UNIVERSITY OF
MODENA AND REGGIO EMILIA

Doctor of Philosophy in
"Information and Communication Technologies (ICT)"

Cycle XXXVIII

Multimodal Document Understanding for LLMs

Candidate: Luca De Grandis

Supervisor (Tutor): Prof. Lorenzo Baraldi

Co-Supervisor (Co-Tutor): Doc. Davide Costa

Coordinator of the Doctoral Course: Prof. Luigi Rovati

Review committee composed of:

Prof. Filippo Furfaro, University of Calabria
Prof. Giuseppe Serra, University of Udine

To my brother.

Contents

List of Figures

List of Tables

List of Abbreviations

– A –

AI	Artificial Intelligence	151
ARES	Automated RAG Evaluation System	98
ATS	Automatic Text Summarization	20

– B –

BEM	BERT Matching	103
BERT	Bidirectional Encoder Representations from Transformers	98
BLEU	Bilingual Evaluation Understudy	33
BLEURT	Bilingual Evaluation Understudy with Representations from Transformers	36
BPE	Byte-Pair Encoding	13

– C –

CoE	Chain of Extractions	61
CLM	Causal Language Modeling	13
CNN	Convolutional Neural Network	17
CoD	Chain of Density	61
CoT	Chain of Thoughts	15
CV	Computer Vision	3

– M –

– N –

– O –

– P –

– Q –

– R –

– S –

– T –

– V –

Chapter 1

Introduction

Over the last decade, the uncontrollable growth of unstructured data utilization has fundamentally reshaped how companies store and access information. Enterprises increasingly rely on documents containing multi-modal information sources such as text, tables, charts, figures, and complex layout structures. A key element for automation is knowing how to access and link information from unstructured and multi-modal sources. Regardless of advances in Machine Learning (ML) and Natural Language Processing (NLP), reliable understanding of multi-modal documents remains challenging. This thesis addresses challenges in developing and evaluating systems for multi-modal and explainable document understanding, with a focus on summarization and context attribution from multi-modal sources.

1.1 Context and Motivations

In recent years, companies have witnessed a dramatic increase in the volume of unstructured data they manage. Contracts, Environmental, Social, and Governance (ESG), manuals, financial reports, slides, and many more, contain textual paragraphs along with tables, charts, and figures, all structured in complex layouts. Together, these elements encode essential information, which would be incomplete if isolated. Intelligent Document Processing (IDP) seeks to automate the extraction, interpretation, and utilization of this content. relying mostly on Machine Learning models and techniques. Among the most utilized: Document Layout Analysis (DLA), Optical Character Recognition (OCR), and LM (Language

Models). As the scale and complexity of documents grow, challenges become more prominent. Models become less efficient, frequent errors undermine real-world applications, and the lack of approaches render some tasks intractable. Long context processing remains particularly demanding, often leading to performance degradation. Finally, the multi-modal nature of documents requires models capable of understanding structured data, visual reasoning, and cross-modal alignment.

Despite advances in summarization, mimicking human abilities remains particularly challenging. When humans write summaries, they look at the entire document, discern relevant information from irrelevant content, synthesize knowledge from non-textual sources, rearrange the document order, and abridge most of the available content. Conversely, current models struggle with global context retention and hallucinations. Motivated by the difficulties of document summarization, the first part of this dissertation describes a series of experiments conducted to evaluate new technologies' capabilities in summarization and processing of multi-modal information. In Chapter 3.5 we evaluate current technologies for the interpretation of knowledge from structured tables. In Chapter 3.2 we delve into the multi-modal issue and test recent ML models to translate complex charts into structured tables. Chapter 3.3 and Chapter 3.5 details the evaluation of current models in document summarization, both generic and aspect-based. Then, in Chapter 4 we merge our findings into the development of a modular summarizer, with multi-modal inputs and outputs. Finally, Chapter 5 reports our findings on reference-free and reference-based LLM-based evaluation of LLM-generated answers.

Despite recent advancements in summarization and Question Answering (QA), user trust remains low due to the systematic lack of mechanism for quality verification. To allow content verifiability and reproduction, humans insert numerous citations within their dissertations, official documents, and academic manuscripts. Machine Learning models, on the other hand, mostly function as black boxes, offering limited to no transparency to the content's source and interpretation. This intrinsic characteristic undermines users trust and limits our ability to verify the content and provide feedback for further improvements. To address these challenges, Chapter 6 describes a model agnostic technique to obtain text-based attributions and Chapter 7 details a novel approach for the automatic development of a visual-grounding dataset at scale.

1.2 The Industrial Scenario

This thesis is contextual to the industrial environment of Altilia.ai, an Italian company focused on large-scale document automation solutions and the founder of this research. This company faces the daily challenge of processing diverse collections of documents originating from several domains with widely heterogeneous levels of structure, noise, and visual complexity. With clients in the financial domain and currently expanding in the global market, Altilia requires new solutions for Information Extraction (IE), Question Answering, Automatic Summarization, and synthetic data generation. Moreover, with the increasingly fast development of powerful Large Language Models (LLMs), the company faces the increasing need to evaluate and deploy LLM-based solutions at scale with minimal costs.

At its core, Altilia's solutions leverage Language Models and Computer Vision (CV) models for Information Extraction, dealing with tasks like Text Classification, Named Entity Recognition (NER), and Object Detection. Recently, the company's platform has been leaning toward LLMs and MLLMs (Multi-Modal Large Language Models) to address these challenges. However, the deployment of these new systems requires attention to reliability, transparency, and explainability. This comes with challenges. First, new LLM-based systems capabilities are largely unknown in many industrial domains, due to limited evaluations in the scientific literature and lack of sector-specific benchmarks. Second, even with sufficient scientific literature, applications to real-world scenarios are often challenging. Third, final users are not empowered with information to evaluate the systems, reducing trust and applicability. Finally, new models applicability for dataset generation require non trivial methodologies and complex evaluations before deployment in production environments.

Given these industrial constraints, this research deliberately prioritizes LLM- and Retrieval-Augmented Generation (RAG)-based architectures over traditional, task-specific supervised models. This methodological choice is driven by the need for adaptability: traditional models often struggle to generalize across heterogeneous document layouts without prohibitive data annotation costs, whereas LLMs offer the zero-shot capabilities required to scale across diverse client domains. Furthermore, incorporating RAG directly answers the critical need for transparency in sectors like finance, providing a structural mechanism for the attribution techniques required to trace outputs back to their multi-modal sources.

In this context, this research aims at the development of robust end-to-end pipeline for processing long, multi-modal documents, with a deep focus on explainability. Specifically, we aim to: (i) evaluate methods for processing multi-modal

inputs (text, tables, charts), (ii) build a multi-modal summarization architecture with multi-modal outputs capabilities, (iii) propose novel attribution techniques to identify the sources supporting the output generated by MLLMs, and (iv) reduce reliance on manual annotation through model- and architecture-level innovations.

1.3 Activities Carried out During the PhD

Being funded by a company and covering the role a full-time employee, most of the work conducted during the past years is not fully reported in this thesis. There are varying infra-project reasons for this. First, the industrial scenario requires solutions to be promptly ready in production environments, limiting time availability for extensive experiments for literature production. Second, the industrial nature of the doctorate prioritized production-readiness and immediate client deliverables. Consequently resource allocation was often directed toward deployment rather than experimental research.

1.3.1 Industrial Research & Development Projects

The following list is comprehensive of all the projects the candidate participated in during the PhD period. However, the list doesn't include the projects that led to this manuscript, which are left for the reader to explore.

- **Information Extraction (IE):** Contributed to annotation, fine-tuning, and deployment of ML models for Information Extraction as part of the company's day-to-day operations.
- **Table Structure Recognition (TSR):** Researched and integrated state-of-the-art TSR models to automate the extraction of complex tabular data from unstructured documents.
- **Document Layout Analysis (DLA):** Evaluated and implemented DLA frameworks to improve the segmentation and classification of multi-page PDF components.
- **ESG Attribution System:** Engineered a fine-grained, LLM-based attribution framework for fine-grained attribution in automatic ESG reporting, enabling precise evidence mapping between textual claims and source data.

- **Encoder-based Visual Annotation Engine:** Architected Altilia's first proprietary encoder-based system for visual document labeling, empowering users with automatic annotations and the first verifiability tool.
- **Legal & Financial RAG:** Contribution to the development of a Retrieval Augmented Generation (RAG) system specialized on the legal and financial domains, optimizing retrieval accuracy for high-stakes QA tasks.
- **MLLM-based Visual Annotation Engine:** Architected Altilia's second and improved MLLM-based visual document labeling engine, providing a comprehensive mechanism for semi-automatic annotation of text and visual elements for structured documents.

Conferences Attended

- **DEXA (2024):** 35th International Conference on Database and Expert Systems Applications.
- **Ital-IA (2024):** 4th National Conference of Italian Artificial Intelligence.
- **ECAI (2025):** 28th European Conference on Artificial Intelligence.

Contributions

- **Sample-Efficient Fine-Tuning for Table-to-Text Generation:** Demonstrated that lightweight fine-tuning (LoRA) of smaller open-source models on highly restricted datasets (1,000 samples) is sufficient to bridge the scale gap, achieving competitive performance against SOTA techniques and massive proprietary models (Detailed in Chapter 3.1 and published in DEXA 2024 [158]).
- **Impact of Table Serialization and Reduction:** Conducted a comprehensive empirical evaluation of table input representations (HTML, JSON, plain text) on LLM generation quality. Furthermore, demonstrated that applying structural table reduction strategies significantly mitigates performance degradation on complex, large-scale tables like ToTTo (Detailed in Chapter 3.1 and published in DEXA 2024 [158]).
- **Operational Cost-Benefit Analysis of LLM Deployment:** Provided a quantitative assessment of inference times and deployment costs, revealing

the counter-intuitive finding that proprietary APIs can currently be more cost-effective and yield higher throughput than deploying open-source adapters on rented infrastructure (Detailed in Chapter 3.1 and published in DEXA 2024 [158]).

- **Comprehensive Benchmarking of C2T Models:** Conducted a systematic evaluation of proprietary and open-source MLLMs across 18 distinct chart topologies. Revealed that while open-source models can rival proprietary baselines on general charts, all current architectures exhibit catastrophic performance degradation on dense, spatially complex visualizations (Detailed in Chapter 3.2).
- **Assessment of Benchmark Discrepancy in Real-World Industrial Scenarios:** Demonstrated a critical discrepancy between model performance on curated academic benchmarks (ChartX) and real-world applicability using complex ESG reporting documents. Proved that current SOTA models suffer from severe extrinsic hallucinations and fail to infer continuous values from discrete axes, concluding that human-in-the-loop validation remains mandatory for enterprise deployment (Detailed in Chapter 3.2).
- **Exposing the Discrepancy Between Syntactic Evaluation and Semantic Fidelity:** Demonstrated that traditional n-gram overlap metrics are insufficient for evaluating modern abstractive summarization. The empirical evaluation proved that while legacy fine-tuned models achieve state-of-the-art lexical overlap, they suffer from poor factual consistency, whereas proprietary LLMs generate highly faithful and semantically accurate synopses despite lower syntactic scores (Detailed in Chapter 3.3).
- **Cost-Benefit Analysis of Summarization Prompting Strategies:** Systematically evaluated advanced prompt engineering techniques, including Chain of Density (CoD) and Chain of Extractions (CoE). Revealed that highly complex reasoning chains introduce generation instability, high variance in abstractiveness, and increased API costs without proportional quality gains, proving that simple task- and length-constrained prompts remain the most optimal and reliable approach for production environments (Detailed in Chapter 3.3).
- **Impact of Domain Homogeneity on In-Context Learning:** Demonstrated that the efficacy of few-shot prompting in aspect-based summarization is strictly dependent on the structural homogeneity of the target domain.

Proved empirically that while few-shot examples significantly enhance alignment and performance on structured data, they introduce semantic noise and degrade generation quality on heterogeneous, high-variance datasets, where zero-shot inference is far more robust (Detailed in Chapter 3.5).

- **Robustness vs. Adaptability Trade-off in LLMs:** Conducted a comparative evaluation between proprietary models and local, open-source alternatives for conditioned text generation. Identified a clear architectural trade-off: massive proprietary models exhibit superior zero-shot robustness and instruction-following, whereas smaller open-source models demonstrate superior few-shot adaptability, ultimately outperforming their larger counterparts when provided with consistent in-context templates (Detailed in Chapter 3.5).
- **A Modular Framework for Multimodal Summarization with Multimodal Outputs:** Designed and evaluated an end-to-end pipeline tailored for long, complex financial documents. This framework introduces a conversion process, combining layout-aware parsing, Table-of-Contents-driven header filtering, and graphical element augmentation (Detailed in Chapter 4).
- **Exposing Formatting Bias in LLM-as-a-Judge Evaluation Protocols:** Conducted a critical evaluation of modern reference-free LLM metrics on long-document summarization. Empirically proved that these evaluators exhibit extreme sensitivity to input formatting; merely cleaning or altering the markdown structure artificially inflates or deflates factual consistency scores (Detailed in Chapter 4).
- **Critical Assessment of LLM-as-a-Judge for RAG Evaluation:** Conducted a rigorous correlational study between automated RAG evaluation frameworks and human judgment. Revealed a critical divergence: while automated metrics correlate strongly with humans on reference-based correctness, they exhibit significant degradation in reference-free scenarios, proving that current automated frameworks cannot reliably evaluate open-ended retrieval without ground truth data (Detailed in Chapter 5 and published in DEXA 2024 [159]).
- **Lightweight Cross-Encoders as SOTA Alternatives for Context Attribution:** Demonstrated that small cross-encoder architectures, when paired with optimized sliding-window strategies, can match or exceed the accuracy of massive proprietary LLMs for both coarse and fine-grained context

attribution. This proves that complex dependency management in RAG verification does not strictly require massive generative models (Detailed in Chapter X and published in ECAI 2025 [50]).

