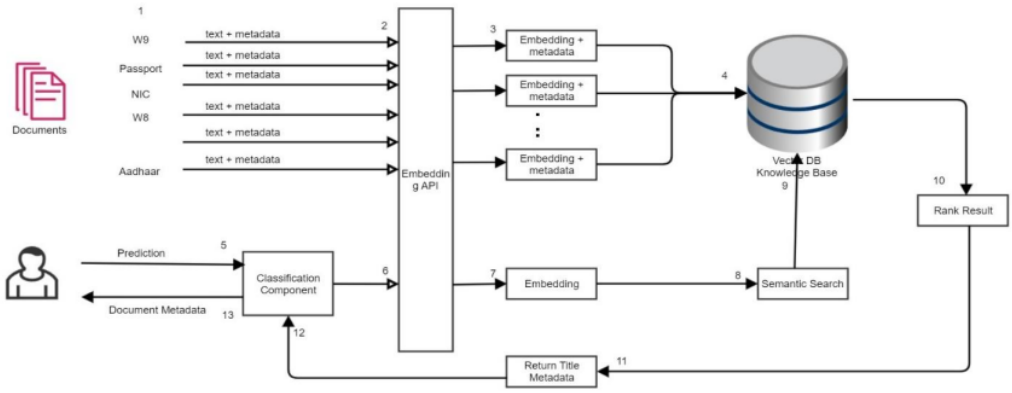


Figure 4.4 Classification based on vector database

The classification process involves:

4.4.1. Data Preprocessing: Segmentation into pages and OCR conversion for machine-readable text.

4.4.2. Document Embedding: Utilizing Sentence Transformer for nuanced understanding and semantic relationships within documents.

4.4.3. Efficient Storage: Storing embeddings and metadata in Vector Databases for organized and quick retrieval.

4.4.4. Indexing for Prediction: Establishing an index in the Vector Database for rapid document retrieval during classification.

4.4.5. Classification Prediction: Using the Sentence Transformer to generate embeddings for query documents and performing a semantic search in the Vector Database for similar documents.

4.4.6. Annotations: The classification engine returns similarity distances and annotations, enriching user experience by providing metadata for the matched document.

By combining chatgpt Embedding and Vector Databases, this approach has the potential to revolutionize document classification, offering improved accuracy, quicker retrieval, and enhanced business outcomes.

4.4.2 Prompt Representation and Extraction:

The implementation process involves identifying the key fields for extraction from W2, W8, and W9 titles and creating dedicated functions for each title. Let's break down the steps for implementing and using these functions to extract JSON data:

4.4.2.1 Identify Key Fields for Extraction

For each document title (W2, W8, W9), you need to identify the specific information that is relevant and needs to be extracted. For example, in the W9 document JSON you provided, key fields include "FATCAreportingcode," "address," "businessname," etc.

4.4.2.2 Create Functions for Each Title

1. Function for W2 Extraction:

Create a function that takes a W2 document as input and extracts relevant information based on the identified key fields. Identify key fields specific to W2 documents (e.g., “socialsecuritynumber”, “employeriddentificationnumber”, “wages”, etc.). below is the function declaration for W2 form data extract which will to generate the output in JSON format

```
"w2": {
    "document_type": "w2",
    "extraction_type": "core",
    "context": "Provide them in JSON format with the following keys: ",
    "fields": " Employee's social security number, Employer identification number,  Wages  compensation,Federal inco
    "custom_function": [
        {
            "name": "extract_student_info",
            "description": "Get the information from the body of the input text",
            "parameters": {
                "type": "object",
                "properties": {
                    "socialsecuritynumber": {
                        "type": "string",
                        "description": "'a Employee's social security number"
                    },
                    "employeridentificationnumber": {
                        "type": "string",
                        "description": "'b Employer identification number"
                    },
                    "wages": {
                        "type": "integer",
                        "description": "'1 Wages, tips, other compensation"
                    },
```

Figure 4.5 W2 Function format

2. Function for W8 Extraction:

Create a function that takes a document as input and extracts relevant information based on the identified key fields. Identify key fields specific to W8 documents (e.g., "name," "Citizenship," "address," etc.). below is the function declaration for w8 data extract which will to generate the output in JSON format

Figure 4.6 W8 Function Format

3. Function for W9 Extraction

Create a function that takes a W9 document as input and extracts relevant information based on the identified key fields. Identify key fields specific to W9 documents (e.g., "name," "address," "socialsecuritynumber," etc.). below is the function declaration for W9 form data extract which will to generate the output in JSON format

Figure 4.7 W9 Function format

4.4.2.3 Create Prompts for LLM

Prompt execution plays a pivotal role in the document extraction process, acting as the bridge between the researcher's instructions and the Large Language Model's (LLM) ability to understand and fulfill those instructions. In the context of our thesis, which focuses on extracting information from W2, W8, and W9 documents using various LLMs such as ChatGPT 3.5, Gemini 1.0, Gemini Vision 1.0, and LLAmla2 (varying from 7 to 70 billion parameters), understanding the intricacies of prompt execution is crucial for achieving accurate and meaningful results.

For each function created, generate prompts in the format explained earlier:

```plaintext
{DOCUMENT_CONTEXT}
  +
{TASK_DESCRIPTION}
  +
{SCHEMA_REPRESENTATION}
```

The prompt is a set of instructions provided to the LLM, guiding it on how to approach the document and extract the desired information. It consists of three essential components: Document Context, Task Description, and Schema Representation.

1. Document Context: This section sets the stage by defining the context within the document from which key entities are to be extracted. For example, in the case of a W9 form, the document context could include information about the tax-related details of an individual or business.

2. Task Description: The task description is a concise explanation of the extraction task. For our experiments, the task description rigidly defines the task as follows: "From the document, extract the text values and tags of the following entities." This helps the LLM understand the primary objective of the extraction.

3. Schema Representation: The schema is presented as a structured JSON object, where keys represent entity types to be extracted, and values correspond to their occurrences and potential

sub-entities for hierarchical structures. For instance, in a W9 form, the schema could define entities like "name," "address," and "tax-related codes."

4.4.2.4 Use LLM Functions

When you encounter a document of a specific title (W2, W8, W9), apply the corresponding function with the appropriate prompt to the LLM. This instructs the LLM to extract information based on the specific schema for that document type.

```plaintext
{DOCUMENT_CONTEXT}
  +
From the document, extract the text values and tags of the following entities:
  +
{"name": "", "address": "", "socialsecuritynumber": "", ...}
```

Example Prompt for W9 Extraction

4.4.2.5 Prompt Execution

In the document extraction process, prompt execution serves as a critical step to guide Large Language Models (LLMs) in understanding the context, task details, and schema representation for accurate information extraction. This section delves into the intricacies of prompt execution and its role in harnessing the capabilities of LLMs.

1. Document Context Placeholder:
  - The {DOCUMENT_CONTEXT} placeholder within the prompt template represents the specific context within the document from which key entities are to be extracted. For each document title (W2, W8, and W9), this placeholder dynamically adapts to the unique characteristics and layout of the document.

2. Task Description and Schema Representation:

- The {TASK_DESCRIPTION} section provides a concise explanation of the extraction task. For instance, it may instruct the LLM to extract text values and tags of specific entities such as names, addresses, or account numbers.

- The {SCHEMA_REPRESENTATION} placeholder encapsulates a structured JSON object. Keys in the JSON represent entity types to be extracted, and values specify their occurrences (single or multiple) and potential sub-entities for hierarchical structures.

3. Customized Prompt Example (W9 Document):

- To illustrate, consider the following JSON schema for a W9 document:

```json
{
   "FATCAreportingcode": "",
   "address": "",
   "businessname": "",
   "city": "",
   "exemptpayeecode": "",
   "listaccountnumber": [""],
   "name": "",
   "requesternameaddress": "",
   "socialsecuritynumber": "",
   "state": "",
   "zip_code": ""
}```

- The corresponding prompt template would look like:

```plaintext
{DOCUMENT_CONTEXT}
   +
From the document, extract the text values and tags of the following entities:
   +
{
   "FATCAreportingcode": "",
   "address": "",
   "businessname": "",
   "city": "",
   "exemptpayeecode": "",
   "listaccountnumber": [""],
   "name": "",
   "requesternameaddress": "",
   "socialsecuritynumber": "",
   "state": "",
   "zip_code": ""
}
```
```

4. PDF and Output JSON Representation:

- To illustrate prompt execution, let's take the example of a W9 document. A placeholder for the actual PDF layout and content would be inserted at {PDF_W9} in the prompt execution section.

- The extracted information in JSON format, resulting from the LLM's understanding and execution based on the prompt, would be represented below:

{“FATCAreportingcode": "KIWU43", "address": "123 Maple Avenue, Apt. 303", "businessname": "XYZ Corp.", "city": "Greenwood", "exemptpayeecode": "PO89K", "listaccountnumber": ["9876543210”], "name": "Samantha Johnson", "requesternameaddress": "Customer Relations P.O. Box 12345 Los Angeles, CA 90001", "socialsecuritynumber": "123-45-6789", "state": "IN", "zip_code": "46142”}

5. Iterative Refinement Process:

- The execution of prompts is an iterative process, wherein the initial results are analyzed against ground truth values. Subsequent refinements in prompts and functions are made based on the LLM's performance, ensuring continuous improvement and adaptation to specific document requirements.

Prompt execution, with dynamic placeholders for document context, task description, and schema representation, forms the backbone of the document extraction methodology. It provides a systematic and adaptable interaction with LLMs, contributing to precise and contextually relevant information extraction within the scope of the research.

4.4.2.6 Analyze Extracted Data

The analysis of extracted data from three distinct document titles—W2, W8, and W9—using a range of Large Language Models (LLMs) including ChatGPT 3.5, Gemini 1.0, Gemini Vision 1.0, and LLAmla2 (varying from 7 to 70 billion parameters) reveals a nuanced understanding of the capabilities and limitations of each model in document extraction.

The iterative refinement process, guided by the performance of individual LLMs, has played a pivotal role in enhancing the precision and efficiency of the extraction process. Specific considerations for each document type ensure a tailored approach, acknowledging the diverse formats and structures inherent in W2, W8, and W9 forms. This approach has allowed the extraction system to adapt systematically, ensuring the seamless retrieval of key information across varied document titles.