- **Mask-based Perplexity-Derived Attribution (MAPPET) Framework:** Proposed a novel, highly scalable pipeline for automatic visual attribution in multimodal datasets. By measuring the shift in a MLLM perplexity when specific visual regions are masked, this method successfully isolates high-confidence, fine-grained evidence bounding boxes, entirely bypassing the bottleneck of manual human annotation (Detailed in Chapter 7 and under review at CVPR 2026).

- **Introduction of DocAttriBench (DAB):** Created and open-sourced the first large-scale benchmark dataset explicitly designed for Visual Answer Grounding in structured documents. By aggregating, harmonizing, and automatically annotating eight diverse public datasets, DAB provides the research community with nearly 300,000 high-quality, region-grounded QA pairs for robust training and standardized evaluation (Detailed in Chapter 7 and under review at CVPR 2026).

- **Comprehensive MLLM Benchmarking for VAG:** Designed a rigorous three-task evaluation protocol (grounded generation, post-hoc grounding, and answer localization) to assess spatial and semantic reasoning jointly. Conducted a large-scale evaluation of 16 state-of-the-art MLLMs, exposing a critical industry-wide gap between textual accuracy and spatial localization. Finally, empirically proved that fine-tuning baseline models on DAB effectively bridges this gap, establishing a new baseline for verifiable document understanding (Detailed in Chapter 7 and under review at CVPR 2026).

Literature Production

- Francesco Maria Granata, Luca De Grandis, Antonio Lanza, Amir Bachir, Ermelinda Oro, Massimo Ruffolo. Evaluating retrieval-augmented generation for question answering with large language models. In: *CEUR Workshop Proceedings*, 2024, 129-134.

- Luca De Grandis, Francesco Maria Granata, Ermelinda Oro, Massimo Ruffolo. Leveraging Large Language Models for Flexible and Robust Table-

to-Text Generation. In: *International Conference on Database and Expert Systems Applications*. 2024, 222-227.

- Luca De Grandis, Franceso Maria Granata, Davide Costa, Antonio Lanza, Ermelinda Oro. Improving Context-Attribution with Semi-Supervised Cross-Encoders. In: *Frontiers in Artificial Intelligence and Applications*. 2025.
- Luca De Grandis, William Raccagni, Silvia Cappelletti, Marcella Cornia, Lorenzo Baraldi. DocAttriBench: Benchmarking Answer Grounding in Document Visual Question Answering. *Under review (CVPR 2026).*

Chapter 2

Theoretical Background

The purpose of this chapter is to provide the reader with the information required to understand the remainder of the manuscript and to comprehend most of the design choices done in the subsequent chapters. In Sec. 2.1 we describe the foundation of sequence to sequence learning. Sec. 2.2 discusses the main approaches to represent multi-modal data for neural networks. In Sec. 2.3 we report the main summarization types and we discuss the problem of hallucinations. Sec. 2.4 discusses the issue of faithfulness in LLM-generated content and the most common approaches for attributions. Finally, Sec. 2.5 explains the context in which the PhD was done, with company details and research constraints.

2.1 Foundation of Sequence-to-Sequence Learning

Modern Artificial Intelligence was defined by a shifting paradigm from task-specific architectures to general-purpose foundational models. The transition was dictated by the convergence of Natural Language Processing and Computer Vision models under the unified framework of the Transformer.

2.1.1 The Transformer Architecture

Before 2017, State-of-The-Art (SOTA) Natural Language Processing relied mostly on the Recurrent Neural Network (RNN) [58] and Long Short Term Memory (LSTM) [85] architectures for a variety of Machine Learning tasks. Although

attention mechanisms alleviated some issues of sequential processing, typical of these architectures, they suffered from two major limitations: sequential bottleneck and context compression. *Sequential bottleneck* implicates that each token's hidden state depended on previous tokens' hidden states, preventing parallelization across training examples and making learning from large-scale datasets inefficient and slow. *Context compression* means that, in order to process long sequences, content was compressed into a single hidden state of fixed-size. Workarounds (bidirectional RNNs [184], hierarchical RNNs [84], attention modules [14, 76, 77]) further limited scalability and modeling power.

The Transformer [213] currently represents the most prominent innovation in the field, effectively replacing recurrence and convolutions with self-attention. This shift addresses two critical limitations from previous architectures. First, it eliminates the sequential bottleneck by processing all tokens at once, in parallel rather than iteratively. Second, the Transformer avoids context compression with self-attention, letting each new token attend to all other tokens at once and with limited information loss.

Self-Attention and Multi-Head Attention. The mathematical core of the transformer is the *Scaled Dot-Product Attention* [213]. Given a sequence of input tokens X, the model learns three projection matrices: the query matrix Q, the key matrix K, and the value matrix V. The attention is computed as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

where $\sqrt{d_k}$ the scaling factor, preventing the dot product from growing and preventing the vanishing gradients problem happening when the softmax is pushed into regions of excessively small gradients. To capture diverse relationships within the data, *Multi-Head Attention* (MHA) runs multiple Scaled Dot-Product Attention operations in parallel, forcing the model to learn multiple projection matrices.

Architecture The Transformer relies on attention as well as other components. First above all is *Positional Encoding*. The Transformer is permutation invariant, so positional information is injected to provide order awareness. The original transformer relied on sinusoidal encoding but more recent implementations use alternative approaches such as learnable positional embeddings [71], rotary positional embeddings [196], and relative positional embeddings [188]. *Residual*

Connections & Layer Normalization are utilized to stabilize training of deep networks and mitigate the vanishing gradients issue. These are heavily used in the Transformer too. Finally, position-wise *Feed Forward Network*s (FFNs) are used to process each token independently, passing through non-linearity layers such as GELU [82].

Attention operations carry quadratic computational complexity $O(N^2)$ with respect to the sequence length. This poses challenges to processing of extremely long documents and high-resolution images. The concept of *"long document"* and *"high-resolution image"* have been changing drastically in the last few-years due to the development of highly optimized implementations such as Flash Attention [48, 49], Paged Attention [105], Sparse Attention [19, 40], and Kernelized Approximations [97].

2.1.2 Shifting Pre-Training Paradigm

The transition from Traditional LMs to modern LLMs was driven by the shift from supervised, task-specific training to large-scale self-supervised pre-training strategies.

Tokenization. Before encoding, raw text is decomposed into discrete units with off-the-shelf sub-word tokenization algorithms like Byte-Pair Encoding (BPE) [187] or SentencePiece [102, 103]. These methods map text into finite vocabularies and are optimized to balance the granularity of character-level models with the efficiency of word-level models, effectively handling out-of-vocabulary terms.

Training Stack LLM development follows a multi-stage development stack. First, there is *Pre-Training*, consisting in training models on trillions of tokens using the Causal Language Modeling (CLM) objective [171], i.e. auto-regressively predicting the next token in a sequence. Then, the model undergoes *Supervised Fine-Tuning*, refining the model on curated instruction-response pairs to learn how to follow human intent and instructions. Finally, *Preference Optimization* is used through techniques such as Reinforcement Learning from Human Feedback (RLHF) [43, 160] and Direct Policy Optimization (DPO) [174], aligning model outputs with human values and safety constraints.

Scaling Laws and Emergence Research into scaling laws [86, 96, 164] has shown that model performance improves predictably as a function of compute, data

size, and parameter count. Crucially, as models scale beyond certain thresholds, they exhibit emergent abilities like arithmetic or logical reasoning that were not explicitly present in smaller-scale versions.

2.1.3 From Transfer-Learning to Prompting

In its final stages, LLMs moved from fine-tuning for every new tasks to zero- and few-shot prompting techniques [28]. This shifting technical paradigm allows users to generalize models to new tasks and domains simply providing clear instructions and a few well written examples.

Model Archetypes Depending on the objective, Transformers are utilized in one of three configurations. *Encoder-only* models (e.g. BERT [54]) are optimized for text understanding and classification using bi-directional attention. *Encoder-Decoder* models (e.g. T5 [175]) are designed for sequence-to-sequence tasks like translation. *Decoder-only* models (e.g. GPT [171]) are employed for generative tasks and rely on auto-regressive techniques.

Multi-Modal Integration MLLMs extend the generative power of LLMs to visual data. These systems typically consist of a *Vision Encoder* (often a Vision Transformer) and a *Language Backbone* (e.g. an LLM itself). A learnable adapter or projector aligns visual features with the LLM's text embedding space. This allows the model to treat visual patches as "visual tokens," enabling joint reasoning over text and images for tasks like Document Understanding (DU), Visual Question Answering (VQA), and complex scene description. Despite their power, these models remain subject to hallucinations [95, 248] and the high computational cost associated with long-context inputs. However, they represent a significant step toward general-purpose agents capable of interpreting the world through both language and vision.

Prompt Engineering and Few-Shot Learning Utilizing LLMs comes with many limitations. First, fine-tuning LLMs requires capable GPUs with high memory capacity and is often subject to long distributed training. Second, dataset curation can be challenging, with issues related to continual learning [189] and catastrophic forgetting [132]. Second, model deployment requires dedicated hardware and it is often impossible to deploy multiple models in the same hardware.

The rise of foundational models has fundamentally shifted the paradigm from fine-tuning to prompt engineering. Through prompt engineering and in-context learning, it is possible to adapt a model to a new task at inference time without updating its parameters. This allows cost-effective customization, especially in low-resource scenarios. Various prompting techniques can be used to improve model efficacy from zero-shot prompting. The most important being few-shot learning [28], which provides a set of input-output demonstrations in the context window to guide the model's output, and Chain of Thoughts (CoT) [219], which encourages models to generate reasoning before answering.

2.2 Neural Representations of Multi-Modal Data

Representing heterogeneous data remains a non-trivial challenge in the development of unified document understanding systems. Modern architectures utilize various strategies to reconcile discrete text with continuous and spatial information to enable reasoning across modalities.

2.2.1 Linearization Strategies

Over time, treatment of structured data has changed significantly. Structured data is often treated through *linearization*, i.e. the mapping of a multi-dimensional structure into a 1D sequence. Early approaches relied on *Natural Language Templates* to convert table cells into sentences containing both the row and column headers [34]. This approach leverages the pre-training knowledge embedded in LMs but introduces significant token overhead. On the other hand of the spectrum, to prevent loss of structural information, other approaches relied on model fine-tuning. One strategy consists in the injection of *coordinate embeddings*. This approach allows the self-attention mechanism to discern between rows and columns and to reason on intra-row and intra-column dependencies [229]. Other relied on reconstruction table-tailored pre-training strategies, effectively learning the table structure from linearized data [52]. More recently, LLMs allowed the utilization of the model's internal knowledge and few-shot techniques to interpret and serialize tables. Researchers found that, while template-based serialization can be useful for few-shot learning with LLMs, it often comes at the cost of hallucinations [81, 160, 183]. More recent approaches utilize *Markup-based Linearization* (HTML, XML) and *Programmatic Formats* (JSONL, DFLoader). In these cases, the tables are kept into rich structured formats while special tokens and tags denote structural boundaries. Research shows that structured formats can improve performance for autoregressive models at the cost of increased token counts [192, 200, 201].

2.2.2 Visual Language Alignment

Differently from structured data, images are not easily associated with ordered information. Instead, they require ad-hoc processing. There, the major bottleneck arises from the fundamental information density between text and images, with sentences containing dozens of tokens and images containing thousands of pixels. CLIP [172] has fundamentally altered images treatment by simultaneously training

separate image and text encoders on large-scale datasets. The alignment is generally done through a contrastive loss function, maximizing the cosine similarity between matched image-text pairs and maximizing it for all other pairs. Through the dual encoder paradigm, visual concepts are grounded in natural language supervision, enabling zero-shot capabilities previously unattainable. Concurrently, the Convolutional Neural Network (CNN) [108] architecture was replaced by the Vision Transformer (ViT) [56], allowing the processing of images through sequences of equally sized patches.

Modern vision-language alignment is achieved through three main components: the Vision Encoder, the LLM Backbone, and the Connector. First, the *Vision Encoder* processes the visual cues from the input image. The encoder is typically derived from architectures like CLIP, SigLIP [210, 238], and ViT. The *LLM Backbone* is generally taken from the vast pool of Language Models available in the open-source see. Finally, the *Connector* is a projection module, translating visual features into the LLM's text embedding space without the need to expensively pre-train the entire model from scratch. Generally, three main approaches are used to build the connector. The simplest is a Multi-Layer Perceptron (MLP) module that maps the visual features directly in the LLM embedding space. This preserves spatial details but increases the token count. Alternatively, a *Q-Former* [115] uses learnable query vectors to extract visual features through cross-attention. The visual inputs are compressed into a fixed number of tokens, filtering only the information that is most relevant to the text. Finally, the *Gated Cross-Attention* [6] approach utilizes cross-attention layers to let the language model attend to the visual tokens dynamically during the encoding phase.