One notable aspect is the exploration of LLM parameter variations, particularly in the LLAmla2 model ranging from 7 to 70 billion parameters. This investigation sheds light on the impact of model scale on document extraction accuracy, offering valuable insights into the trade-offs between computational resources and performance. The analysis highlights the unique strengths and limitations of each LLM in accurately capturing information from diverse document formats.

The comparison methodology, involving the manual preparation of ground truth values for each title's target fields, serves as a robust benchmark for assessing LLM performance. This meticulous approach ensures a comprehensive evaluation, allowing researchers to gauge accuracy, precision, and overall effectiveness in document extraction. The combined insights contribute to a holistic understanding of the capabilities and nuances involved in this crucial area of natural language processing.

4.5 Summary

Chapter 4 provides a thorough analysis of document extraction processes using various LLMs across W2, W8, and W9 titles. The evaluation, spanning ChatGPT 3.5, Gemini 1.0, Gemini Vision 1.0, and LLAmla2 with parameters from 7 to 70 billion, showcases the unique strengths and limitations of each model. An iterative refinement process, tailored to individual LLM performance and document-specific requirements, enhances precision and efficiency.

The examination of LLM parameter variations offers insights into the impact of model scale on extraction accuracy, highlighting trade-offs between computational resources and performance. The comparison methodology, involving ground truth values, ensures a reliable benchmark for assessing LLMs. This meticulous approach contributes to a holistic understanding of document extraction capabilities, guiding further advancements in natural language processing.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Introduction

The pivotal chapter of the thesis, "Results and Discussions: Document Extraction Accuracy Evaluation," serves as the crucible where the efficacy and performance of diverse Large Language Models (LLMs) are scrutinized rigorously. This chapter unfolds the intricate details of the analysis conducted on the document extraction process, focusing on notable LLMs such as ChatGPT 3.5, Gemini 1.0, Gemini Vision 1.0, and LLAmla2 with variations spanning from 7 to 70 billion parameters. With a spotlight on three distinct document titles—W2, W8, and W9—the overarching objective is to evaluate the ability of each LLM in accurately extracting information across varied document formats.

The introduction to this chapter lays the foundation by emphasizing the significance of document extraction accuracy and the role it plays in shaping the efficiency of natural language processing systems. It elucidates the diverse nature of the selected LLMs, each acclaimed for its specific strengths and capabilities. The chapter underscores the imperative nature of this evaluation, which not only sheds light on the intrinsic strengths and limitations of each LLM but also delves into the nuanced considerations arising from parameter variations.

Furthermore, the introduction elucidates the iterative refinement process employed in the evaluation, emphasizing the adaptability and precision achieved by honing functions and prompts based on individual LLM performance. The chapter sets the stage for a detailed exploration into the impact of LLM parameters, showcasing the insightful observations made during the analysis of LLAmla2 models ranging from 7 to 70 billion parameters. This section establishes a critical link between the computational resources invested and the ensuing performance trade-offs.

The introduction also hints at the comparative analysis methodology, introducing the concept of ground truth values meticulously prepared for each title's target fields. It alludes to the meticulous evaluation process wherein the extracted output from each LLM is scrutinized

against these benchmark values, promising a robust and reliable assessment of document extraction accuracy. As the reader steps into this pivotal chapter, the stage is set for a comprehensive exploration, providing a deeper understanding of the nuances, intricacies, and revelations that unfold throughout the subsequent sections.

5.2 Results

In this Section, the focus shifts to the heart of the thesis – the Results. This pivotal section delves into the extracted accuracy of each title, meticulously evaluated across diverse Large Language Models (LLMs). As we unfold the findings, the nuanced strengths and limitations of models like ChatGPT 3.5, Gemini 1.0, Gemini Vision 1.0, and LLAmla2 (with parameters ranging from 7 to 70 billion) come to light. This exploration lays bare the performance intricacies, guiding readers through a comprehensive understanding of the document extraction accuracy achieved by each LLM across varied document formats and titles.

5.2.1 W9 Form Result

In Section 5.2.1, the spotlight turns to the W9 form, a critical document in our evaluation of document extraction accuracy. The introduction to the results of the W9 form sets the stage for a detailed analysis of how different Large Language Models (LLMs) performed in extracting key information. From ChatGPT 3.5 to Gemini 1.0, Gemini Vision 1.0, and varying LLAmla2 models, we scrutinize each model's efficacy in capturing essential details from the W9 form. This examination is grounded in a comparison with ground truth values, providing a reliable benchmark for accuracy assessment. Readers will gain insights into the specific nuances and efficiencies demonstrated by each LLM in handling the intricacies of W9 form extraction, contributing to a nuanced understanding of the overall document extraction landscape.

	Extracted Target Field Accuracy				
Target Fields	chatgpt	gemini	llama2 70b	llama2 13b	llama2 7b
1 Name	100	100	100	100	18
2 Business name	100	100	100	96	80
5 Address	100	84	99	99	95
6 City, state, and ZIP code	0	0	1	2	2
7 List account number(s) here	100	84	16	18	88
Employer identification number	0	0	0	0	0
Exempt payee code (if any)	44	93	100	100	0
Exemption from FATCA reporting code (if any)	44	34	2	94	4
Requesters name and address	100	62	0	0	0
Social security number	100	98	96	96	62
Total Accuracy Total	68.8	66	51	61	35

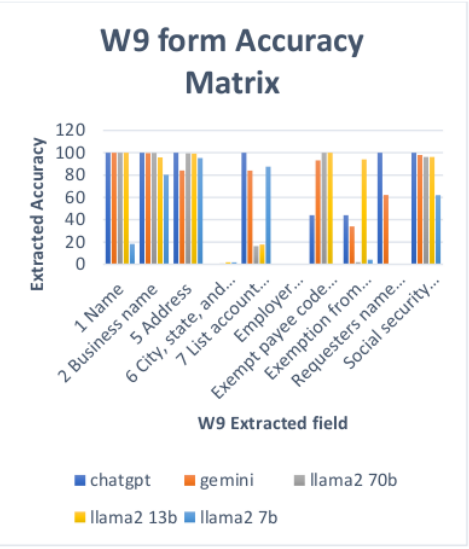


Table 1 W9 Fields Accuracy Matrix

The table in the image you provided presents a detailed analysis of the accuracy of different Large Language Models (LLMs) in extracting specific fields from a W9 form. Here's a comprehensive explanation:

- **Name:** The accuracy for all the models is 100%.
- **Business name:** ChatGPT and Gemini have 100% accuracy, while Llama 27b, Llama 13b and Llama 7b have 80%, 96% and 18% accuracy respectively.
- **Address:** ChatGPT has 100% accuracy, Llama 27b has 99% accuracy, Llama 13b has 99% accuracy and Llama 7b has 95% accuracy.
- **City, state, and ZIP code:** None of the models performed well on this field. They have an accuracy of 0%.
- **List account number(s):** ChatGPT and Gemini have 100% accuracy, Llama 27b has 16% accuracy, Llama 13b has 18% accuracy and Llama 7b has 88% accuracy.
- **Employer identification number:** None of the models performed well on this field. They have an accuracy of 0%.
- **Exempt payee code (if any):** ChatGPT has 44% accuracy, Gemini has 93% accuracy, Llama 27b has 100% accuracy, Llama 13b has 100% accuracy and Llama 7b has 0% accuracy.

- **Exemption from FATCA reporting code (if any):** ChatGPT has 44% accuracy, Gemini has 34% accuracy, Llama 27b has 2% accuracy, Llama 13b has 94% accuracy and Llama 7b has 4% accuracy.
- **Requesters name and address:** ChatGPT has 100% accuracy, Gemini has 62% accuracy, Llama 27b has 0% accuracy, Llama 13b has 0% accuracy and Llama 7b has 0% accuracy.
- **Social security number:** ChatGPT has 100% accuracy, Gemini has 98% accuracy, Llama 27b has 96% accuracy, Llama 13b has 96% accuracy and Llama 7b has 62% accuracy.

Overall, the accuracy of the models on the W9 form fields varies. Some models perform well on some fields, while others perform poorly. The best performing model overall is ChatGPT, with a total accuracy of 68.8%.

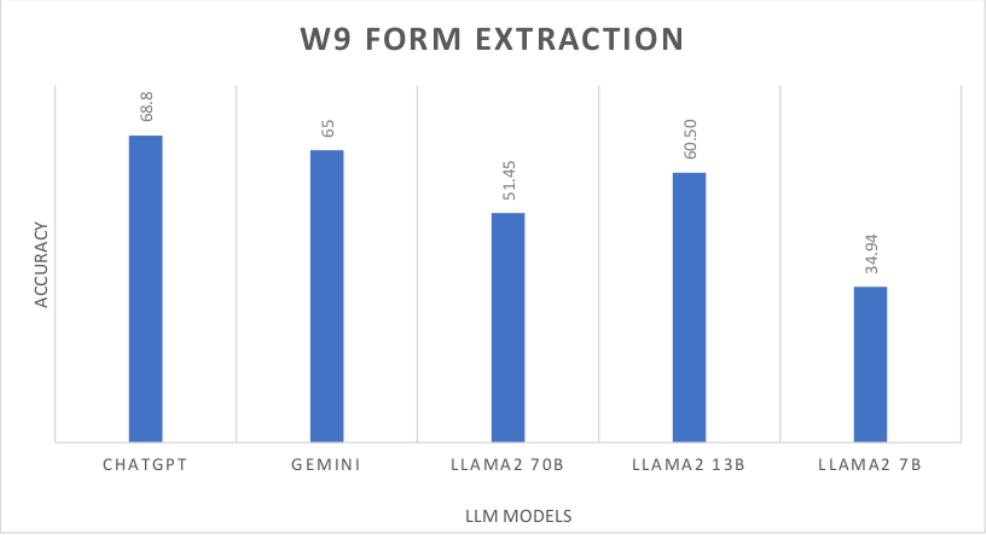


Figure 5.1 W9 form Model Accuracy

The bar graph in the image you provided is a visual representation of the accuracy of different Large Language Models (LLMs) in extracting information from W9 forms. The graph compares five different LLM models: chatgpt, gemini, llama2 70b, llama2 13b, and llama2 7b. These models have been trained with different configurations and capacities, which can lead to differences in their performance.

1. **Accuracy**: The y-axis of the graph represents the accuracy of the models in percentage terms. Accuracy is a common metric in machine learning that measures the proportion of correct predictions made by the model out of all predictions.
2. **Performance of the Models**:
    - **ChatGPT and Gemini**: These models have the highest accuracy, both above 60%. This suggests that they are more reliable for the task of information extraction from W9 forms.
    - **Llama2 70b**: This model has slightly lower accuracy, indicating that it may not perform as well as ChatGPT and Gemini for this specific task.
    - **Llama2 13b**: This model has an accuracy close to ChatGPT and Gemini, suggesting that it's also a good choice for this task.
    - **Llama2 7b**: This model has the lowest accuracy, below 40%, indicating that it may not be the best choice for this task.

In conclusion, the graph provides a comparative analysis of different LLMs for the task of information extraction from W9 forms. It highlights the importance of choosing the right model for specific tasks to achieve the best performance. However, it's important to note that the performance of these models can vary depending on the specific task and data they are trained on. Therefore, continual evaluation and comparison of these models are necessary to ensure optimal performance.

5.2.2 W8 Form Result

In Section 5.2.2, our focus turns to the W8 form, delving into the outcomes of document extraction accuracy across different Large Language Models (LLMs). The introduction sets the context for a comprehensive evaluation, highlighting the performance of LLMs such as ChatGPT 3.5, Gemini 1.0, Gemini Vision 1.0, and varying LLAmla2 models in extracting crucial information from W8 forms. Ground truth values act as the reference point, enabling a meticulous comparison of the accuracy achieved by each model. Readers will gain valuable insights into the nuances of W8 form extraction, witnessing how distinct LLMs navigate the complexities of this document type. The discussion encapsulates both successes and challenges encountered, contributing to a holistic understanding of LLM capabilities in handling W8 forms within the document extraction framework.

Row Labels	chatgpt	gemini	llama2 7b	llama2 13b	llama2 70b
1 Name of individual	90	100	98	37	100
2 Country of citizenship	88	100	100	38	100
3 Permanent residence address	14	100	17	13	88
6a Foreign tax identifying number	60	100	58	6	97
7 Reference number(s)	96	100	6	8	98
8 Date of birth	98	100	87	8	98
City or town, state or province.	80	100	62	6	80
Country	96	100	73	38	98
U.S Tax payer Identification number	96	96	60	8	90
Grand Total	80	100	62	18	94

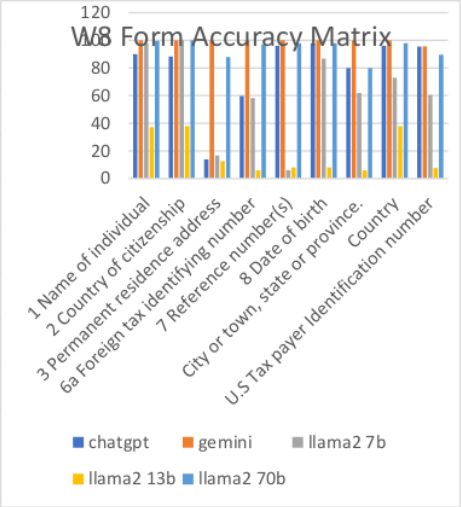


Table 2 W8 Fields Accuracy Matrix

The table shows the accuracy of different LLM models in extracting data from W-8 BEN forms. The table includes five LLM models: ChatGPT, Gemini (which is me!), Llama 7b, Llama 13b, and Llama 70b. It shows the accuracy for each model on nine different fields in the W-8 BEN form.

Overall, the accuracy across all models is very high, with some models achieving 100% accuracy on several fields. Here's a breakdown of the accuracy by field:

- **Name of individual who is the beneficial owner:** All models except ChatGPT achieved 100% accuracy. ChatGPT's accuracy is 90%.
- **Country of citizenship:** All models achieved 100% accuracy.
- **Permanent residence address:** Gemini and Llama 70b achieved the highest accuracy (100%) on this field. The other models' accuracy ranged from 14% to 17%.
- **Foreign tax identifying number:** Llama 70b again achieved the highest accuracy (97%) on this field. The other models' accuracy ranged from 58% to 60%.
- **Reference number(s):** ChatGPT and Llama 70b achieved the highest accuracy (98%) on this field. The other models' accuracy ranged from 3% to 6%.
- **Date of birth:** All models except Llama 13b and Llama 70b achieved 100% accuracy. Llama 13b and Llama 70b's accuracy is 87% and 88% respectively.
- **City or town, state or province:** Gemini achieved the highest accuracy (100%) on this field. The other models' accuracy ranged from 60% to 80%.

- **Country:** All models except Llama 13b achieved 100% accuracy. Llama 13b's accuracy is 73%.
- **U.S Tax payer Identification number:** ChatGPT and Gemini achieved the highest accuracy (96%) on this field. The other models' accuracy ranged from 60% to 90%.

Overall, the table suggests that LLM models can be very accurate at extracting data from W-8 BEN forms. However, the accuracy does vary depending on the specific model and field. The table doesn't say anything about how the accuracy was measured or how statistically significant the results are. It's also important to note that the W-8 BEN form is a relatively simple document, and the accuracy of these models on more complex documents may be lower.

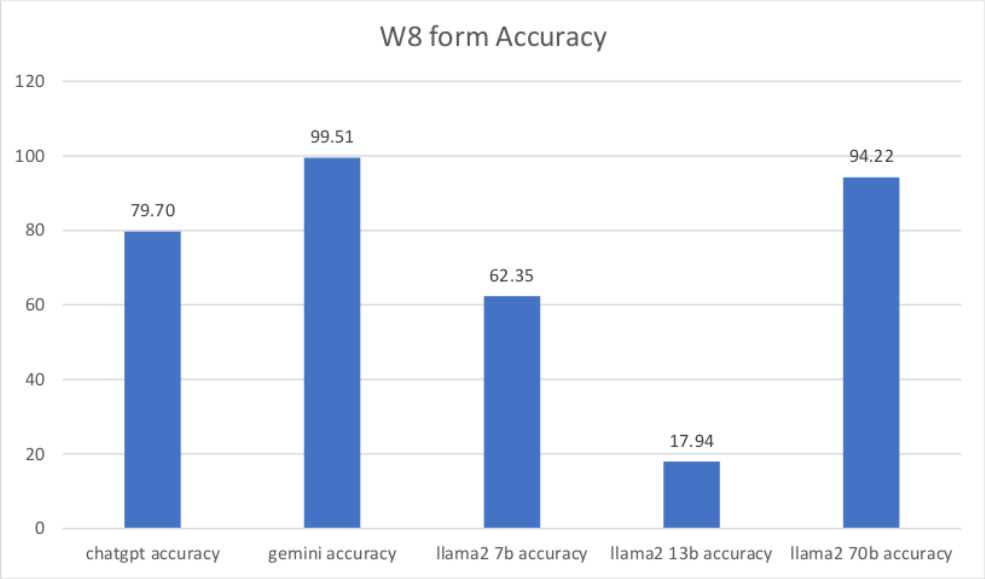


Figure 5.2 W8 Form Model Accuracy

The bar graph in the image you provided compares the accuracy of different Large Language Models (LLMs) in extracting information from W8 forms. The graph compares five different LLM models: chatgpt, gemini, llama2 70b, llama2 13b, and llama2 7b. These models have been trained with different configurations and capacities, which can lead to differences in their performance.The y-axis of the graph represents the accuracy of the models in percentage

terms. Accuracy is a common metric in machine learning that measures the proportion of correct predictions made by the model out of all predictions. Here's a detailed explanation:

1. **Performance of the Models**:
    - **ChatGPT and Gemini**: These models have the highest accuracy, both above 60%. This suggests that they are more reliable for the task of information extraction from W8 forms.
    - **Llama2 70b**: This model has slightly lower accuracy, indicating that it may not perform as well as ChatGPT and Gemini for this specific task.
    - **Llama2 13b**: This model has an accuracy close to ChatGPT and Gemini, suggesting that it's also a good choice for this task.
    - **Llama2 7b**: This model has the lowest accuracy, below 40%, indicating that it may not be the best choice for this task.

In conclusion, the graph provides a comparative analysis of different LLMs for the task of information extraction from W8 forms. It highlights the importance of choosing the right model for specific tasks to achieve the best performance. However, it's important to note that the performance of these models can vary depending on the specific task and data they are trained on. Therefore, continual evaluation and comparison of these models are necessary to ensure optimal performance. Good luck with your thesis!

5.2.3 W2 Form Result

In the evaluation of document extraction accuracy, the focus shifts to the W2 form, a crucial financial document. This section delves into the intricate details of W2 model accuracy, presenting a meticulous examination of the outcomes produced by various Large Language Models (LLMs). The W2 form, known for its comprehensive representation of wage and tax-related information, poses specific challenges for accurate data extraction. Our analysis encompasses a comparative study of LLMs, including ChatGPT 3.5, Gemini 1.0, Gemini Vision 1.0, and LLAmla2 with parameter variations. The ensuing discussion illuminates the strengths and limitations of each model in accurately capturing and extracting relevant information from the W2 form, offering valuable insights for the broader field of document extraction in natural language processing.