2.3 Document Summarization

Document summarization is the process of distilling the most important information from a document. Over the years, summarization has undergone radical transformation, especially after the advent of LLMs. This chapter details a formal taxonomy for the field, categorizing input complexity, output nature, and algorithmic paradigm.

2.3.1 Summarization Taxonomy

Summarization is classified through many fuzzy variables. Here, we report the most important classifying distinctions.

Summarization by Output Type. The most prominent distinction is related to the output type. *Extractive Summarization* consists in the identification and reporting of the most salient information from the source document, often in the form of sentences from the input document itself [144, 240]. *Abstractive Summarization* involves paraphrasing the most important information into a human-like summary [2, 106, 126, 170, 185, 242]. While being more human-like, this approach is prone to hallucinations. Finally, *Hybrid Approaches* utilize extractive methods to identify the most salient information and abstractive modules to obtain human-like summaries, effectively maintaining high fidelity while generating high quality summaries [104, 240].

Summarization by Input Complexity. We define *Single-Document Summarization* when information must be synthesized from a single source and *Multi-Document Summarization* when multiple sources concur to be part of the same synopses [104, 116, 227]. The first is relatively simple and only necessitates the identification of important information and attention to hallucinations. The latter necessitates removal of redundancies and resolution of conflicting information. Moreover, we distinguish between *Short-Document Summarization* and *Long-Document Summarization* when the existing approaches don't allow to effectively or efficiently deal with a single document due to its excessive length [167]. Due to the rapid increments of LLMs context-length, this distinction is not absolute but contextual. Often, summarization of long documents requires techniques like Hierarchical Summarization [221], Map Reduce [254], and Retrieval Augmented Generation [10].

Summarization by Output Content. We refer to *Generic Summarization* when the generated synopses only contain the most important information from the source documents. Depending on the final user, generic summaries might contain superfluous knowledge and lack important details. For this reason, we refer to *Aspect-Based Summarization* when the generated summary only contains information referring to a specific spectrum of the input document content [74, 230]. The latter is often found under different names and ramified into different settings, all requiring to include some information and exclude irrelevant content on the basis of user-specific requirements. Finally, *Controllable Summarization* refers to the task of giving the output summary a specific length and structure [63, 80, 211].

Summarization by Modality. We distinguish between *Uni-Modal Summarization*, when the input is only made of a single modality, and *Multi-Modal Summarization*, when the input is constituted by many modalities conveying same or different information [256]. Often, Uni-Modal Summarization is based on the text modality, while Multi-Modal Summarization requires the ability to process text as well as other means like images, videos, and audio files. Finally, the output modality can vary too, requiring the creation of a *Multi-Modal Output Summary*, consisting of text as well as tables and images, either created by the system or taken from the input document. This last approach to summarization is rarely dealt with due to the limited dataset availability and current architectural limitations.

Summarization by Algorithmic Paradigm Finally, the last distinction pertains to the model utilized to produce the summary. Early methods relied *Rule-based Models*, employing heuristic algorithms [144], and Latent Semantic Analysis (LSA) [143] to identify key segments from the source documents. More advanced extractive methods relied of *Classification Models*, using Transformers [124, 145] or other learnable architectures [38, 151, 165, 178] to classify sentences from the source material and using them to construct a bullet-points summary. Finally, more recent approaches employ *Autoregressive Models* to generate abstractive summaries. In this case models can be either encoder-decoder models [175, 185, 242], or decoder-only (generative) models [2, 74, 104, 128, 230].

2.4 Faithfulness and Interpretability

As Large Language Models have achieved near-human fluency, the primary research challenge has shifted from linguistic coherence from semantic accuracy. Specifically, the shift has moved the research interest to faithfulness and interpretability. Extractive summarization inherently preserves faithfulness by copying text segmented. Instead, abstractive models function as black boxes, generating fluid sentences that may semantically diverge from the source. Consequently, interpretability serves as a diagnostic tool, allowing the user to evaluate the summary content through attribution of the generated content to the source input.

2.4.1 Hallucinations

A primary challenge in the deployment of Automatic Text Summarization (ATS) systems is the hallucinations phenomenon. Given some source material, *"hallucination"* generally refers to the generation of nonsensical or unfaithful content [95]. While overlap-based and semantic metrics are generally sufficient to measure lexical overlap, they fail to capture semantical inconsistencies, allowing high metric values while generating unfaithful content. Earlier definitions focused strictly on faithfulness to the source text but the modern definition, evolved with LLMs, includes deviations from established world knowledge.

Hallucination Taxonomy

Relationship with the Source Content. The most foundational categorization divides hallucinations based on their relationship to the source text. As such, this categorization is bounded to in-context learning. *Intrinsic Hallucinations* occur when the generated text contradicts the source content [95]. An example of intrinsic hallucination is when the model connects erroneous events and subjects and are often the consequence of the model synthesizing content using terms not present in the document [141]. *Extrinsic Hallucinations* occur when the generated text contains information unverifiable utilizing the source content [141, 95]. These are prevalent in abstractive summarization, since the model can add information from its own internal knowledge [141].

Distinction by Conflict Type. Apart from the standard distinction between Intrinsic and Extrinsic Hallucinations, recent research has categorized it based on

other parameters. Specifically related to Large Language Models, hallucinations can be categorized by the source of their conflict. *Input-Conflicting Hallucinations* happen when the generated content deviates from the user instructions [248]. *Context-Conflicting Hallucinations* happen when the model generates content that contradicts information it previously generated [248]. This is often related to a loss of context tracking. *Fact-Conflicting Hallucinations* are found when the generated text contradicts established world knowledge [248]. This is a known issue with LLM reliability that tend to fabricate information when they lack relevant knowledge or have internalized different knowledge.

Other Hallucination Categorizations. Others have proposed a fine-grained taxonomy of hallucinations, where errors might be related to: wrong entities (incorrect names and locations), relation errors (incorrect semantic relationships), contradictory sentences (statements that contradict the source material), invented information (completely fabricated knowledge), subjective statements (statements that contradict world knowledge), and unverifiable statements (information that cannot be verified with in-context knowledge) [147].

2.4.2 Attribution

Attribution is the process of linking LLM-generated content to verifiable evidence. Attribution has several advantages. First, it incentivizes *Trust and Transparency*, allowing the users to see exactly where the information is sourced. The demand for attributable answers rises from users experiences with hallucinated responses and the inability to navigate large knowledge bases to manually identify the sources. Second, attribution mechanisms allow for *Verifiability*, supporting fact-checking and human feedback. Finally, RAG systems require attribution for *Hallucination Reduction*. Some studies have reported that models with grounding capabilities exhibit significant factuality improvements [234, 241]. Others, prefer to utilize evidence to correct answers after the generation process [67].

Attribution Taxonomy

In-line Citations vs Answer-level Attribution. Several approaches emphasize granular citations. *In-line* evidence models generate answers and citations in a single forward pass. As such, the generated output is a single string containing both free-form text and verbatim quotes or references linked to specific sources

[150]. The attributed sources themselves might be linked to the generated content by means of the entire document [67, 69] or fine-grained passages [113, 142, 234]. The latter approach is preferred in long-context scenarios, where citing the entire source document forces the user to search for evidence manually [8, 241]. Other approaches operate with a coarse granularity on *answer-level attributions*. Some systems in Attributed-QA output a longer output attached to a single pointer linking it to a paragraph-document from a fixed corpus [27]. Additionally, distinctively from factual grounding, some fine-tuned models output a watermark to identify data provenance, not from a retrieval corpus but from the model training data [130].

LLMs vs Encoders. Methods can be classified by their underlying architecture. Generative systems leverage *LLMs* to generate citations. While most approaches rely on a fixed corpus, others rely on web research tools to generate answers with citations [150]. Most importantly, utilizing LLMs allows for both answer generation and post-hoc refinement [67]. On the other hand of the spectrum, *encoder-based* systems rely on rerankers [149] or LLM-derived classifiers [163] to identify attributing sources post-hoc.

Zero-Shot vs Fine-Tuning. Some systems rely on frozen LLMs inherent capabilities to solve attribution. As for other LLM-based tasks, few-shot learning was found to be a viable solution to generate answers with citations without the need for expensive training [27, 67]. While viable for short-context problems, few-shot learning can be expensive, and in-context learning baselines were found to lack complete citation support a significant portion of the times [27]. To address the limitations of prompting, fine-tuning strategies have been proposed to solve the attribution problem. While proprietary models were found to be generally capable on the task, they were repeatedly surpassed by RLHF [142, 150], fine-tuning on synthetic datasets [241], and weak supervision [8].

Context-Attribution vs Source-Attribution The term ”attribution” is used to describe two fundamentally different tasks. *Context-attribution* [46] focuses on verifying generated statements against a provided context and aims to provide evidence that supports the generated claims [69, 142, 150] or influenced the generation process [46]. Finally, *source-attribution* distinguishes itself for the fact that no information is provided to generate the final answer. Instead, a model is

tasked with answer generation and the supporting knowledge is searched ”in the wild” after the generation process has ended [67, 149].

2.5 Industrial PhD Context

The Operational Environment

This research was conducted within the industrial framework and applications of Altilia.ai, an Italian company specialized in document Information Extraction at scale, of which the PhD candidate is a full time employee. Our operational context involves processing heterogeneous document collections. Unlike controlled academic datasets, industrial corpora contain different structures, noise, visual complexity, information density, and formats. The company is currently serving clients in the financial and legal domains and is expanding globally, creating specific requirements for Information Extraction, Question Answering, and Automatic Text Summarization.

The Technological Shift

The company has a history for leveraging Language Models (LMs) and Computer Vision models for tasks like IE, Named Entity Recognition, Object Detection, and Text Classification. However, the industrial scenario has recently shifted toward the utilization of LLMs and MLLMs, driven by the need to address complex reasoning tasks and the growing demand for generative capabilities, such as advanced document understanding.

The Academic-Industrial Gap

The adoption of new technologies highlights the significant and increasing gap between academic research, delving into recent innovations, and industrial applications, relying on well established models and techniques. While users and clients frequently request the deployment of solutions based on the latest prototypes and technologies, the application of these models to real-world scenarios presents distinct challenges. First, the capabilities of LLM-based systems are currently largely unknown in many specific and industrial domains due to the lack large-scale dataset and relevant benchmarks. Second, companies face the need to evaluate and deploy new LLM-based solutions at scale while maintaining minimal costs.

Research Constraints and Methodology

This research was subject to specific industrial constraints, typical of the corporate environment, and not present in academic settings. First, solutions and prototypes must be promptly deployed in production environments, with limited available time for extensive experimental iterations. Second, most of the themes addressed in this manuscripts were not fully mature at the time of development, drastically limiting the availability of literature-ready datasets and methods. Finally, the main objective of this research was the development and study of robust end-to-end pipelines capable of addressing specific industrial needs, rather than solely pursuing theoretical novelty.

Chapter 3

Multi-Modal Summarization Components

The development of a robust pipeline for multi-modal input and multi-modal output document summarization is an open challenge due to the complexity of unstructured and multi-modal data. Moreover, financial users often requires summarization systems to be controllable and aspect-based. Developing such a complex system requires understanding of how current state-of-the-art models handle modalities like text, tables, and charts. The construction of an architecture for multi-modal summarization requires addressing three components: tabular data, visual data, and output control. First, regarding *tabular data*, we must determine how effectively LLMs can interpret tables without specific pre-training. Second, concerning *visual data*, it is necessary to assess to what extent MLLMs can effectively translate chart images into structured data. Finally, *output control* is assessed establishing whether LLMs can satisfy user requirements to focus on specific topics solely through prompt engineering.

This chapter details fundamental preliminary studies to answer these three questions. Our experiments were mostly evaluative but decisive in shaping our final pipeline decisions. Isolating components allows us to establish current models' boundaries and to identify the most efficient methods for multi-modal integration. Consequently, this chapter deliberately excludes exhaustive comparisons with alternative, specialized architectures for table-to-text and chart-to-table transformations. Our scope is strictly restricted to evaluating whether the inherent,

prompt-driven capabilities of foundational models are sufficient for our pipeline, bypassing the need for complex heuristic pipelines or task-specific models. Specifically, the following summarizes the output of our experiments.

Table-to-Text Transformations. We evaluated the hypothesis that modern LLMs are able to interpret linearized tabular data and generate coherent descriptions. This eliminates the need for complex pre-processing steps and deployment of specialized models. Our findings demonstrate the LLMs can indeed reliably interpret and summarize the content of complex tables solely through prompt engineering.

Chart-to-Table Transformations. We investigated MLLM capabilities to reliably transform charts into tables. Our hypothesis was that MLLMs are able to correctly map charts into structured tables. This research step is motivated by two main factors: (i) the excessive cost of multi-modal inputs and (ii) the ability to store charts as textual representations. Specifically, this second part is required to facilitate the inclusion and retrieval of content from Altilia's system. Current MLLMs are unable to map charts into tables with high fidelity, often resulting in hallucinations. Due to the negative results, we rejected the chart-to-table approach for the final pipeline.

Controllable Text Summarization. We analyzed approaches to constraint the length, style, and structure of the generated summaries. This experiments are made to test wether Parameter Efficient Fine-Tuning (PEFT) strategies are required to align the model output with strict user requirements. Through extensive evaluations, we found that LLMs are highly responsive to human instructions and concluded that fine-tuning is generally not required for controllability.