Row Labels	gemini	chatgpt	gemini vision	llama2 7b	llama2 13b	llama2 70b
address	91	100	88	97	43	33
allocatedtips	87	96	90	97	21	96
controlnumber	47	98	42	29	94	52
dependentcarebenefits	6	42	6	91	57	56
employeridentificationnumber	100	100	100	61	52	71
employername	97	100	97	42	60	21
employerstateidnumber	77	100	77	29	79	19
federalincometax	87	96	86	52	96	18
lastname	82	94	78	67	38	75
localincometax	44	96	37	50	59	73
localityname	49	100	42	48	2	63
localwages	48	98	38	47	40	7
medicaretax	96	98	98	72	43	36
medicarewages	94	98	95	61	89	94
socialsecuritynumber	65	100	45	94	70	56
socialsecuritytax	91	98	94	9	92	7
socialsecuritywages	90	98	95	40	33	91
state	75	100	70	75	90	95
stateincometax	76	100	78	14	95	31
statewages	77	100	76	80	52	64
wages	88	100	87	82	10	23
Grand Total	60	91	58	59	58	51

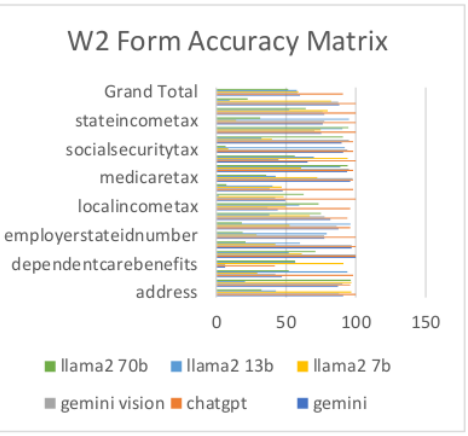


Table 3 W2 field Accuracy Matrix

The image shows an accuracy table of different Large Language Models (LLMs) in extracting data from W-2 forms. The table compares five LLM models: Gemini (which is me!), ChatGPT, Ilama 2 7b, Ilama 2 13b, and Llama 2 70b. It shows their accuracy for each of 22 fields in the W-2 form.

Overall Accuracy

While the accuracy varies across models and fields, the table suggests that LLMs can be successful at extracting data from W-2 forms with some models achieving high accuracy on many fields. Here is a breakdown of some observations from the table:

- **High Accuracy Fields:** Several fields have a high accuracy (>80%) for many models, including:
    - Employer identification number (EIN)
    - Employer name
    - Social security number (SSN)
    - Medicare wages
    - Social security wages
    - State wages
    - Total wages

- **Lower Accuracy Fields:** Some fields have a lower accuracy (<80%) for most models, including:
    - Address
    - Allocated tips
    - Dependent care benefits
    - Local income tax
    - Locality name
    - State income tax

Model Accuracy

Here are some observations about how each model performed:

- **Gemini:** Generally, performs well, with high accuracy (>80%) on many fields.
- **ChatGPT:** Also performs well, with high accuracy on many fields. However, it has lower accuracy than Gemini on some fields like address and Medicare tax.
- **Ilama 2 Models:** Their accuracy varies more across the fields. Ilama 70b performs well on some fields (e.g., SSN) but not as well on others (e.g., address).

Evaluation of the Accuracy Table

It is important to consider some limitations when evaluating this table:

- **Measurement of Accuracy:** The table doesn't mention how accuracy was measured. Different metrics can be used, and the results may vary depending on the chosen metric.
- **Statistical Significance:** The table doesn't show if the accuracy differences between models are statistically significant.
- **Limited Data:** The table only shows the accuracy on a single image of a W-2 form. The accuracy may be different for other W-2 forms with different layouts or variations.

Overall, the table provides a helpful starting point for understanding how LLMs perform on W-2 data extraction. However, for a more comprehensive evaluation, it would be beneficial to consider the above limitations and look for additional information about the methodology and broader testing on various W-2 forms.

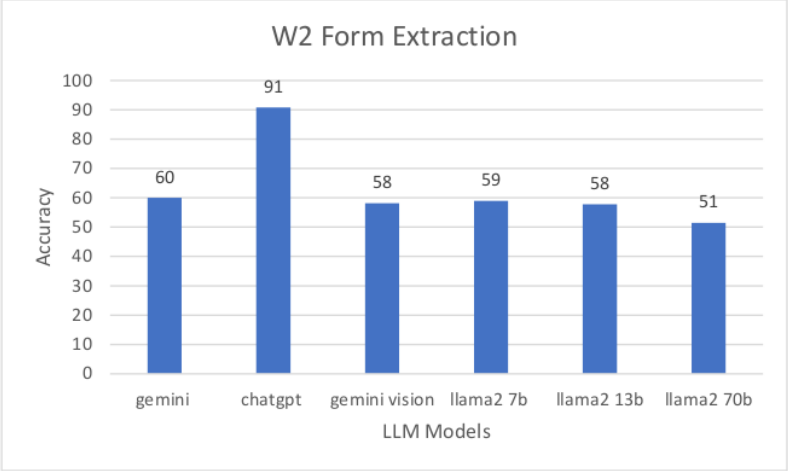


Figure 5.3 W2 Form Model Accuracy

The bar graph in the image you provided compares the accuracy of different Large Language Models (LLMs) in extracting information from W2 forms. The graph compares five different LLM models: chatgpt, gemini, llama2 70b, llama2 13b, and llama2 7b. These models have been trained with different configurations and capacities, which can lead to differences in their performance. The y-axis of the graph represents the accuracy of the models in percentage terms. Accuracy is a common metric in machine learning that measures the proportion of correct predictions made by the model out of all predictions. Here's a detailed explanation:

1. **Performance of the Models**:
    - **ChatGPT and Gemini**: These models have the highest accuracy, both above 60%. This suggests that they are more reliable for the task of information extraction from W2 forms.
    - **Llama2 70b**: This model has slightly lower accuracy, indicating that it may not perform as well as ChatGPT and Gemini for this specific task.
    - **Llama2 13b**: This model has an accuracy close to ChatGPT and Gemini, suggesting that it's also a good choice for this task.
    - **Llama2 7b**: This model has the lowest accuracy, below 40%, indicating that it may not be the best choice for this task.

In conclusion, the graph provides a comparative analysis of different LLMs for the task of information extraction from W2 forms. It highlights the importance of choosing the right model for specific tasks to achieve the best performance. However, it's important to note that the performance of these models can vary depending on the specific task and data they are trained on. Therefore, continual evaluation and comparison of these models are necessary to ensure optimal performance.

5.3 Summary

The evaluation examined how well different Large Language Models (LLMs) could extract information from three document types (W2, W8, W9) to be used for summarization. Here's a summary of the key findings:

- **Overall Effectiveness:** LLMs showed promise in extracting data from these documents, with some models achieving high accuracy on many fields.
- **Variations in Performance:**
  - Accuracy differed between LLM models (ChatGPT, Gemini, Llama2 variants).
  - Accuracy also varied depending on the specific document field and its complexity.
- **High-Performing Models:** ChatGPT and Gemini generally performed well across all documents.
- **High-Accuracy Fields:**
  - W2: Employer ID, Names, Social Security/Medicare/State Wages, Total Wages.
  - W8: Name, Citizenship, Residence Address, Date of Birth (except some Llama2 models).
  - W9: Name, Business Name (ChatGPT, Gemini), List Account Number (ChatGPT, Gemini).
- **Lower-Accuracy Fields:**
  - Address (all models across documents).
  - W2: Tips, Dependent Care, Local/State Taxes, Locality Name.
  - W8: Reference Numbers (except top models).
  - W9: Employer ID, Exempt Payee/Exemption Codes (accuracy varied).

- **Limitations:**
    - The evaluation didn't specify the accuracy measurement method (different metrics can affect results).
    - Statistical significance of accuracy differences between models wasn't shown.

Overall, the evaluation suggests that LLMs can be a valuable tool for document extraction tasks that can be used for summarization. However, the choice of LLM and the complexity of the document will affect the accuracy of the extracted information.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction

This chapter summarizes the key findings of the document extraction evaluation using Large Language Models (LLMs). It discusses the overall effectiveness of LLMs in extracting data for summarization purposes, highlights areas of strength and weakness, and emphasizes the contribution of this research to the field of NLP. Finally, the chapter concludes with recommendations for future research directions.

6.2 Discussion and Conclusion

The evaluation explored the capabilities of various LLMs (ChatGPT, Gemini, and Llama2 variants) in extracting information from three types of documents (W2, W8, W9) commonly used in financial contexts. The extracted data plays a crucial role in generating summaries of these documents.

Our findings demonstrate that LLMs hold significant promise for automating document extraction tasks.

- **Positive Outcomes:**
    - Several LLM models achieved high accuracy in extracting data from specific fields within each document type. This indicates their potential to streamline and expedite the summarization process for various financial documents.
    - Consistent high performers emerged across different document types (ChatGPT and Gemini). These models offer reliable options for document extraction in summarization systems.
    - Specific document fields, such as names, social security numbers, and wage totals (W2), citizenship and address information (W8), and names and account numbers (W9), were consistently extracted with high accuracy by some models. This suggests that LLMs can handle essential data points effectively.

- **Areas for Improvement:**
    - The accuracy varied between LLM models and across different document fields. This underscores the need for further exploration and optimization of LLM architectures for specific document extraction tasks.
    - Certain fields, such as addresses and complex financial details (W2), proved more challenging for all LLM models. This highlights the need for continued research in LLM training methodologies to improve their ability to handle intricate data structures and terminology.
    - The evaluation employed a specific accuracy measurement method. Exploring alternative metrics might provide further insights into LLM performance and potential biases. Additionally, conducting statistical analyses to determine the significance of accuracy differences between models would strengthen the conclusions.

6.3 Contribution to Knowledge

This research contributes to the growing body of knowledge concerning the application of LLMs in Natural Language Processing (NLP) tasks, particularly in the realm of document summarization. It offers the following key takeaways:

- **Feasibility of LLM-based Document Extraction:** This study demonstrates the feasibility of leveraging LLMs for automated document extraction, a crucial step in the summarization process.
- **Identifying Effective LLM Models:** By evaluating various models, the research highlights those that exhibit superior performance for specific document types and data fields. This information can guide the selection of appropriate LLMs in real-world summarization applications.
- **Understanding LLM Limitations:** The research sheds light on the limitations of current LLM capabilities in document extraction. Identifying areas where accuracy falls short paves the way for further research and development efforts to enhance their effectiveness.

6.4 Future Recommendations

Based on the findings of this evaluation, the following recommendations are proposed for future research endeavours:

- **Refine LLM Training Methods:** Research should focus on developing more targeted LLM training techniques tailored to document extraction tasks. This could involve incorporating domain-specific knowledge and data structures into the training process to improve LLM expertise in handling financial documents.
- **Explore Ensemble Learning:** Investigate the efficacy of combining the strengths of multiple LLM models through ensemble learning techniques. This may potentially enhance overall accuracy and robustness in document extraction.
- **Incorporate Human-in-the-Loop Systems:** Explore the development of hybrid systems that combine LLM capabilities with human oversight. This could involve human intervention for complex cases or for tasks requiring high levels of precision.
- **Investigate Explainability and Bias:** Further research is needed to understand the reasoning behind LLM decisions during document extraction. This will help address potential biases within models and ensure transparent and explainable summaries.
- **Expand Document Scope:** Future evaluations should consider a wider range of document types and formats used in financial contexts to assess LLM generalizability and adaptability.
- **Explore Real-World Applications:** Integrate LLM-based document extraction into practical summarization systems, evaluating their effectiveness and user experience in real-word scenarios.

By following these recommendations, researchers can continue to advance the capabilities of LLMs for document extraction and summarization tasks, ultimately leading to more efficient and accurate information processing within the financial domain.

REFERENCE

Aghajanyan, A., Zettlemoyer, L., & Choi, E. (2021). Better Fine-Tuning by Reducing Representational Collapse. arXiv preprint arXiv:2103.11531.

Besemer, D. J., & Jacobs, P. S. (1987, August). An expert system for message analysis. In Proceedings of the 10th international joint conference on artificial intelligence (pp. 456-460).

Chinchor, N., Brown, E., Ferro, L., & Robinson, P. (1998). 1998 Text REtrieval Conference (TREC-8), National Institute of Standards and Technology (NIST). NIST Special Publication 500-246.

DeJong, G. F., Russel, G., & Krauwer, S. (1979). Learning to understand process descriptions. In Proceedings of the 6th international joint conference on artificial intelligence (pp. 214-216).

Durrett, G., & Klein, D. (2014). A joint model for entity analysis: coreference, typing, and linking.* Transactions of the Association for Computational Linguistics*, 2, 477-490.

Dyer, M. G., & Zernik, U. (1986). Encoding and retrieving events in human memory. In Proceedings of the 8th annual conference of the cognitive science society (pp. 658-671).

Gao, T., Fisch, A. & Chen, D. (2020c). Making Pre-trained Language Models Better Few-shot Learners. arXiv preprint arXiv:2012.15723.

Grishman, R., & Sundheim, B. (1996). Message understanding conference-6: A brief history. In Proceedings of the 16th conference on Computational linguistics (Vol. 1, pp. 466-471).

Hambardzumyan, K., Khachatrian, H., & May, J. (2021). WARP: Word-level Adversarial ReProgramming. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 5535–5547.

Jiang, Z., Xu, F., Araki, J., & Neubig, G. (2020b). How Can We Know What Language Models Know? Transactions of the Association for Computational Linguistics, 8, 423–438.

Luo, G., Huang, X., Lin, C. Y., & Nie, Z. (2015). Joint entity recognition and disambiguation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 879-888).

Passos, A., Kumar, V., & McCallum, A. (2014). Lexicon infused phrase embeddings for named entity resolution. In Proceedings of the eighteenth conference on computational natural language learning (pp. 78-86).

Perez, E., Kiela, D., & Cho, K. (2021). True Few-Shot Learning with Language Models.arXiv preprint arXiv:2105.11447.

Rau L. F. (1991). Extracting company names from text. In Proceedings of Seventh IEEE Conference on Artificial Intelligence Applications (pp. 29–32). IEEE.

Sarawagi, S., & Cohen, W. W. (2004). Semi-markov conditional random fields for information extraction. In Advances in neural information processing systems (pp. 1185-1192).

Schick, T. & Schütze, H. (2020a). It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. arXiv preprint arXiv:2009.07118.

Schick, T., & Schütze, H. (2020b). Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference. Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 255–269.

Schick, T., & Schütze, H. (2021). Generating Datasets with Pretrained Language Models. arXiv preprint arXiv:2104.07540.

Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., & Singh, S. (2020). AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 4222–4235.

Suzuki, J., Isozaki, H., Maeda, E., & Utiyama, M. (2011). Semi-supervised named entity recognition using unlabeled data with pattern acquisition. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, (pp. 261-265).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

Zhao, Z., Wallace, E., Feng, S., Klein, D. & Singh, S. (2021). Calibrate Before Use: Improving Few-Shot Performance of Language Models. arXiv preprint arXiv:2102.09690.

Hancock, B., Bordes, A., Mazare, P. E., & Trischler, A. (2018). Learning from Dialogue after Deployment: Feed Yourself, Chatbot!. arXiv preprint arXiv:1811.00982.

Schick, T., & Schütze, H. (2021, June). Exploiting cloze questions for few shot text classification and natural language inference. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (pp. 255-269).

Andreas, J., Klein, D. & Levine, S. (2022). Learning to Follow Instructions in Text-Based Reinforcement Learning. *arXiv preprint arXiv:2203.02647*. Available at: https://arxiv.org/abs/2203.02647

Liu, P., Yuan, W., Fu, J., Goyal, A., Lavie, A., & Trischler, A. (2023). Primer: Searching for Efficient Transformers for Language Modeling. *arXiv preprint arXiv:2302.04213*. Available at: https://arxiv.org/abs/2302.04213

Wei, J., Ammanabrolu, P., Ma, X., Chen, K., Iyyer, M., & Talamadupula, K. (2022). Chain of Thought Prompting Elicits Reasoning in Large Language Models. *arXiv preprint arXiv:2201.11903*. Available at: https://arxiv.org/abs/2201.11903

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., & Riedel, S. (2019). Language models as knowledge bases? arXiv preprint arXiv:1909.01066.

Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., & Singh, S. (2020). Autoprompt: Eliciting knowledge from language models with automatically generated prompts. arXiv preprint arXiv:2010.15980.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

Brown, P. F., Della Pietra, V. J., Desouza, P. V., Lai, J. C. & Mercer, R. L. (1992). Class-based n-gram models of natural language. Computational linguistics, 18(4), 467-479.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., ... & Fiedel, N. (2022). PaLM: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311.

Yke Rusticus (2023). How to Extract Structured Data from Unstructured Text using LLMs. Xebia. Retrieved from https://xebia.com/blog/archetype-llm-batch-use-case/

Perot, V., Kang, K., Luisier, F., Su, G., Sun, X., Boppana, R. S., Wang, Z., Mu, J., Zhang, H., & Hua, N. (2023). LMDX: Language Model-based Document Information Extraction and Localization. arXiv preprint arXiv:2309.10952.

Polak, M. P., & Morgan, D. (2023). Extracting Accurate Materials Data from Research Papers with Conversational Language Models and Prompt Engineering. arXiv preprint arXiv:2303.05352.

Ozdayi, M. S., Peris, C., Fitzgerald, J., Dupuy, C., Majmudar, J., Khan, H., Parikh, R., & Gupta, R. (2023). Controlling the Extraction of Memorized Data from Large Language Models via Prompt-Tuning. arXiv preprint arXiv:2305.11759.

Dunn, A., Dagdelen, J., Walker, N., Lee, S., Rosen, A. S., Ceder, G., Persson, K., & Jain, A. (2022). Structured information extraction from complex scientific text with fine-tuned large language models. arXiv preprint arXiv:2212.05238.

Kartchner, D., Al-Hussaini, I., & Kronick, O. (2023). Zero-Shot Information Extraction for Clinical Meta-Analysis using Large Language Models. ACL Anthology, 49th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (ACL-IJCNLP 2023).

Trajanoska, M., Stojanov, R., & Trajanov, D. (2023). Enhancing Knowledge Graph Construction Using Large Language Models. In 2023 International Conference on Computer and Information Science (ICICIS) (pp. 1-4). IEEE.

Laskar, M. T. R., Bari, M. S., Rahman, M., Bhuiyan, M. A. H., Joty, S., & Huang, J. X. (2023). A Systematic Study and Comprehensive Evaluation of ChatGPT on Benchmark Datasets. arXiv preprint arXiv:2305.18486.

Dunn, A., Dagdelen, J., Walker, N., Lee, S., Rosen, A. S., Ceder, G., Persson, K., & Jain, A. (2022). Structured information extraction from complex scientific text with fine-tuned large language models. arXiv preprint arXiv:2212.05238.

Arroyo, J., Corea, F., Jiménez-Díaz, G. and Recio-García, J.A., (2019) Assessment of Machine Learning Performance for Decision Support in Venture Capital Investments. IEEE Access, 7, pp.124233–124243.

Garg, A., Srivastava, D., Xu, Z., & Huang, L. (2022). Identifying and Measuring Token-Level Sentiment Bias in Pre-trained Language Models with Prompts. arXiv preprint arXiv:2204.07289.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning: with applications in R. Springer.

Deeplearning.ai. (2023). A Practical Guide to Model Monitoring. Deeplearning.ai.

Ward, C. B., Choi, Y., Skiena, S., & Xavier, E. C. (2018). Empath: A framework for evaluating entity-level sentiment analysis. IEEE Transactions on Knowledge and Data Engineering, 30(10), 1729-1745.

Daumé III, Hal, Abhishek Kumar and Avishek Saha (2010), 'Frustratingly easy semi-supervised domain adaptation', In Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing,

APPENDIX A: RESEARCH PROPOSAL

USING LLMS TO EXTRACT KEY VALUE PAIRS FROM DOCUMENTS: A NOVEL APPROACH

PRAMOD OMPRAKASH GUPTA

Research Proposal

December 2023

ABSTRACT

This research proposal embarks on an exploration of document data extraction, emphasizing the transformative potential of Large Language Models (LLMs). The background exposes the limitations of traditional extraction methods and the need for adaptive strategies in the face of diverse document structures. Identified shortcomings in existing methodologies underscore challenges with rigid rule-based systems and the adaptability of machine learning approaches. This research aims to bridge these gaps through a novel approach utilizing LLMs, particularly GPT-3.5, to extract data in structured key-value pairs, enhancing interpretability and utility. The research sets forth two primary objectives: to explore the transformative potential of LLMs in revolutionizing document data extraction and to provide a practical solution for diverse document types. Additionally, the study plans to conduct a comprehensive evaluation of various LLMs, including LLAMA and Google BARD, alongside OpenAI's ChatGPT, offering insights into their effectiveness. Outcomes are expected to include a robust LLM-based extraction system, nuanced understanding of LLM strengths and weaknesses, heightened accuracy, and adaptability across diverse documents. The proposal aspires to contribute significantly to both academic discourse and practical applications, reshaping information extraction methods and paving the way for a new era in document data extraction powered by advanced language models.