3.1 Table-to-Text with Large Language Models

3.1.1 Literature Review

The Neural Networks utilization paradigm consists of extensive self-supervised pre-training on large-scale dataset followed by fine-tuning on additional data. Despite this, their domain generalization capabilities remains limited, often requiring further fine-tuning to solve downstream tasks on new domains. LLMs posses the potential of generalizing to new domains outside of the current paradigm,

if they were properly taught to solve a downstream task in advance. The first approach that utilized a LLM [21] to generate textual descriptions of tables relied on Quantized Low-Rank Adaptation (QLoRA) to fine-tune two modules out of LLaMa2-7B [208]. Despite the potential of LLMs to solve the task, the approach was still divided into the retriever-generator paradigm typical of RAG. The first module is a table reasoner, identifying the most crucial information withing a table, and the second a table summarizer, that generates natural language based on the text highlighted from the first model. Yang et al. [228] employed a different strategy, using knowledge distillation to develop smaller models emulating the results from proprietary GPT models.

LLMs can be used on new domains without task-specific fine-tuning. Chen et al. [33] utilize GPT-3.5 [28] to generate reasoning paths from tables and concludes that fine-tuned LMs exhibit superior performance in surface realizations, whereas LLMs excel in quality-based human evaluation, though they do not quantitatively assess the performance difference. DATER [235] utilizes LLMs and few-shot learning for table and question decomposition, then utilizes smaller tables and simpler questions to solve Table Question Answering (Table QA). BINDER [39] uses in-context learning to determine relevant entities within the input table before answering queries. ToolWriter [72] utilizes GPT-3.5 as a tool calling model to process tables before answering queries with a QA-tuned BERT-based model. SCITAB [131] employs models to study LLMs' fact-checking capabilities, concluding that they fall short when compared to human performance levels, that CoT [219] does not aid the task, and that LLMs outperform fine-tuned LMs on the proposed dataset. Finally, Zhao et al. [251] investigate data insight generation on LogicNLG [34] and LoTNLG [251].

3.1.2 Experimental Setting

Given a structured table containing relational data, numerical data, or a combination of the two, the goal is to generate a natural language description that coherently conveys the key information present in the table. The generated text should capture the most salient facts while omitting irrelevant details. The task requires the model to understand the table structure, content, and context. Due to the heterogeneous nature of table data, we evaluate models on the following requirements: (i) handling diverse table structures, (ii) identifying and focusing on the most relevant subsets of table data, (iii) performing logical reasoning and numerical operations, and (iv) generating grammatically correct text.

Dataset	Domain	Documents			Dimensions		Tokens
		train	val	test	rows	columns	(table description)
WebNLG [70]	General Purpose	5,573	790	-	14.82	3.00	19.77
NumericNLG [198]	Scientific Tables	1,084	136	135	8.13	5.56	128.42
ToTTo [162]	General Purpose	131,849	7,700	7,700	32.87	5.31	14.84

Table 3.1: Table-to-text datasets

Datasets. We have selected three publicly available table-to-text datasets to test LLMs, as detailed in Table 3.1. WebNLG [70] is constituted by sets of factual statements, in the form of triples. Each table is coupled with a textual description that covers the entire content of the table. NumericNLG [198] contains numerical tables extracted from scientific sources. Each table is accompanied by a list of target row indexes and column headers, which must be referenced in the textual description. The remainder of the table is essential for logical inference. Finally, ToTTo [162] contains large tables, with the description only covering a few highlighted cells. Due to the size of the ToTTo dataset and the costs associated with running inference with LLMs, we utilize the same sample commonly used in literature experiments [197].

Models. We select two language models with significantly different scales, deployment characteristics, and accessibility profiles. *GPT-3.5* [28] represents the larger option, with approximately 175B parameters. This model has demonstrated strong generalization capabilities across a wide range of natural language tasks and is accessible exclusively through OpenAI [1] API. GPT-3.5 is privately owned, so no computational resources are necessary for deployment. For the open-source counterpart, LLaMa2-7B [208] is significantly smaller and can be deployed on consumer-grade GPUs. Despite the smaller size, the model has shown competitive performance on multiple benchmarks and represents a realistic option for local deployment [208]. These models allows us to investigate three dimensions of the table-to-text problem. First, it allows to compare the effect of model scale on performance, especially when structural understanding is required. Second, by comparing proprietary and open-weight models we can understand the trade-offs between accuracy and costs under substantially different infrastructure requirements. Finally, the setup provides insights into the extent to which open-source models can approximate the performance of large commercial LLMs on text

[1] https://openai.com/

Prompt Template

```
User:  You are given a table in one of the following formats:
html, plain text, or json.
Your job is to use the table content to produce a short
paragraph.
The paragraph must have the following properties:
- it must be a few sentences long, if you believe that a
 single sentence is enough you can use a single sentence
- all the information in the table must be included in the
paragraph
- it must not include information that is not available in
the table
- it must not mention that the paragraph comes from a table
TABLE:
{table}

PARAGRAPH:
```

Figure 3.1: Table-to-text prompt template for WebNLG

generation from structured data.

Generation Approaches. We empirically analyze models' performance under different settings and table structures. First, we prompt models with *unstructured tables*, i.e. tables linearized into plain text without separators. Then, models are queried with *structured tables*, using JSON and HTML formats. In the JSON setting, tables are linearized into a list of dictionaries containing cell values, rows indexes, column headers, and information about spanning cells, while the HTML setting consists of tables in HTML representation.

The WebNLG dataset contains stacked triples rather than tables. These are organized with headers labeled as *subject*, *relationship*, and *object*. NumericNLG tables are concatenated with the provided context before feeding them to the model. Additionally, the prompt specifies a list of target entities, along with their locations, that must be included in the generated description. Finally, *cells highlights* from ToTTo are used to augment the table representations. Structured representations are augmented adding a *highlight* attribute to each cell and unstructured tables are augmented using special `"HIGHLIGHT"` tokens. All our prompt templates are reported in Table 3.1, Table 3.2, and Table 3.3.

Prompt Template

```
User:  You are given a table in one of the following formats:
html, plain text, or json.
The table is accompanied by a context and a target list.
The targets are to be found in the row indexes.
The context can be in the form of table caption, title, or
some other text.
Your job is to use the table and the context to produce a
paragraph.
The paragraph must have the following characteristics:
- it must be structured in a way that allows the reader to
understand it without seeing the table
- it can't contain lists
- it can only mention the entities in the target list

TABLE:
{plain_table}

TARGET LIST:
{target_list}

PARAGRAPH:
```

Figure 3.2: Table-to-text prompt template for NumericNLG

Fine-Tuning Strategy. As far as tuning is concerned, we sample 1,000 examples from each training dataset and fine-tune using Low-Rank Adaptation (LoRA) [88]. We run fine-tuning for a single epoch using a learning rate of 1e-5, standard LoRA settings, and a batch size of 2. Fine-tuning and inference are run on a single A100 GPU with 40GB of RAM.

3.1.3 Evaluation Metrics

We employ n-gram overlap metrics for syntactic assessment and entailment-based metrics for semantic evaluation. Regarding n-gram metrics, we present results for BLEU [161], ROUGE [118], METEOR [17], and TER [193]. Additionally, we utilize PARENT [55] to compare the generated text not only to the reference table descriptions but also to the input tables. Among the entailment-based metrics, we include BERTScore [246] and BLEURT [186].

Prompt Template

```
User:  You are given a table in one of the following formats:
html, plain text, or json.
The table is accompanied by a context.
The context is in the form of page title, table title,
section title, and a few sentences taken from the same
section as the table.
The table has some highlighted cells.
Your job is to use the table and the context to produce a
paragraph.
The paragraph must have the following characteristics:
- it must be structured in a way that allows the reader to
understand it without seeing the table
- it can't contain lists
- it must be only a single sentence long
- {highlight_instructions}
- it must not explicitly mention that some cells are
highlighted

TABLE:
{plain_table}

CONTEXT:
page title:  {page_title}
section title:  {section_title}
section text:  {section_text}

PARAGRAPH:
```

Figure 3.3: Table-to-text prompt template for ToTTo

BLEU. The Bilingual Evaluation Understudy (BLEU) metric measures the differences between machine-generated and human-written text. BLEU computes the number of n-grams (ordered sequence of N words) matching between machine- and human-written text. The metric is based on *Modified N-gram Precision*:

$$p_n = \frac{\sum_{C \in Candidates} \sum_{\textit{n-gram} \in C} Count_{clip}(\textit{n-gram})}{\sum_{C' \in Candidates} \sum_{\textit{n-gram'} \in C'} Count(\textit{n-gram'})}$$

where $Count_{clip}(\textit{n-gram})$ is the number of times the n-gram is matched, clipped by the maximum count in the reference sentence.

To avoid bias toward shorter text, BLEU is computed by multiplying the geometric average of precisions by a brevity penalty:

$$BP = \begin{cases} 1 & \text{if } c > r \\ \exp(1 - \frac{r}{c}) & \text{if } c \leq r \end{cases}; BLEU = BP \times exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

where c and r are the candidate and reference lengths respectively.

ROUGE. The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is a recall oriented metric designed to determine the quality of a machine-generated text with respect to one or more human-written versions. This measures the number of n-grams matching between a candidate and reference text. Among the numerous versions of ROUGE, two are used the most: ROUGE-N and ROUGE-L. The N-gram co-occurrence statistic, ROUGE-N measures the n-gram recall between the candidate and reference text. Specifically, ROUGE-N measures the ratio of matching n-grams over the total from the reference texts:

$$ROUGE - N = \frac{\sum_{S \in References} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in References} \sum_{gram_n \in S} Count(gram_n)}$$

ROUGE-L is based on the Longest Common Subsequence (LCS) and is an F1-based metric. Given the LCS between a candidate X, of length m, and a reference Y, of length n, the measure is based on the following equations:

$$R_{lcs} = \frac{LCS(X,Y)}{m}; P_{lcs} = \frac{LCS(X,Y)}{n}; F_{lcs} = \frac{1 + \beta^2 P_{lcs} R_{lcs}}{\beta^2 P_{lcs} + R_{lcs}}$$

METEOR. The Metric for Evaluation of Translation with Explicit Ordering (METEOR) addresses some limitations of BLEU, solely relying on exact word matches. Specifically, METEOR integrates a *Porter Stem module* and a *WordNet Synonymy module*. The first maps unigrams with each other if they share the same stems after being processed, and the second maps unigrams if they are synonyms. Once words are aligned, precision is computed as the ratio of matched unigrams over the total from the candidate text and recall as the ratio of matched unigrams over the total from the reference. The metric accounts for fragmentation using a penalty.

$$\rho = 0.5 \times \left(\frac{\text{\# chunks}}{\text{\# matched unigrams}} \right)^3; F_{mean} = \frac{10PR}{R + 9P}; s = F_{mean} \times (1 - \rho)$$

TER. The Translation Error Rate (TER) measures the quality of machine generated text by measuring the minimal amount of editing required to transform the machine-generated text into the human-written counterpart. Editing is define by operations of insertion, deletion, substitution, and shift. Once the number of edits is counted, TER is computed as

$$TER = \frac{\text{\# edits}}{\text{average number of words in the reference}}$$

PARENT. The Precision And Recall of Entailed N-grams from the Table (PARENT) improves over n-gram overlap metrics by accounting for text containing information not found in the source tables. First, n-grams from the generated text are matched with n-grams from the reference texts. An n-gram is considered correct if it is contained in the reference or it has a high probability of being entailed in the table. N-gram entailment is computed as the percentage of tokens matching in the flattened table. Then, n-gram precision (P_N) and recall are computed as METEOR and the PARENT score uses a standard F1 formulation:

$$E_p = (\prod_{i=1}^{4} P_i)^{\frac{1}{4}}; E_r = E_r(R)^{1-\lambda} E_r(T)^{\lambda}; PARENT = \frac{2E_p E_r}{E_p + E_r}$$

BERTScore. This is an automatic evaluation metric designed for text generation. The metric measures the semantic equivalence between a candidate and a reference sentence. BERTScore uses pre-trained contextual embedding models to represent sequences of tokens as vector representations. First, all tokens in a candidate and reference sequence are embedded. Then, cosine similarity and greedy matching are used to match the most similar tokens from the two sequences. Precision and recall are calculated by matching the tokens from the *reference* and *candidate* sequences respectively. F1 is computed as the harmonic mean of precision and recall.

Model	Template	BLEU	R-1	R-2	R-L	METEOR	TER	BERTScore	BLEURT
Zero-Shot Prompting									
GPT-3.5	plain	34.80	69.60	44.72	54.66	40.01	74.67	94.17	70.58
	HTML	**39.58**	**72.61**	**47.88**	**57.52**	**41.29**	**64.45**	**94.47**	**72.43**
	JSON	37.55	70.48	45.78	54.74	40.83	70.64	93.84	70.27
LLaMa2	plain	22.06	56.81	33.71	42.45	34.65	126.92	91.96	60.5
	HTML	23.88	58.14	35.05	43.69	35.17	118.7	92.18	61.56
	JSON	21.68	53.74	31.44	40.13	33.73	132.84	91.05	54.73
Fine-Tuned Models									
LLaMa2		51.42	**78.14**	**53.91**	**62.92**	40.72	**43.46**	**95.40**	**75.55**
ExT5$_{large}$ [12]		35.03	-	48.17	-	36.50	-	-	-
PaLM$_{540B}$ [42]		49.30	-	-	-	-	-	-	-
T5$_{CP}$ [45]		**55.41**	-	-	-	**42.00**	39.10	-	63.00

Table 3.2: Metrics on WebNLG.