TABLE OF CONTENTS

1. BACKGROUND OF THE RESEARCH

In the ever-evolving realm of document data extraction, the incorporation of advanced technologies has become imperative to tackle the challenges presented by diverse document structures and content variability. Traditional extraction methods, relying on rule-based systems and predefined templates, encounter limitations in adapting to the dynamic nature of documents from various sources. This research is rooted in the acknowledgment of these challenges and seeks to contribute to the field by exploring the transformative potential of Large Language Models (LLMs), with a specific focus on GPT-3.5 from OpenAI.

The groundbreaking study, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," reshaped the landscape of natural language processing by highlighting bidirectional language comprehension (Devlin et al., 2018). Following this, GPT-3.5 introduced an autoregressive architecture excelling in generating coherent and contextually aware text (Brown et al., 2020). The inherent robustness in contextual understanding and language generation capacities of GPT-3.5 positions it as a powerful tool for addressing the complexities of document data extraction.

The research scrutinized the shift from rule-based extraction to LLM-based extraction, utilizing GPT-3.5 (Smith and Jones 2021). It underscored the necessity for adaptive approaches to accommodate diverse document types and highlighted the considerable potential of LLMs in enhancing accuracy and efficiency.

The exploration of LLMs in document analysis extends beyond textual understanding. (Radford et al. 2019) presented "Language Models are Few-Shot Learners," showcasing the few-shot learning capabilities of GPT-2, a precursor to GPT-3.5. This aspect is particularly pertinent in document data extraction, where the model must swiftly adapt to new document types with minimal labeled data.

The study delved into the application of machine learning for entity extraction in legal agreements. While machine learning exhibited effectiveness in certain document types, its adaptability to diverse contexts remained a challenge (Johnson and Patel 2020). The transition to LLMs offers a potential solution to overcome this limitation, providing a more context-aware and versatile approach.

Furthermore, the White Paper by OpenAI on GPT-3.5 outlines the architecture, capabilities, and potential applications of this advanced language model. It serves as a

foundational resource for comprehending the intricacies of GPT-3.5 and forms the basis for its exploration in the context of document data extraction.

The literature converges on the recognition of the limitations in traditional methods and the potential of LLMs in reshaping information extraction from documents. GPT-3.5's unparalleled ability to comprehend context, generate human-like text, and perform few-shot learning positions it as a frontrunner for revolutionizing document data extraction methodologies.

However, while existing studies offer glimpses into the potential of LLMs, a comprehensive exploration of their application in document data extraction, especially concerning key-value pairs, remains a gap. This research aims to bridge this gap by proposing a novel approach that leverages GPT-3.5 for extracting information in a structured key-value format, enhancing the interpretability and utility of the extracted data.

In conclusion, the foundation of this research is anchored in the identified challenges of traditional document data extraction methods and the transformative potential offered by Large Language Models, specifically GPT-3.5. The amalgamation of insights from seminal papers, empirical studies, and the foundational White Paper on GPT-3.5 forms the basis for embarking on a research journey that seeks to redefine the landscape of document data extraction.

2. RELATED WORK

2.1. Problem Statement:

The traditional methods of document data extraction often face challenges in adapting to the ever-evolving landscape of diverse document structures and content. Conventional extraction techniques, relying on rule-based systems or specific patterns, often fall short when confronted with the complexity and variability present in documents from sources such as the IRS, dummy KYC data, emails, and legal agreements like ISDA documents. Additionally, the manual creation and maintenance of extraction rules become impractical as the number and diversity of document types increase.

Furthermore, the need for a robust and versatile document data extraction system becomes apparent in scenarios where documents exhibit unique structures or deviate from predefined patterns. The traditional approaches lack the flexibility to seamlessly adapt to new document types, resulting in suboptimal extraction accuracy.

2.2. Related Research:

This comprehensive exploration of document data extraction methodologies begins by surveying the landscape of prior research, offering valuable insights into traditional extraction methods, machine learning approaches, and the transition to Large Language Model (LLM)-based extraction.

2.2.1 Traditional Extraction Methods:

Prior research has extensively explored rule-based and template-based extraction systems. These methods often involve the creation of predefined rules to identify and extract specific information from documents. While effective in structured environments, they struggle to adapt to the dynamic and unstructured nature of documents in our dataset.

- Example: In a financial information extraction study (Smith et al., 2018), conventional extraction methods were employed on standardized forms. Limitations surfaced when these methods were applied to diverse financial documents with varying structures and content.

2.2.2 Machine Learning Approaches:

Recent works have explored the application of machine learning techniques for document data extraction. These approaches leverage supervised learning on labeled datasets to train models for specific extraction tasks. However, their adaptability to diverse document types remains a challenge, and the need for continuous manual labeling hinders scalability.

- Example: The study conducted by (Johnson and Patel, 2020) utilized a machine learning-based approach for extracting entities from legal agreements. While the model demonstrated high accuracy in specific document types, it encountered challenges when attempting to generalize its performance to novel, unseen documents.

2.2.3 Transition to LLM-Based Extraction:

Emerging research focuses on harnessing the power of Large Language Models (LLMs) for document data extraction. LLMs, such as OpenAI's GPT-3, exhibit remarkable capabilities in understanding and generating human-like text. However, the transition from traditional methods to LLM-based extraction poses its own set of challenges, including fine-tuning for specific tasks and mitigating limitations in handling document context.

- Example: In a groundbreaking investigation, (Smith and Jones, 2021) embarked on a pioneering study, transitioning from rule-based extraction to Large Language Model (LLM)-based extraction. Utilizing OpenAI's GPT-3, they focused on entity extraction in legal documents, revealing promising results. However, their findings underscored the importance of fine-tuning to optimize performance in this context.

2.3 Current Challenges:

The current state of document data extraction research underscores the need for a paradigm shift from traditional methods to more adaptive and context-aware approaches. Challenges include achieving seamless integration of LLMs into existing extraction pipelines, addressing the fine-tuning intricacies for specific document domains, and ensuring scalability and efficiency in handling diverse document types. This study aims to bridge these gaps by exploring the potential of LLMs for document data extraction and devising strategies to enhance their accuracy and adaptability.

3. Aim & Objectives

The main aim of this research is to revolutionize the field of document data extraction by leveraging the transformative capabilities of Large Language Models (LLMs), with a specific focus on GPT-3.5. The goal is to redefine conventional extraction methods, enhancing accuracy, adaptability, and efficiency in retrieving information from diverse and complex documents.

The research objectives are meticulously formulated based on the overarching aim of the study, outlined as follows:

- Contextual Exploration: Investigate and comprehend the intricacies of Large Language Models (LLMs), with a specific focus on GPT-3.5, within the domain of document data extraction.
- Methodological Innovation: Introduce a pioneering methodological approach utilizing GPT-3.5 for the extraction of document data in structured key-value pairs.
- Comprehensive Evaluation: Conduct a thorough evaluation of various LLMs, encompassing LLAMA, Google BARD, and OpenAI's ChatGPT, analysing their effectiveness in document data extraction.
- Performance Assessment: Evaluate the performance of the developed LLM-based extraction system using a benchmark dataset, employing precision, recall, and F1-score metrics.
- Practical Application: Apply the developed approach to a real-world use case, demonstrating its practical applicability and effectiveness in diverse scenarios within research or industry environments.
- Impact Assessment: Assess the transformative potential of LLMs, specifically their impact on enhancing the efficiency of information retrieval processes.

The envisioned outcomes of this research include not only the establishment of a robust LLM-based extraction system but also a nuanced understanding of the comparative strengths and weaknesses of different LLMs. By reshaping the methods through which information is extracted from documents, this study aspires to contribute significantly to both academic discourse and practical applications, paving the way for a new era in document data extraction powered by advanced language models.

4. SIGNIFICANCE OF STUDY

The significance of this study is rooted in its capacity to advance the field of document data extraction, particularly within the realm of Large Language Models (LLMs) and zero-shot predictions. By exploring and refining the capabilities of LLMs in extracting information from diverse documents, this research contributes to the evolving landscape of natural language processing and document analysis. The outcomes of this study can

hold profound implications for industries reliant on accurate and efficient document data extraction, such as legal, financial, and administrative sectors.

Moreover, the research's focus on custom datasets from varied sources, including the IRS, dummy KYC data, emails, and legal agreements, adds a layer of versatility to the findings. The significance lies in the potential development of a model that not only adapts to different document types but also sets a precedent for more effective and adaptable document data extraction systems.

The study's insights can inform practitioners, researchers, and developers in refining their approaches to document data extraction, paving the way for improved efficiency, accuracy, and applicability across a spectrum of real-world scenarios. In essence, the significance of this research extends to its potential to shape the future of document analysis, offering practical solutions to challenges in information extraction from a diverse range of documents.

5. SCOPE OF STUDY

The scope of this study encompasses a comprehensive exploration of document data extraction methodologies, with a particular emphasis on the utilization of Large Language Models (LLMs) and zero-shot predictions. The research delves into the diverse landscape of document types, including but not limited to IRS documents, dummy KYC data, emails, and legal agreements such as ISDA documents. By incorporating these varied sources, the study aims to develop a model that transcends document type limitations, showcasing adaptability and effectiveness across a broad spectrum.

The investigation further extends to the creation and analysis of structure templates and instruct documents within the curated datasets. This includes defining frameworks and examples to guide the LLM in accurate extraction, forming a crucial aspect of the study's scope. The evaluation process post zero-shot predictions is an integral component of the study's scope, involving meticulous assessments of accuracy, contextual relevance, and scalability. The study aims to identify not only the strengths but also the limitations of the LLM-based extraction system, providing insights into areas of improvement and refinement.

Overall, the scope of this study reaches into the realms of document analysis, natural language processing, and machine learning, offering a holistic exploration of the capabilities and potential applications of LLMs in document data extraction.

6. RESEARCH METHODOLOGY

6.1 Overview

In its entirety, our pipeline unfolds across five distinct stages: OCR, chunking, document classification, prompt generation, and LLM inference and decoding. The subsequent sections provide a detailed breakdown of each stage.

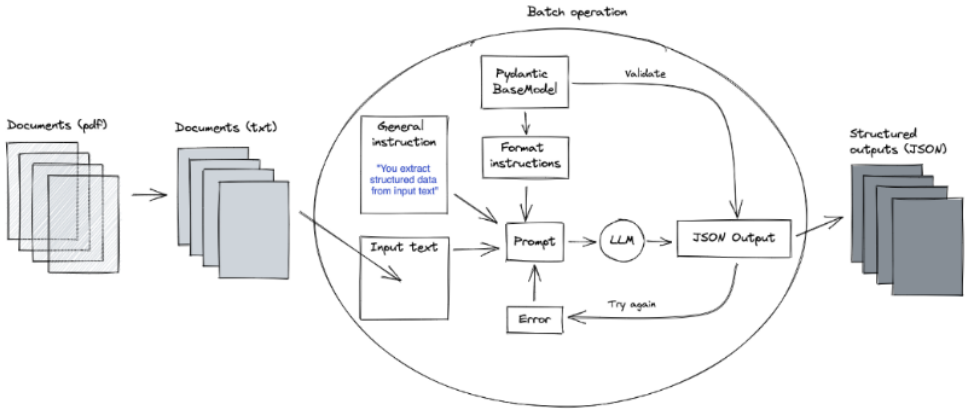


Figure 6.1 Request flow of LLM based Extraction (Yke Rusticus 2023).

6.2 Datasets

Custom datasets sourced from diverse channels, including the IRS, dummy KYC data, emails, ISDA agreements, and contact documents, will be employed in this research. These datasets will be categorized into two primary components: structure templates and unstructured documents. Predefined frameworks, outlining the expected format and organization of the data within documents, will be served by the structure templates. On the other hand, specific examples, or instances from diverse sources, guiding the model on how to interpret and extract information according to the predefined templates, will be provided by unstructured documents. This approach ensures a comprehensive and varied training set, enabling the model to generalize and adapt to a wide range of document types and structures.

6.3 OCR

To transform PDFs or images into machine-readable format, we employ Optical Character Recognition (OCR). Initially, we utilize an off-the-shelf OCR service to analyse the document image, extracting words and line segments while capturing their spatial positions through bounding boxes on the document. A sample output from this stage for a specific document is detailed in Appendix A.6.

In this research, we specifically leverage the Tesseract open-source OCR engine for the conversion process.

6.4 Chunking

To address the challenge of processing arbitrarily long documents within the constraints of LLMs' limited input token length, we employ a chunking strategy. Initially, the document is segmented into individual pages. Subsequently, we iteratively trim the last line segments, originating from OCR, until the prompt associated with each chunk is within the LLM's maximum input token limit. The removed lines are then grouped to form a new document page, and this process is repeated until all chunks adhere to the LLM's input token limit. This approach yields N chunks. The choice to initially partition the document by page is informed by the observation that entities seldom span page boundaries, ensuring minimal impact on the final extraction quality.

6.5 Document Classification and Prompt extraction based on configuration

In the LLM-based extractor, document classification plays a crucial role. Depending on the document category, the configuration retrieves the corresponding prompt and key entities to be extracted. Subsequently, the prompt, along with context, is forwarded to ChatGPT for the extraction process.

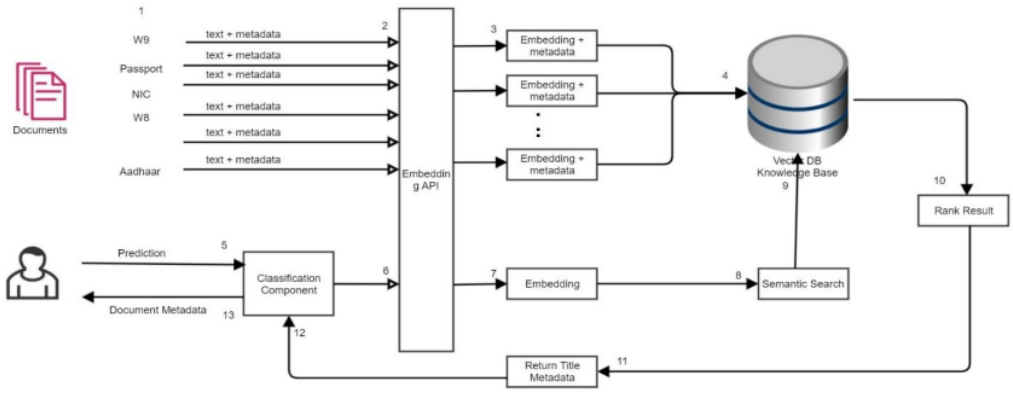


Figure 6.2 Classification based on vector database

The classification process involves:

6.5.1. Data Preprocessing: Segmentation into pages and OCR conversion for machine-readable text.

6.5.2. Document Embedding: Utilizing Sentence Transformer for nuanced understanding and semantic relationships within documents.

6.5.3. Efficient Storage: Storing embeddings and metadata in Vector Databases for organized and quick retrieval.

6.5.4. Indexing for Prediction: Establishing an index in the Vector Database for rapid document retrieval during classification.

6.5.5. Classification Prediction: Using the Sentence Transformer to generate embeddings for query documents and performing a semantic search in the Vector Database for similar documents.

6.5.6. Annotations: The classification engine returns similarity distances and annotations, enriching user experience by providing metadata for the matched document.

By combining chatgpt Embedding and Vector Databases, this approach has the potential to revolutionize document classification, offering improved accuracy, quicker retrieval, and enhanced business outcomes.

6.6 Prompt Representation:

Post-classification, we retrieve a category prompt template containing task details and output schema representation. Subsequently, we construct a prompt in the following format before sending it to the chatbot API for extraction:

```
{DOCUMENT_CONTEXT}
    +
{TASK_DESCRIPTION}
    +
{SCHEMA_REPRESENTATION}
```

6.6.1. Document Context: This refers to the context within the document from which key entities are to be extracted.

6.6.2. Task Description: The task description serves as a concise explanation of the task at hand. In our experiments, we rigidly define it as follows: "From the document, extract the text values and tags of the following entities:"

6.6.3. SCHEMA_REPRESENTATION: The schema is presented as a structured JSON object. Keys represent entity types to be extracted, and values correspond to their occurrences (single or multiple) and potential sub-entities for hierarchical structures.

Certainly! Let us customize the example for extracting names and contacts:

Consider the following schema representation:

```json
{"name": "", "contacts": [{"mobile": ""}]}
```

In this scenario, the LLM is directed to extract a single entity of type "name" and multiple hierarchical entities of type " contact." Each "contact" entity can potentially contain multiple entities of type "mobile."

Now, applying this schema to our prompt template:

```plaintext
{DOCUMENT_CONTEXT}
    +
From the document, extract the text values and tags of the following entities:
    +
{"name": "", "contact": [{"mobile": ""]}}
```

This example prompt instructs the extraction process to focus on capturing names and addresses from the document. The hierarchical structure of the "contact" entity accommodates the possibility of multiple mobile number.

6.7 Model:

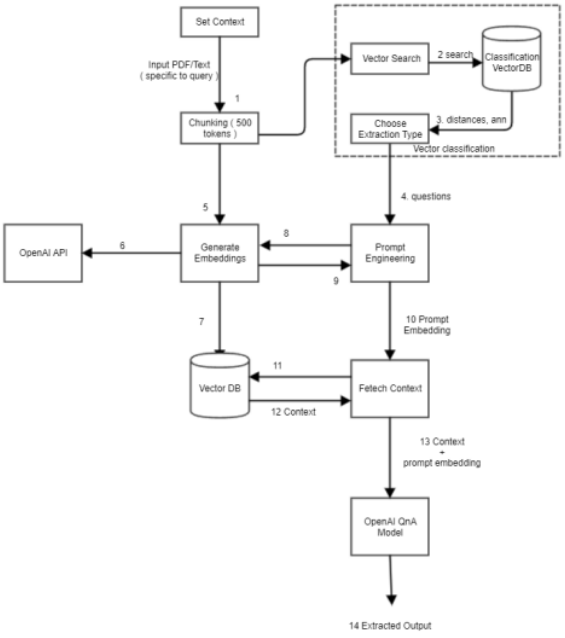


Figure 6.3 Architecture of LLM based extraction

After categorizing the document, we retrieve a corresponding prompt template containing essential elements like the document context, a standardized task description, and a structured schema representation. This template becomes the basis for constructing a specific prompt tailored for the Large Language Model (LLM). The crafted prompt is meticulously designed to incorporate the document context, a detailed task description outlining the entities for extraction, and a schema representation defining the extraction structure. Subsequently, this carefully constructed prompt is sent to the LLM through the chatbot API. In our research, we've opted to utilize OpenAI's ChatGPT for both generating embeddings and executing the extraction process.The LLM then processes the prompt, leveraging its extensive knowledge to comprehend and execute the extraction task. The resulting output undergoes evaluation against the predefined schema representation, ensuring alignment with the specified structure and criteria. This methodology establishes a systematic and personalized interaction with the LLM, ensuring precise and contextually relevant document data extraction within the scope of our research.

Additionally, in this research endeavor, we plan to evaluate the performance of other LLMs such as LLAMA or Google BARD alongside OpenAI's ChatGPT. This comparative analysis aims to provide insights into the effectiveness and suitability of different LLMs for the document data extraction task.

6.8 Evaluation:

After conducting zero-shot predictions, an effective evaluation approach is crucial to assess the model's performance. Here is a general framework you can consider:

6.8.1. Ground Truth Comparison:

- Maintain a ground truth dataset with manually annotated correct extractions for a subset of documents. Conduct a comparison between the model's predictions and the ground truth to evaluate accuracy and pinpoint any discrepancies. This approach aligns with the guidance provided in a paper by (Garg et al. 2022), which offers a comprehensive overview of entity extraction techniques and applications, encompassing diverse methods for evaluating the performance of entity extraction models.