BLEURT. This is a learned metric designed to evaluate similarity between sentences. Bilingual Evaluation Understudy with Representations from Transformers (BLEURT) uses BERT to capture semantic-level similarities. Given a candidate and reference, BERT is trained to output the embedding for a special `[CLS]` token, which is then mapped to a $[0, 1]$ score using a linear layer.

3.1.4 Results

This section details the analysis of the experimental results on the three selected datasets. We assess performance of zero-shot and fine-tuned approaches.

WebNLG analysis. In the zero-shot settings, GPT-3.5 consistently demonstrates superior performance over LLaMa2. The performance gap is evident across both semantic and syntactic evaluations and prompting strategies. About input representations, both models exhibit consistently better results when prompted with HTML tables. LLaMa2 shows significant improvements when fine-tuned compared to the zero-shot counterparts. The comparisons with literature models shows that the fine-tuned model exhibits competitive performance regardless of the minimal training sample. The metrics for WebNLG are reported in Table 3.2.

NumericNLG analysis. The results for NumericNLG, reported in Table 3.3 present a different pattern regarding input formats. Unlike with WebNLG, GPT-3.5 displayed no distinct preference for table representations, with plain, HTML, and JSON performing comparably. LLaMa2 generally achieved higher scores using

Model	Template	BLEU	R-1	R-2	R-L	METEOR	TER	PARENT	BERTScore	BLEURT
Zero-Shot Prompting										
GPT-3.5	plain	5.44	33.47	10.48	21.57	**16.49**	133.53	**16.52**	85.74	32.53
	HTML	5.47	33.04	10.23	21.40	15.94	134.52	**16.52**	85.48	**32.69**
	JSON	5.25	32.93	10.34	21.38	16.03	**135.77**	16.51	85.32	32.08
LLaMa2	plain	5.23	**35.26**	9.74	21.59	14.82	107.32	12.49	86.32	29.22
	HTML	**5.52**	35.23	**10.78**	**22.49**	15.07	107.16	13.25	**86.42**	29.55
	JSON	4.54	33.73	9.91	21.41	14.59	128.63	12.67	85.64	28.50
Fine-Tuned Models										
LLaMa2		**5.71**	**36.47**	**14.18**	27.08	12.48	**88.11**	14.44	87.55	**33.62**
$T5_{template}$ [198]		5.02	–	–	**30.25**	**20.11**	–	**15.09**	**87.68**	–
TASD [32]		–	–	–	20.40	11.87	–	–	–	–

Table 3.3: Metrics on NumericNLG.

the HTML format. However, on this dataset, the performance improvement is not strictly related to formatting and seems to be negligible. As for WebNLG, fine-tuning LLaMa2-7B is sufficient to achieve superior scores, even though PARENT remains greater with GPT-3.5. Finally, the fine-tuned LLaMa2 achieved results comparable to the literature counterparts.

ToTTo analysis Table 3.4 shows the results on ToTTo. This dataset poses unique challenges due to table size and the requirement to focus on the highlighted cells. In zero-shot experiments with GPT-3.5, the plain table format consistently outperformed structured formatting across all metrics. LLaMa2, on the other hand, showed mixed results. While HTML achieved the highest BLEU score, plain text was superior for other metrics. Notably, BLEURT scores remained exceptionally low for zero-shot attempts. Fine-tuning LLaMa2 provided significant performance gains over the zero-shot baseline.

Model	Template	BLEU	R-1	R-2	R-L	METEOR	TER	PARENT	BERTScore	BLEURT
Zero-Shot Prompting										
GPT-3.5	plain	**8.30**	**37.04**	**18.91**	**27.92**	**26.75**	290.10	**39.05**	**88.11**	**-0.484**
	HTML	**8.30**	35.52	17.89	26.77	26.41	**284.80**	37.44	87.09	-0.521
	JSON	6.90	31.89	15.65	24.18	23.71	324.07	34.38	83.08	-0.614
LLaMa2	plain	3.60	27.12	11.24	20.68	17.53	358.13	24.11	76.93	-0.769
	HTML	4.00	23.79	10.29	18.04	17.36	369.59	22.07	75.11	-0.864
	JSON	3.40	16.98	7.12	12.79	13.91	388.46	16.25	59.00	-1.131
Fine-Tuned Models										
LLaMa2		**28.00**	**54.81**	**34.04**	**46.88**	**26.03**	**72.06**	**36.93**	**85.26**	**-0.218**

Table 3.4: Metrics on ToTTo.

Model	Template	BLEU	R-1	R-2	R-L	METEOR	TER	PARENT	BERTScore	BLEURT
Zero-Shot Prompting										
GPT-3.5	plain	16.31	54.45	30.00	41.07	33.48	143.70	47.38	85.58	-0.269
	HTML	17.57	55.10	29.97	41.38	34.44	129.41	47.01	86.53	-0.180
	JSON	17.13	53.12	29.29	40.29	34.24	136.65	**47.84**	86.10	-0.244
	ToTTo	**18.70**	**56.57**	**31.65**	**43.68**	**34.70**	**118.36**	47.52	**87.09**	**-0.081**
LLaMa2	ToTTo	15.30	51.80	27.18	38.86	30.98	126.52	38.48	85.53	-0.188
Fine-Tuned Models										
LLaMa2	ToTTo	45.19	**67.91**	**44.38**	**57.39**	**35.39**	**59.51**	56.43	**90.58**	0.176
T5-base-CONT [9]		49.10	–	–	–	–	–	**58.90**	–	0.238
Plan-then-Generate [197]		**49.20**	–	–	–	–	–	58.70	–	**0.249**

Table 3.5: Metrics on ToTTo reduced tables.

Table Reduction Impact Recognizing that model performance is likely to degrade with excessive input length, we designed specific experiments on ”reduced” ToTTo tables. The experiment involves the removal of non-highlighted cells and the concatenation with their respective row and column headers. The results are reported in Table 3.5. First, all metrics have improved significantly across all zero-shot experiments. Secondly, the results of fine-tuning on pre-processed tables produces metrics comparable to the SOTA literature counterparts despite using an order of magnitude less training data. The effectiveness of this representation is likely due to the removal of unnecessary elements, preventing the model from incorporating irrelevant content despite being instructed not to do so.

3.1.5 Human Evaluation

WebNLG. During examination of GPT-3.5 outputs on WebNLG, critical errors are not readily apparent but some minor inconsistencies are noticeable. For instance, when the text is generated from unstructured tables, there might be inconsistencies correlated with shifting subjects from active to passive voice. Despite the occasional errors, the generated outputs are of good quality and errors altering the sentence's meaning are infrequent. Similar errors are observed with the finetuned version of LLaMa2. Despite the quality of the generated sentences, it is imperative to acknowledge the model's inability to fully incorporate the table details from the provided content. Finally, zero-shot generation with LLaMa2 exhibit significant hallucinations overall, as the model consistently sources information from memory, ignoring the table content.

NumericNLG. For NumericNLG, the output generated by GPT-3.5 is typically satisfactory, covering the necessary information. Despite instructions to solely reference entities listed in the provided target list, the model often incorrectly unnecessary entities. Additionally, the model sometimes introduces a table structure description before delving into the content, which is undesirable. The logical inference remains accurate overall. Zero-shot generations from LLaMa2 exhibit many inaccuracies, such as table structure misinterpretations, referencing wrong entities, and omitting important values. Although fine-tuning enhances performance, we observed that the resulting outputs lack informativeness. This probably stems from shortcomings of the training dataset. Consequently, the emphasis seems to be on mimicking the training dataset style rather than improving the quality of the generated content.

ToTTo. GPT outputs often incorporate surplus details from the table. Occasionally, these outputs reference non-highlighted cells and the logical deductions may be inaccurate. Frequently, LLaMa2 doesn't produce outputs due to the excessive length of HTML and JSON tables. Generations from plain tables contain excessive information, attempting to encompass a large portion of the table. Fine-tuning yields nearly flawless outputs with occasional omissions. Noteworthy is the pre-processing technique, which lead to favorable generations. The output still contains excessive information but the generated text is highly satisfactory.

Model	Throughput			Cost		
	WebNLG	NumericNLG	ToTTo	WebNLG	NumericNLG	ToTTo
plain						
GPT-3.5	$0.30_{it/s}$	$0.24_{it/s}$	$0.72_{it/s}$	0.30$	0.08$	1.66$
LLaMa2-7B	$0.20_{it/s}$	$0.16_{it/s}$	$0.26_{it/s}$	4.27$	0.93$	7.60$
LLaMa2-7B$_{ft}$	$0.12_{it/s}$	$0.03_{it/s}$	$0.13_{it/s}$	7.12$	5.00$	15.21$
JSON						
GPT-3.5	$0.25_{it/s}$	$0.22_{it/s}$	$0.59_{it/s}$	1.65$	0.31$	14.92$
LLaMa2-7B	$0.25_{it/s}$	$0.19_{it/s}$	$0.29_{it/s}$	4.27$	0.79$	6.82$
LLaMa2-7B$_{ft}$	–	–	–	–	–	–
HTML						
GPT-3.5	$0.29_{it/s}$	$0.24_{it/s}$	$0.69_{it/s}$	0.92$	0.19$	7.30$
LLaMa2-7B	$0.21_{it/s}$	$0.17_{it/s}$	$0.24_{it/s}$	4.07$	0.88$	8.23$
LLaMa2-7B$_{ft}$	–	–	–	–	–	–

Table 3.6: Throughput and costs of generating table descriptions under different settings. Inference is run sequentially.

3.1.6 Costs and Inference Times

We report throughput and total experiments costs in Table 3.6. The table reveals significant variation in model efficiency and operational costs depending on the input format and dataset. Across all settings, GPT-3.5 consistently achieves the highest throughput and the lowest cost, with the exception of the inferences run on the ToTTo dataset under the JSON setting. In contrast, LLaMa2 exhibits stable throughput across all formats while remaining generally slower and more expensive than GPT. The higher cost of the open-source model is due to the significantly slower inference time and the high cost associated GPU (A100 GPU with 40GB of vRAM) rental. Finally, the fine-tuned version of LLaMa2 required additional inference times, further increasing the cost from the open-source model. While this could be reduced by merging the adapter weights with the original model, we decided not to merge them and utilize three different adapters to simulate production settings. This significantly increases the computational overhead resulting in significantly higher costs.

3.1.7 Conclusions

In this work, we conducted a comprehensive investigation of LLMs for table-to-text generation across multiple datasets. Our results highlight the zero-shot capabilities of large-scale LLMs. However, we demonstrated that fine-tuning smaller models on a small dataset can help to significantly bridge the performance gap between larger and smaller models. Our fine-tuned version of LLaMa2-7B

achieved results comparable with the latest SOTA techniques on certain datasets and metrics, demonstrating that LLMs can leverage their vast knowledge from pre-training to generalize to new table-to-text distribution in a sample-efficient manner.

Despite the promising findings, our analysis revealed several important limitations that need to be addressed. First, zero-shot generation from smaller models often suffer from hallucinations and factual inconsistencies when describing tabular data, limiting their applicability in real-world scenarios. Then, finetuned models tend to produce outputs lacking informative details, likely due to limitations and biases encountered in the training data. Third, handling complex and large tables results in degrading quality as the table complexity increases. Finally, open-source model fine-tuning remains computationally expensive and methods like LoRA only begin to address these challenges for modest model sizes. The price associated with slower inference and GPU renting prices, makes deploying open-source models more expensive and less desirable overall.

3.2 A Comparative Analysis of State of the Art Chart-to-Table Models

Data visualization has recently proliferated in the scientific literature and business documents. This has been creating vast repositories of information locked behind graphical representations. Charts, designed to simplify the interpretation of complex information for the human eyes, act as barriers for machine interpretability. The task of recovering tabular data from charts is formally known as Chart-to-Table (C2T) mapping [120]. Structured tables are optimized for delivering granular data, exact figures essential for scientific and financial analysis. Consequently, C2T models serve as the crucial link between visual perception and machine readable knowledge. Traditional LLMs are blind to multi-modal data, underscoring the need for this comparative evaluation. Today, about 80% of enterprise data contains visual elements carrying crucial business intelligence and over 60% of critical business decisions rely on information embedded in visual components[2]. Robust C2T capabilities are necessary for digital inclusion and adhering to web content accessibility guidelines requires providing structured tables as alternatives to charts.

Retrieving a chart's underlying table can be challenging. Unlike generic images, charts rely on intricate visual cues, implicit numerical information, and complex spatial relationship to convey meaning. While recent MLLMs were reported to be quite accurate on Chart-to-Table (C2T) after fine-tuning, the biggest challenge comes from the heterogeneity of real-world chart images, often employing uncommon formats and multiple overlapping chart types. Moreover, models must deal with modality hallucinations, where the lack of alignment between visual and textual elements leads to erroneous data recovery. While recent proprietary models have demonstrated remarkable capabilities on standard benchmarks like ChartQA [138], these are becoming saturated, creating an overly optimistic perception of progress. There is still a significant discrepancy between benchmark performance and real world applicability. Models that excel at simple reproduction often fail with new and complex charts. Understanding which architecture generalize to diverse and complex chart types is essential for developing reliable automated data recovery pipelines. This underscores the need for rigorous evaluation.

In this chapter, we contribute with a rigorous comparative analysis of SOTA

[2]https://www.capellasolutions.com/blog/charts-tables-and-dollars-why-visual-intelligence-is-your-next-strategic-investment

Chart-to-Table models on standardized benchmarks and real-world usage. We provide a comprehensive evaluation across a diverse range of chart types to stress model generalization capabilities. By systematically analyzing the performance gap, we aim to identify the limitations of current architectures and establish the baseline for future advancements in the C2T task.

3.2.1 Experimental Setup

Datasets

To systematically to evaluate models in the reconstruction of a variety of chart types we select the ChartX [224] benchmark, containing 1,152 test samples, each composed of a chart image, the underlying table, python code to render the chart, and text descriptions. ChartX contains 18 different chart types, 22 chart topics, and 7 chart related tasks. Most of the chart tasks are related to QA and are unimportant for our testing. Instead, we rely on the chart reconstruction task. Charts are divided in three macro categories by level of difficulty. *General charts* (bar, line, pie, and variations) are the most simple and are commonly found in scientific and business related documents. *Fine-grained charts* (ring, radar, box, 3D-bar, histogram, treemap, rose, bubble, multi-axis, and area charts) contain dense information in a small area. The most difficult tier, *Specific charts* (heatmap, funnel, and candlestick), contain the most difficult chart to parse.

Models

We select six SOTA models from the scientific literature, with varying architectures and sizes, and one proprietary model for comparisons. Among commercial models, GPT-4o [91] is known for exceptional performance in both reasoning and multimodal understanding, serving as a high performing baseline to compare against smaller, open-source systems. While the model showed promising results in the interpretation of scientific figures and charts, benchmarks in current papers often refer GPT-4v [1] for comparison, which is more expensive and offers slower inference. The smaller and cheaper solution, GPT-4o-mini is also selected to evaluate the cheapest proprietary solution.

Among open-source counterparts, we select models with varying sizes to evaluate the correlation between model size and output quality. Our largest choice is InternVL2-8B [5], that utilizes InternViT [37] vision encoder and uses dynamic resolution strategies to handle different data types and represents a robust

open-source alternative against large proprietary models. While this model was tested for chart understanding, it was never systematically evaluated on the C2T task. With a similar size, ChartVLM [224] is an interpretable Vision-Language Model featuring a cascade decoder architecture and an instruction adapter. This model was selected due to the paper prioritization of structural extraction as a fundamental task required for interpretability. From the mPLUG series, we utilize the 8B models mPLUG-DowOwl-1.5-omni and mPLUG-DowOwl-1.5-chat [87]. Both models utilize and *H-Reducer*, that aggregates horizontal visual features using convolutions to preserve spatial layout information while reducing the total length of the sequence, while being directly trained on the C2T task. On the smaller end of the spectrum, ChartGemma [139] is a 2B parameters Large Visual Language Model (LVLM) based on the PaliGemma [35] architecture. This model relies on data tables for training and it is instruction-tuned to directly generate markdown tables from chart images. Finally, TinyChart [244] is a 3B parameters model designed to efficiently encode high resolution images by merging similar visual tokens and trained to generate Program-of-Thought outputs, generating python code for numerical reasoning instead of performing the reasoning itself. The model is reported to outperform several $>$10B parameters models [244], even on the C2T task. However, the literature lacks systematic chart-type-based evaluation and comparison with the most recent proprietary solutions.

Each of the selected models is equipped with a different template for table parsing. For this reason, we consistently use the same template for inference and design custom post-processing strategies to parse the generated outputs into structured tables.

Evaluation Metrics

Among the metrics commonly used for Chart-to-Text evaluations, we refer to two metrics designed to measure the quality of the generated table: RMS [120] and SCRM [223].

RMS. The Relative Mapping Similarity (RMS) is a metric developed to evaluate the quality of the generated table with respect to the ground truth, treating the table as a list of unordered records. Each entry is represented as a tuple $p_i = (p_i^r, p_i^c, p_i^v)$ consisting of a row header, a column header, and the cell value respectively. The metric considers both the predicted row and column headers' textual accuracy and the table values' numerical precision. For each pair of gt and predicted table entry, RMS computes the textual distance between the headers and the numerical

distance between the predicted values. The textual distance is computed as the Normalized Levenshtein Distance between the concatenated row and column headers. This is set to 1 if it overcomes a threshold τ. As far as table entries are concerned, a simple relative distance is computed and the same thresholding logic is applied. Entry similarity is computed as:

$$D_{\tau,\theta}(p,t) = (1 - \text{TextDistance}) \times (1 - \text{NumericDistance})$$

Entries from the generated and gt tables are aligned by minimal cost matching. RMS Precision (RMS_p) and RMS Recall (RMS_r) are computed as the sum of the similarities from the matched entries over the number of predicted entries (N) and over the number of target entries (M) respectively. The final metric (RMS_{F1}) is the harmonic mean of precision and recall.

SCRM. Structuring Chart-oriented Representation Metric (SCRM) transforms each table entry into triplet format and then performs matching to determine accuracy ensuring that the evaluation focuses on factual data relationships rather than the specific formatting order. Text is compared through the edit distance and numbers using a relative error threshold. SCRM employs three levels of tolerance to account for the complexity of different chart types. Tolerances are defined based on two mathematical thresholds: J_{thr} is the edit distance threshold between the predicted and ground truth strings, and e_{thr} is the relative threshold distance between the predicted and target numerical values.

3.2.2 Quantitative Results

We present the comprehensive results of our experiments in Table 3.7 and Table 3.8 for the RMS and SCRM metrics respectively.

The numbers reported in Table 3.7 show a clear hierarchy of models. GPT-4o achieved the highest average score or 34.03 while we observe a significant performance degradation in the distilled counterpart. The smaller GPT-4o achieves almost systematic lower scores, averaging at $\sim$8 less points. However, the RMS results show highly competitive open-source models that are not systematically outperformed by proprietary counterparts. Ranking first, ChartVLM-large (33.21 points) and ChartVLM-base (30.26 points) are both very close to the much larger proprietary counterpart. Moreover, these model even surpassed GPT-4o for some chart types. First among all, the candlestick chart, in which only ChartVLM was able to achieve appropriate results. While the results obtained by some of this

models are impressive, their quality remains bounded to the variety in the training dataset.

Despite the successes, certain chart modalities are intractable for current models. Across all evaluated models, performance collapses on area charts, box plots, bubble charts, and radar charts. While this suggests a fundamental limitation in the models' ability to interpret complex spatial data, it must be acknowledge that the purpose of these chart types is not to provide the reader with fine-grained numerical information, but rather to allow the human to make quick multi-faceted comparisons without the need to read and compare numbers.

The SCRM metric in Table 3.8 provides a more nuanced view of the models robustness by introducing relaxation levels. Specifically, the metric reveals both that smaller models can achieve respectable performance under lenient constraints and that the gap with larger models could be bigger than previously expected. While the trend remains substantially the same under strict and slight tolerance settings, we observe notable shifts in performance dynamics under the high setting. First, GPT-4o-mini outperforms the larger GPT-4o on tree-maps, scoring 49.0 compared to GPT-4o 40.8, with all other models lagging significantly behind. Second, the SCRM metric highlights the superior spatial reasoning of the GPT family. GPT-4o and GPT-4o mini are the only models to exceed a score of 30 on 3D-Bar charts whereas the nearest competitor, ChartVLM-base, only reaches 14. Finally, the constraints relaxation it highlights that some models are unable to report values without high relative errors. Some models achieve significantly higher scores for spatially complex charts like bubble, multi-axes, and radar charts, highlighting their ability in capturing the semantic trend of the data.

3.2.3 Qualitative Results

To complement the quantitative metrics presented in Table 3.7 and Table 3.8, we conducted qualitative inspections of the generated tables. This allows further understanding of the differences between the models. For each chart type, we report examples in Figure 3.4, Figure 3.5, and Figure 3.6.

While larger architectures (GPT-4o, ChartVLM-Large, ChartVLM-Base, TinyChart, ChartGEMMA) demonstrate robustness in table generation, several systematic error patterns are evident. The most significantly present are *data translation errors*, characterized by the misinterpretation of visual signals into numerical values. This limitation can be observed even in the low-complexity examples of Figure 3.4 and the histogram of Figure 3.5, specifically when the model must interpolate values without explicit data labels. Furthermore, scale

Model	bar	bar$_{num}$	line	line$_{num}$	pie	rings	box	hist	tree-map	rose	area	3D-Bar	bubble	multi-axes	radar	heat	funnel	candlestick	avg
Proprietary Models																			
GPT-4o	22.62	**64.30**	41.43	**60.36**	**77.37**	**49.55**	3.30	53.68	29.27	**34.64**	0.50	**5.60**	**24.98**	**6.73**	1.83	**57.10**	77.95	1.24	**34.03**
GPT-4o-mini	9.89	47.03	18.88	44.38	72.34	37.76	0.88	43.90	26.28	19.61	0.00	4.95	17.74	4.57	0.00	47.91	77.26	2.16	26.42
Open-source Models																			
ChartVLM-large	36.29	53.84	**49.53**	50.43	69.43	28.78	**8.60**	80.58	39.29	15.76	**2.27**	0.37	0.35	4.61	**3.72**	19.18	**88.26**	**46.52**	**33.21**
ChartVLM-base	30.70	40.89	48.58	53.24	61.01	25.39	4.95	77.32	**41.26**	16.00	1.58	1.79	0.98	0.95	1.30	25.15	80.56	33.04	30.26
TinyChart	**38.37**	60.47	38.42	39.46	67.82	27.48	0.00	**81.55**	9.63	3.98	0.59	0.00	0.00	0.47	1.09	10.32	41.22	0.00	23.38
InternVL-v2-8B	34.88	55.45	55.44	52.43	70.34	40.25	0.00	74.59	26.36	16.46	1.94	0.59	3.15	5.55	0.00	42.27	75.78	1.25	30.93
InternVL-v2-2B	26.07	52.31	51.44	46.51	51.52	24.13	0.00	61.00	12.85	5.35	1.32	0.00	2.49	3.44	0.34	33.43	51.82	0.69	23.59
ChartGEMMA	21.33	43.93	27.99	27.40	65.93	34.89	0.00	68.04	30.52	8.43	0.71	0.37	1.81	0.50	0.32	19.23	17.59	0.57	20.53
MiniCPM	28.36	56.52	28.15	47.44	68.48	32.98	0.00	68.11	26.71	17.01	0.89	3.28	7.98	2.52	0.00	50.72	55.17	0.88	27.51
DocOWL-1.5-Omni	26.80	20.78	19.59	14.34	29.99	11.39	0.00	72.23	10.95	4.36	0.28	0.00	0.00	0.00	0.00	8.23	45.07	0.57	14.70
DocOWL-1.5-chat	11.16	11.40	5.55	3.16	28.63	11.99	0.00	12.50	13.05	2.10	0.37	0.00	0.00	0.00	0.00	1.37	12.29	0.00	6.31

Table 3.7: Comparison of various models on the C2T task using the RMS metric.