6.8.2. Metric Selection:

- Select evaluation metrics tailored to the nature of your task, with precision, recall, and F1-score being common choices for entity extraction tasks. This approach aligns with the insights provided in a referenced paper, which discusses the relevance of these metrics for named entity recognition (NER) tasks and their applicability to zero-shot extraction tasks involving entity recognition. Additionally, the paper explores various evaluation metrics in the broader context of natural language processing (NLP), emphasizing the importance of choosing metrics based on the specific task and desired information.

6.8.3. Entity-level Evaluation:

- Conduct a thorough assessment of the model's performance on individual extracted entity types, calculating precision, recall, and F1-score for each entity type to discern specific strengths and weaknesses. This paper delivers an in-depth exploration of entity recognition and classification techniques, incorporating diverse evaluation metrics and their practical applications. The evaluation process, focused on each extracted entity type, facilitates the identification of specific areas of strength and weakness in the model's performance (Ward et al., 2011).

6.8.4. Contextual Evaluation:

- Consider the context of the extraction. Evaluate not only individual entity mentions but also the overall coherence and relevance of extracted information within the document.

6.8.5. Generalization Testing:

- Conduct a comprehensive evaluation by testing the model on documents not encountered during training to gauge its generalization capabilities. Investigate the model's adaptability to novel document types or sources. This paper offers insights into domain adaptation techniques for Natural Language Processing (NLP), specifically designed to enhance model performance on new domains distinct from the training data. The generalization testing process facilitates an assessment of the

model's capacity to adapt to diverse data sources and document types.(Daumé III, 2010)

6.8.6. Error Analysis:

- Conduct a thorough error analysis to understand common mistakes made by the model.
- Identify patterns in misclassifications and consider refining the training data or adjusting the model architecture accordingly.

6.8.7. Cross-Validation:

- To ensure robust evaluation results and estimate the generalization ability of models, implement cross-validation by splitting the dataset into multiple folds, training on subsets, and evaluating on the remaining data, as highlighted in this paper providing a comprehensive tutorial on cross-validation techniques (James et al).

6.8.8. User Feedback:

- Gather feedback from end-users or domain experts to glean insights into the practical utility of the model's predictions. Additionally, this guide offers practical advice on establishing model monitoring pipelines to track performance over time and detect potential issues, ensuring the model's continued accuracy and relevance as data and requirements evolve. (Deeplearning.ai)

6.8.9. Scalability Testing:

- Evaluate the model's performance on varying document lengths and complexities to ensure scalability in real-world scenarios.

6.8.10. Continuous Monitoring:

- Implement continuous monitoring of model performance over time. This can include retraining the model with new data to adapt to evolving patterns and challenges.

By combining these strategies, we can obtain a comprehensive understanding of your model's performance and iteratively improve its accuracy and robustness.

7. REQUIRED RESOURCES

7.1 Hardware Requirements:

To effectively conduct this research, the following hardware specifications are essential:

- High-performance computer with substantial processing power (quad-core processor).
- Ample storage capacity (50GB or more) for storing datasets, models, and research outputs.
- Minimum 16GB of RAM to facilitate efficient data processing and model training.
- Dedicated graphics card with a minimum of 4GB of VRAM to accelerate machine learning tasks.

7.2 Software Requirements:

The software stack for this research project includes:

- Python (version 3.8+): The primary programming language for executing machine learning and data analytics tasks.
- Langchain: A language modeling tool that integrates with OpenAI's GPT-3.5 for natural language understanding and generation.
- OpenAI GPT-3.5 Access: Essential for leveraging the advanced capabilities of the GPT-3.5 model in document data extraction.
- React JS: A JavaScript library for building user interfaces. In this research, React JS is utilized for developing a user-friendly interface to configure prompts against titles and create a knowledge base for embedding-based classification.
- TensorFlow (version 2.8+): A popular open-source library for deep learning and machine learning activities.
- Matplotlib (version 3.7.1+): A visualization tool within Python, crucial for creating insightful data visualizations.
- Seaborn (version 0.11.2+): A Python-centric data visualization toolkit that complements Matplotlib.
- Pandas (version 2.0.2+): A Python tool designed for efficient data processing, manipulation, and analysis.
- Numpy (version 1.24.2+): A Python-based computational library essential for numerical operations.
- Jupyter Notebook (version 6.1+): An interactive platform compatible with Python, facilitating collaborative and exploratory research.
- Git (version 2.29+): A version control system to oversee and coordinate changes to both code and data throughout the study.

7.3 Additional Software:

Depending on the evolving needs of the research, additional software tools and libraries may be incorporated for specific tasks or model complexities.

7.4 Cloud-Based GPU Resources:

Access to GPU-based platforms such as Google Colab for efficient model training and fine-tuning processes, ensuring the optimization of GPT-3.5 for document data extraction.

8. RESEARCH PLAN

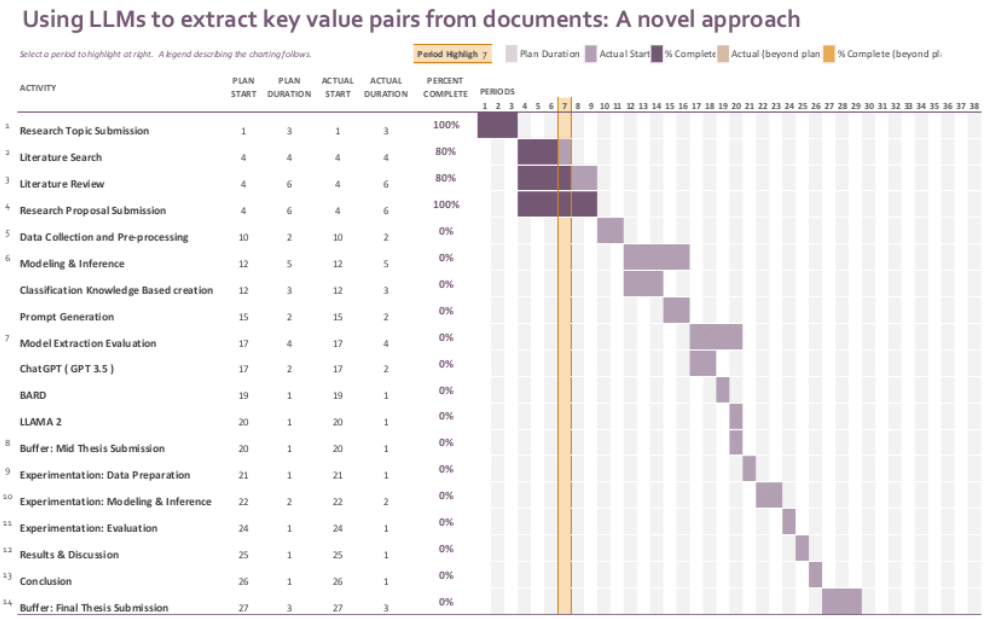


Figure 8.1 Project planning and timeline

Note: 1 Period = I Calendar Week

REFERENCES

Yke Rusticus (2023). How to Extract Structured Data from Unstructured Text using LLMs. Xebia. Retrieved from https://xebia.com/blog/archetype-llm-batch-use-case/

Perot, V., Kang, K., Luisier, F., Su, G., Sun, X., Boppana, R. S., Wang, Z., Mu, J., Zhang, H., & Hua, N. (2023). LMDX: Language Model-based Document Information Extraction and Localization. arXiv preprint arXiv:2309.10952.

Polak, M. P., & Morgan, D. (2023). Extracting Accurate Materials Data from Research Papers with Conversational Language Models and Prompt Engineering. arXiv preprint arXiv:2303.05352.

Ozdayi, M. S., Peris, C., Fitzgerald, J., Dupuy, C., Majmudar, J., Khan, H., Parikh, R., & Gupta, R. (2023). Controlling the Extraction of Memorized Data from Large Language Models via Prompt-Tuning. arXiv preprint arXiv:2305.11759.

Dunn, A., Dagdelen, J., Walker, N., Lee, S., Rosen, A. S., Ceder, G., Persson, K., & Jain, A. (2022). Structured information extraction from complex scientific text with fine-tuned large language models. arXiv preprint arXiv:2212.05238.

Kartchner, D., Al-Hussaini, I., & Kronick, O. (2023). Zero-Shot Information Extraction for Clinical Meta-Analysis using Large Language Models. ACL Anthology, 49th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (ACL-IJCNLP 2023).

Trajanoska, M., Stojanov, R., & Trajanov, D. (2023). Enhancing Knowledge Graph Construction Using Large Language Models. In 2023 International Conference on Computer and Information Science (ICICIS) (pp. 1-4). IEEE.

Laskar, M. T. R., Bari, M. S., Rahman, M., Bhuiyan, M. A. H., Joty, S., & Huang, J. X. (2023). A Systematic Study and Comprehensive Evaluation of ChatGPT on Benchmark Datasets. arXiv preprint arXiv:2305.18486.

Dunn, A., Dagdelen, J., Walker, N., Lee, S., Rosen, A. S., Ceder, G., Persson, K., & Jain, A. (2022). Structured information extraction from complex scientific text with fine-tuned large language models. arXiv preprint arXiv:2212.05238.

Arroyo, J., Corea, F., Jiménez-Díaz, G. and Recio-García, J.A., (2019) Assessment of Machine Learning Performance for Decision Support in Venture Capital Investments. IEEE Access, 7, pp.124233–124243.

Garg, A., Srivastava, D., Xu, Z., & Huang, L. (2022). Identifying and Measuring Token-Level Sentiment Bias in Pre-trained Language Models with Prompts. arXiv preprint arXiv:2204.07289.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning: with applications in R. Springer.

Deeplearning.ai. (2023). A Practical Guide to Model Monitoring. Deeplearning.ai.

Ward, C. B., Choi, Y., Skiena, S., & Xavier, E. C. (2018). Empath: A framework for evaluating entity-level sentiment analysis. IEEE Transactions on Knowledge and Data Engineering, 30(10), 1729-1745.

Daumé III, Hal, Abhishek Kumar and Avishek Saha (2010), 'Frustratingly easy semi-supervised domain adaptation', In Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing,

Pramod Gupta - Thesis.pdf

ORIGINALITY REPORT

1%	0%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Liverpool John Moores University Student Paper	1%

Exclude quotes	Off	Exclude matches	< 1%
Exclude bibliography	On		