Model	Setting	bar	bar_num	line	line_num	pie	rings	box	hist	tree	rose	area	3D-Bar	bubble	multi	radar	heat	funnel	candle	avg
Proprietary Models																				
GPT-4o	strict	0.0	46.0	9.0	29.2	85.0	34.0	0.0	3.2	26.2	8.6	0.0	0.2	8.0	2.8	1.4	64.2	76.8	0.0	21.9
	slight	5.8	55.0	30.6	53.6	85.0	34.0	0.4	8.6	26.2	24.4	0.0	14.8	28.6	18.6	15.0	69.0	78.0	0.0	30.4
	high	18.8	60.0	50.2	58.4	85.8	41.2	10.0	27.4	40.8	63.4	6.4	38.0	47.0	39.6	41.0	70.8	91.0	2.8	44.0
GPT-4o-mini	strict	0.0	21.0	4.0	14.2	74.4	26.0	0.0	3.0	24.2	2.4	0.0	1.0	0.2	0.2	0.0	52.8	74.6	0.0	16.6
	slight	1.4	34.2	19.6	32.6	74.4	26.0	0.0	4.0	24.2	7.8	2.0	17.0	9.4	9.0	5.0	58.8	75.8	0.0	22.3
	high	5.2	40.0	40.0	42.4	77.6	30.0	0.0	26.4	49.0	36.6	4.2	31.8	27.0	33.0	15.2	60.8	84.2	3.4	33.7
Proprietary Models																				
ChartVLM-large	strict	10.4	34.4	30.2	27.6	64.8	6.8	0.4	42.2	9.6	0.2	0.0	0.0	0.0	0.0	0.0	17.6	85.0	0.2	18.3
	slight	15.6	37.0	47.0	40.6	64.8	6.8	3.2	77.2	9.6	6.8	2.8	2.4	2.0	2.0	14.4	19.4	85.0	48.4	26.9
	high	18.8	37.8	54.6	49.0	66.4	6.8	22.4	85.2	9.6	29.4	8.6	10.0	4.4	4.8	19.2	21.2	85.2	65.4	33.3
ChartVLM-base	strict	2.0	19.4	23.0	33.0	47.2	7.2	0.8	40.2	6.4	2.6	0.0	0.0	0.0	0.0	0.0	23.8	74.0	0.0	15.5
	slight	9.2	23.4	41.2	44.2	48.0	7.2	3.0	72.6	7.8	6.4	0.0	8.2	1.2	0.0	11.6	30.4	77.0	34.0	23.6
	high	14.6	28.4	50.0	50.8	51.2	8.0	15.4	79.8	7.8	23.0	5.8	14.0	2.2	0.8	13.4	36.4	78.0	47.0	29.3
TinyChart	strict	2.2	26.8	5.6	9.0	57.8	15.6	0.0	12.8	4.8	0.0	0.0	0.0	0.0	0.0	0.0	1.8	48.4	0.0	10.3
	slight	17.4	42.0	19.6	20.6	59.4	16.4	0.0	55.0	4.8	1.2	0.0	2.0	0.0	0.0	5.6	5.2	53.0	0.0	16.8
	high	33.4	56.8	30.4	37.6	65.6	16.4	0.0	73.0	5.8	6.6	2.6	9.4	0.0	0.2	10.0	8.8	60.2	0.0	23.2
InternVL-v2-8B	strict	6.4	39.8	15.0	28.4	69.8	30.4	0.0	37.6	21.4	3.0	0.0	0.0	0.4	0.0	0.0	49.4	72.8	0.0	20.8
	slight	13.6	47.2	45.4	43.8	72.2	30.4	0.0	62.0	24.0	5.0	0.0	0.4	2.0	3.0	0.8	52.8	73.8	0.0	26.5
	high	21.8	49.4	50.8	51.8	73.0	30.4	0.0	72.0	24.0	22.2	0.8	2.0	3.8	7.6	2.6	55.2	76.8	0.0	30.2
InternVL-v2-2B	strict	2.6	31.6	10.8	18.4	41.8	15.2	0.0	14.0	11.6	2.0	0.0	0.0	0.0	0.0	0.0	33.2	51.4	0.0	12.9
	slight	9.6	42.2	35.4	38.0	46.4	15.2	0.0	48.0	11.8	3.0	0.6	0.8	0.0	0.0	2.6	38.8	51.4	0.0	19.1
	high	18.8	44.8	43.8	46.4	50.6	15.2	0.0	61.8	12.4	10.2	4.6	4.8	0.4	2.4	6.0	40.8	56.4	0.0	23.3
ChartGEMMA	strict	3.6	17.8	6.6	2.0	58.4	13.0	0.0	19.2	22.6	0.0	0.0	0.0	0.0	0.0	0.0	13.0	15.8	0.0	9.6
	slight	8.6	32.8	21.8	15.6	62.4	13.0	0.0	43.6	22.6	3.4	0.4	4.4	1.4	0.4	4.6	21.8	17.8	0.0	15.3
	high	24.4	42.0	33.0	26.0	68.2	14.6	0.0	64.8	22.8	26.2	8.8	18.0	5.8	1.2	11.2	33.6	25.0	0.0	23.6
MiniCPM	strict	0.0	30.4	2.2	22.0	65.6	20.8	0.0	3.4	22.0	5.0	0.0	0.0	0.0	0.4	0.0	51.8	55.8	0.0	15.5
	slight	9.2	44.4	17.4	37.4	67.2	22.0	0.0	23.6	22.0	8.2	0.4	7.6	0.8	2.8	7.8	60.6	57.8	0.0	21.6
	high	19.6	55.2	46.2	45.6	70.8	26.8	0.0	51.2	23.2	29.0	3.0	22.2	12.4	10.4	20.6	63.0	65.8	0.0	31.4
DocOWL-1.5-Omni	strict	0.0	0.0	0.0	2.2	21.6	3.6	0.0	2.6	8.8	0.0	0.0	0.0	0.0	0.0	0.0	13.2	39.4	0.0	5.1
	slight	10.4	8.8	12.4	8.8	21.8	3.6	0.0	44.0	8.8	0.2	0.0	1.8	0.0	0.0	0.2	14.0	39.4	0.0	9.7
	high	16.4	13.4	20.8	13.2	22.0	4.0	0.0	62.8	8.8	7.8	0.0	4.0	0.0	0.0	3.2	15.0	46.2	0.0	13.2
DocOWL-1.5-chat	strict	0.0	1.4	0.0	0.0	12.0	4.0	0.0	1.2	6.8	0.0	0.0	0.0	0.0	0.0	0.0	1.8	10.6	0.0	2.1
	slight	3.2	3.4	3.0	0.6	12.0	4.0	0.0	6.4	6.8	0.0	0.0	1.8	0.0	0.0	0.0	2.0	12.6	0.0	3.1
	high	7.6	9.8	10.8	1.2	12.0	4.4	0.0	6.6	6.8	3.0	0.0	2.0	0.0	0.0	0.0	3.4	13.0	0.0	4.5

Table 3.8: Comparison of various models on the C2T task using the SCRM metric.

discrepancies appear in complex visualizations; for instance, TinyChart derives values an order of magnitude divergent from the represented data in the bubble chart (Figure 3.6). Finally, value approximation is frequent. While classified as an error, this reflects the intrinsic ambiguity of unlabeled charts, where precise extraction is arguably an ill-posed problem even for human observers.

The second most frequent error mines *structural integrity*, with parsed tables missing rows and columns, or containing superfluous cells and misaligned headers. This errors can be commonly observed with a tendency of models to ”flatten” dimensions, effectively omitting hierarchical structures. While this phenomenon is most acute in smaller models, it persists in larger architectures, as evidenced by the examples from TinyChart, ChartGEMMA, and InternVL. Notably, structural integrity errors are significantly less frequent in proprietary models compared to their open-source counterparts. This disparity is likely attributable to the massive datasets used for proprietary tuning, which presumably cover a wider spectrum of complex chart typologies. Finally, instances occur where headers are incorrectly reported, even when they are explicitly visible in the chart.

Finally, *visual perception errors* are evident in the translated charts, typically arising when the model fails to correctly classify the chart type or decipher textual content within the image. Optical Character Recognition failures are predominantly observed in smaller models. This is particularly noticeable when reported numerical values diverge from those explicitly depicted in the images, such as in pie, ring, and funnel charts. In some instances, axes are truncated, suggesting the model is unable to perceive the chart's entirety and consequently reports only a subset of the underlying tables. Furthermore, higher failure rates correlate with increased chart complexity. This is most apparent when models report incorrect column counts (e.g., in box plots, rose charts, 3D-bar charts, bubble charts, multi-type charts, radar charts, and candle charts) or hallucinate non-existent axes due to chart type misclassifications (as seen in tree-maps, rose charts, and bubble charts).

3.2.4 Applicability on Complex Real World Documents

Literature tends to prioritize successes, often under-representing edge cases. The numerical results presented in this section highlight current MLLM capabilities in the C2T task and only highlights a few edge cases in which models tend not to produce satisfying results. The largest among the tested models tend to produce satisfying results, implying deployment to real world scenarios as the next step. Through this chapter, we would like to further highlight the difficulties in effectively using this models through the analysis of a few charts, randomly

Figure 3.4: Chart to table examples with various models.

Figure 3.5: Chart to table examples with various models.

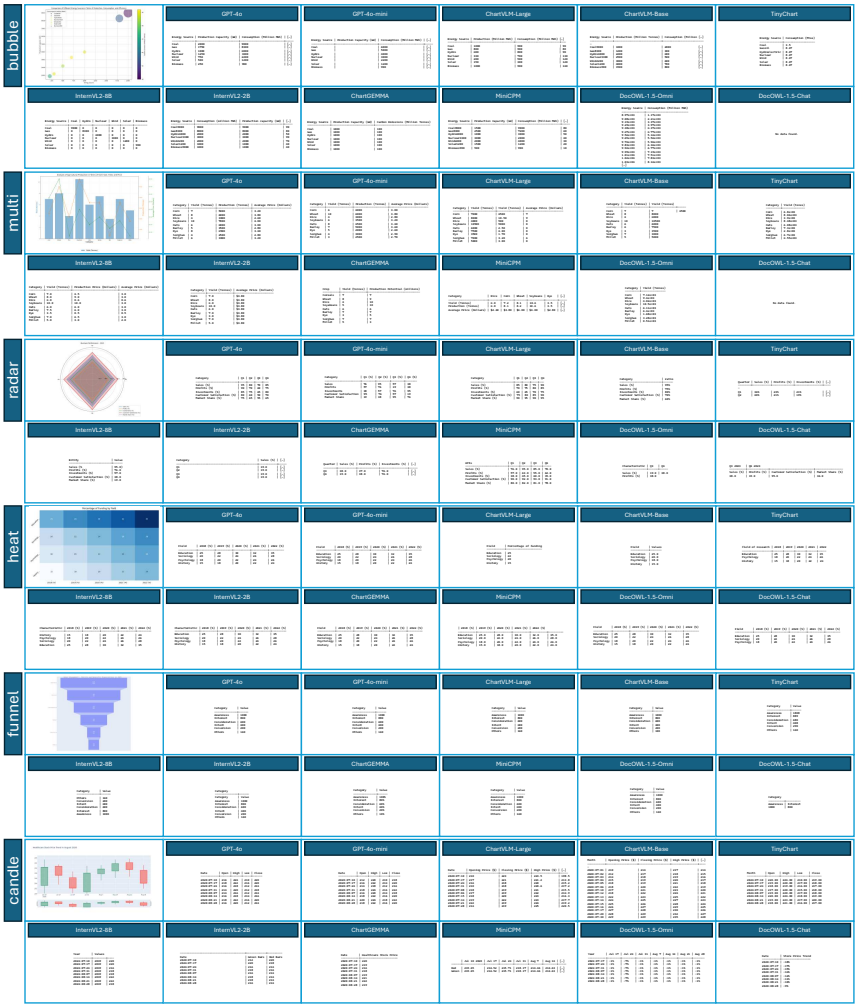


Figure 3.6: Chart to table examples with various models.

sampled from ESG documents.

We select four models, according to the results obtained from previous experiments. GPT-40, ChartVLM-large, and InternVL-v2-8B as selected for achieving the best results overall and TinyChart is chosen as a smaller alternative with satisfying performance. Moreover, we sample 10 charts from ESG document. We report some of them in Figure 3.7, Figure 3.8, Figure 3.9, and Figure 3.10 to exemplify the errors incurrent when dealing with complex real world documents.

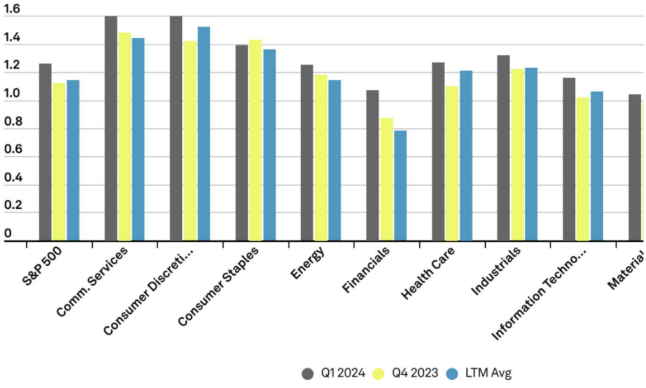
First, extrinsic hallucinations are frequent with open-source models. Exemplified in Figure 3.10, smaller models have a tendency toward reporting values that don't match with the source material. The second major issue is factual accuracy, where models report erroneous values. While it is arguably acceptable for a MLLM to report values with up to a certain percentage error, the reported results show that values can be often wrong, reverting trend and value orders. For instance, Figure 3.9 and Figure 3.8 show several values wither with opposite signs or with wrong values. Another issue is related to charts containing discrete axis but continuous values. In these cases, models focus solely on the values corresponding to the associated discrete labels, exemplified in Figure 3.10. Fourth, models hallucinate when when the visual context is occluded, as shown in Figure 3.7, where all models report values for two bars that are not included in the image.

3.2.5 Conclusions

This chapter presented the systematic evaluation of State-of-The-Art C2T models in order to assess their capabilities to map visual information to machine-readable tabular data. Several key insights have emerged regarding the maturity of automated data recovery modules. First, proprietary MLLMs currently establish an upper bound for performance while open-source alternatives are rapidly narrowing the gap. Specialized architectures achieve competitive results, with superior performance bounded to specific data distributions. This suggests that smaller, fine-tuned, LVLMs offer cost-effective alternatives to proprietary APIs. Second, the C2T task remains unsolved for complex visual modalities. The reported metrics reveal a performance collapse across all models when dealing with dense spatial data such as radar, bubble, and area charts. Moreover, the qualitative analysis highlights that current solutions lead to frequent data translation errors where visual signals are correctly identified but mapped to erroneous values. Finally, there exists a substantial discrepancy between benchmark performance and real world applicability. Our evaluation on ESG documents reveals that high scores on curated datasets does not translate into good performance on complex, noisy,

industrial environments. The prevalence of extrinsic hallucinations and the inability to infer continuous values from sparse axis labels indicate that current models rely heavily on explicit textual cues rather than true visual mathematical reasoning. Consequently, while C2T models have evolved in their semantic interpretations, their deployment in business intelligence requires rigorous human validation and the development of more factually accurate approaches.

The timeline for this experimental phase was adjusted to align with the industrial partner's production schedules and resource availability. As a result, it was not possible to investigate or develop further solutions to solve the Chart-to-Table task. For industrial partners like Altilia, commonly processed documents include charts characterized by high variance and noise. Specifically, the presence of intricate multi-type and high-granularity charts introduces a high level of difficulty, that standard models often fail to interpret correctly.



Sector	Q1 2024	Q4 2023	LTM Avg
S&P 500	1.2	1.1	1.2
Comm. Services	1.5	1.3	1.4
Consumer Discretionary	1.5	1.3	1.4
Consumer Staples	1.3	1.2	1.3
Energy	1.2	1.1	1.2
Financials	1.1	1.0	1.1
Health Care	1.0	0.9	1.0
Industrials	1.3	1.2	1.3
Information Technology	1.3	1.2	1.3
Materials	1.2	1.1	1.2

(a)

Characteristic	Q1 2024	Q4 2023	LTM Avg
S&P 500	12.5%	1.1%	1.1%
Comm. Services	1.6%	1.5%	1.4%
Consumer Discretion	0.0%	1.6%	1.4%
Consumer Staples	0.0%	1.3%	1.4%
Energy	0.0%	1.2%	1.1%
Financials	1.0%	0.9%	0.8%
Health Care	0.0%	1.2%	1.1%
Industrials	0.0%	1.2%	1.2%
Information Technology	0.0%	1.1%	1.0%
Material	0.0%	1.0%	1.0%

(b)

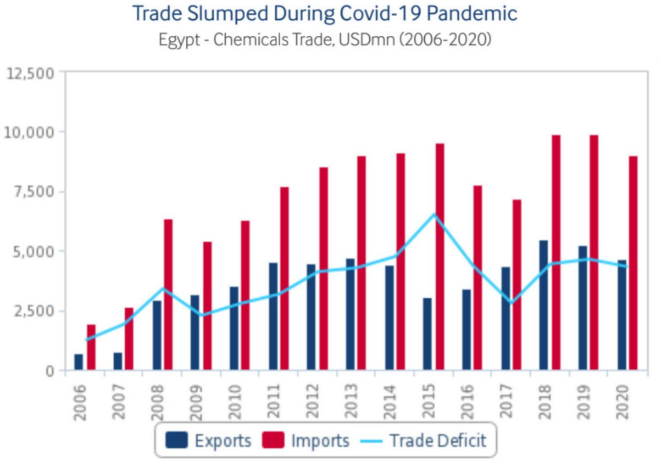
Characteristic	Q1 2024	Q4 2023	LTM Avg
S&P 500	1.2	1.1	1.1
Comm. Services	1.6	1.4	1.4
Consumer Discretionary	1.6	1.4	1.5
Consumer Staples	1.3	1.4	1.3
Energy	1.2	1.1	1.1
Financials	1.0	0.8	0.8
Health Care	1.2	1.0	1.1
Industrials	1.3	1.2	1.2
Information Technology	1.1	1.0	1.0
Materials	1.0	0.9	1.0

(c)

Characteristic	Q1 2024	Q4 2023	LTM AG
S&P 500	1.22	1.1	1.12
Commodity Services	1.58	1.45	1.41
Consumer Discretionary	1.59	1.39	1.49
Consumer Staples	1.36	1.41	1.37
Energy	1.2	1.13	1.09
Financials	1.02	0.8	0.69
Health Care	1.23	1.05	1.15
Industrial	1.3	1.17	1.18
Information Technology	1.1	1.0	1.08
Material	1.02	0.95	1.03

(d)

Figure 3.7: Example of chart parsed from an ESG document. (a) is GPT-4o, (b) is ChartVLM-large, (c) is InternVL-v2-8B, (d) is TinyChart.



Year	Exports	Imports	Trade Deficit
2006	500	1,000	500
2007	1,000	2,500	1,500
2008	2,500	5,000	2,500
2009	2,000	4,000	2,000
2010	2,500	5,000	2,500
2011	3,000	6,000	3,000
2012	3,500	7,000	3,500
2013	4,000	8,000	4,000
2014	4,500	9,000	4,500
2015	5,000	10,000	5,000
2016	3,000	7,500	4,500
2017	3,500	8,000	4,500
2018	4,000	10,000	6,000
2019	4,500	10,000	5,500
2020	4,000	9,000	5,000

(a)

Year	Exports	Imports	Trade Deficit
2020	4509	4467	9034
2019	5212	5176	4673
2018	4085	5451	4524
2017	3118	4285	3109
2016	3378	3353	4054
2015	3034	6567	5973
2014	4447	4673	5248
2013	4577	4582	4240
2012	4388	4350	4152
2011	4484	3205	3389
2010	3473	3484	2880
2009	2378	3179	2348
2008	2954	3241	3353
2007	757	1693	2276
2006	669	845	1624

(b)

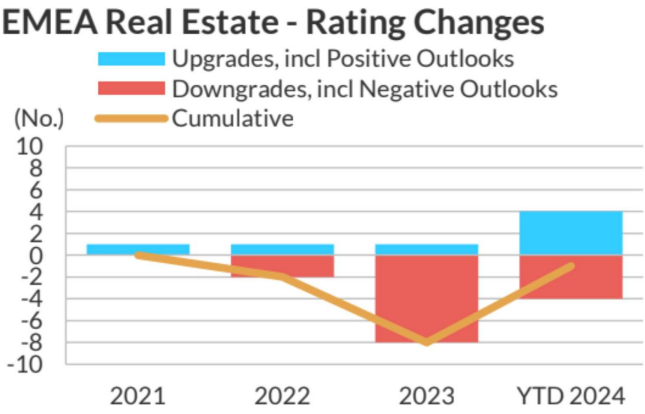
Year	Exports	Imports	Trade Deficit
2006	0	0	0
2007	0	0	0
2008	0	0	0
2009	0	0	0
2010	0	0	0
2011	0	0	0
2012	0	0	0
2013	0	0	0
2014	0	0	0
2015	0	0	0
2016	0	0	0
2017	0	0	0
2018	0	0	0
2019	0	0	0
2020	0	0	0

(c)

Year	Exports	Imports	Trade Deficit
2020	4615	8791	-
2019	5137	9667	-
2018	5274	9692	4362
2017	4352	7003	2696
2016	3302	7582	4448
2015	2921	9332	6295
2014	4242	8882	4642
2013	4642	8797	4294
2012	4349	8276	3810
2011	4643	7458	3118
2010	3315	6105	2886
2009	2925	5426	2437
2008	2796	6179	3323
2007	785	2437	1753
2006	602	1853	1610

(d)

Figure 3.8: Example of chart parsed from an ESG document. (a) is GPT-4o, (b) is ChartVLM-large, (c) is InternVL-v2-8B, (d) is TinyChart.



Year	2021	2022	2023	YTD 2024
Upgrades, incl Positive Outlooks	1	1	1	4
Downgrades, incl Negative Outlooks	−2	−3	−4	−3
Cumulative	−1	−2	−3	1

(a)

Year	2021	2022	2023	YTD 2024
Upgrades, incl Positive Outlooks	0.5	−0.8	−7.2	4.0
Downgrades, incl Negative Outlooks	0.2	−2.5	−8.3	2.7
Cumulative	1.2	1.8	2.0	2.4

(b)

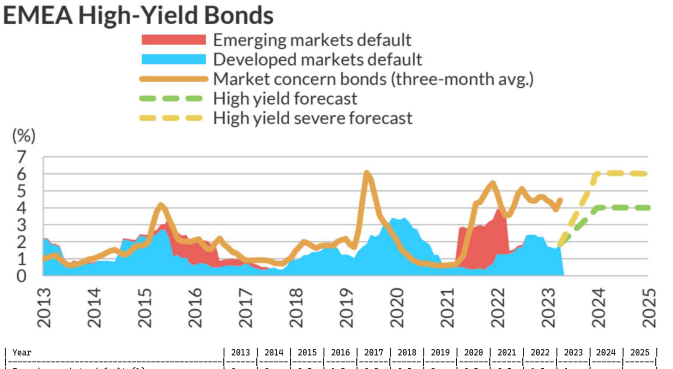
Year	2021	2022	2023	YTD 2024
Upgrades	0	0	0	4
Downgrades	0	0	0	2
Cumulative	0	0	0	6

(c)

Year	2024	2023	2022	2021
Upgrades, incl. Positive Outlays	2.0	0.5	0.5	0.5
Downgrades, incl. Negative Outlays	2.0	−4.5	−1.0	−1.5
Cumulative	0.0	−1.0	−1.5	0.0

(d)

Figure 3.9: Example of chart parsed from an ESG document. (a) is GPT-4o, (b) is ChartVLM-large, (c) is InternVL-v2-8B, (d) is TinyChart.



Year	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025
Emerging markets default (%)	0	0	0.5	1.5	0.5	0.5	2	0.5	0.5	1.5	1		
Developed markets default (%)	0	0	0.5	1.5	0.5	0.5	2	0.5	0.5	1.5	1	0	0
Market concern bonds (three-month avg.) (%)	0.5	0.5	1	2	1	1	3	1	1	2	1.5		
High yield forecast (%)												1.5	1.5
High yield severe forecast (%)	0	0	0	0	0	0	0	0	0	0	0	2.5	2.5

(a)

Year	2013	2014	2015	2016	2017	2018	2019*	2020	2021	2022	2023	2024	2025
Emerging markets default	1.2	1.1	1.2	1.3	1.4	1.5	1.7	1.9	2.1	2.6	2.8	3.0	3.0
Developed markets default	1.5	1.1	1.2	1.3	1.4	1.6	1.7	2.1	2.2	2.5	2.7	3.2	3.3
Market concern bonds (three-month avg.)	1.1	0.9	1.3	1.4	1.5	1.7	2.0	3.1	3.5	4.2	4.9	5.8	6.0
High yield forecast	0.7	0.7	0.9	1.1	1.1	1.2	1.3	3.4	3.5	4.7	5.1	6.1	6.2
High yield severe forecast	0.2	0.2	0.4	0.2	0.2	0.1	0.2	0.4	0.4	0.5	0.6	0.7	0.7

(b)

Year	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025
Emerging markets default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Developed markets default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Market concern bonds (three-month avg.)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
High yield forecast	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
High yield severe forecast	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

(c)

Year	2025	2024	2023	2022	2021	2020	2019	2018	2017	2016	2015	2014	2013
Emerging markets deficit	–	–	1.2	1.6	1.5	2.2	0.7	0.3	0.1	0.7	1.2	0.5	1.1
Developed markets deficit	–	–	1.2	1.6	0.1	1.7	0.7	0.3	0.2	0.5	1.4	0.4	0.6
Market concern bonds (three-month average)	–	–	3.5	4.5	0.5	3.5	1.5	1.2	0.5	1.2	3.5	0.7	0.5
High yield forecast	–	2.5	3.2	3.5	0.5	0.5	1.2	0.5	0.5	1.2	1.2	0.5	0.5
High yield severe forecast	5.2	5.2	2.2	–	–	–	–	–	–	1.2	1.2	0.5	0.5

(d)

Figure 3.10: Example of chart parsed from an ESG document. (a) is GPT-4o, (b) is ChartVLM-large, (c) is InternVL-v2-8B, (d) is TinyChart.

3.3 A Short Evaluation of Text Summarization

The increasingly fast growth of digital textual content has rendered human comprehension of available sources increasingly difficult. Automatic Text Summarization (ATS) aims at the distillation of the most critical knowledge from a source to produce an abridged version that retains the key points of the original text. The recent advent of LLMs has fundamentally shifted the research landscape, with models promising to produce high-quality summaries without task-specific fine-tuning. This chapter presents a short evaluation of modern text summarization approaches, focusing on single-document summarization, and the comparative performance of prompting strategies and parameter-efficient fine-tuning.

In the era of Transformers, standard fine-tuning techniques established strong baselines for extractive and abstractive summarization. However, the recent innovations suggests that LLMs can match, and sometimes exceed, the quality of the summaries generated by fine-tuned models. Consequently, our evaluation prioritizes the assessment of instruction tuned LLMs over older architectures. This chapter explores the first dimension of summarization, utilizing multilingual datasets to evaluate LLMs on the Single Document Summarization task. We evaluate two models, both utilized by Altilia, and currently state of the art on numerous tasks. Moreover, we evaluate different prompting strategies designed to generate high quality summaries. Finally, we analyze how different prompt templates influence the output's syntactic and semantic quality.

3.3.1 Experimental Design

This section evaluates the capabilities of LLMs on the single-document summarization task. We assess performance of different models across both English and Italian, utilizing various prompting techniques. The target is the generation of informative, concise, and factually consistent summaries.

Datasets

To provide a comprehensive evaluation of LLM capabilities for single-document summarization, we select four datasets equally divided between English and Italian. For the English part, we utilize two news summarization datasets: CNN/DailyMail [152] and XSum [153]. For the Italian dataset, we employed FanPage and IlPost [107]. The dataset statistics are available in Table 3.9.

Dataset	Language	Size		Tokens		
		Train	Val	Test	Article	Summary
CNN/DailyMail [152]	English	287,113	13,368	11,490	810.57	56.20
XSum [153]	English	204,045	11,332	11,334	431.07	23.26
FanPage [107]	Italian	64,446	8,431	8,431	312.70	43.85
IlPost [107]	Italian	35,220	4,402	4,403	174.43	26.39

Table 3.9: Single-document summarization datasets.

CNN/DailyMail. This corpus is among the biggest in abstractive text summarization. The original dataset [83] is constituted by human-written abstractive bullet-point answers from CNN and Daily Mail news. The version we are using repurposed the bullet point lists into abstractive summaries [152].

XSum. The corpus constituting this dataset is collected from BBC. The provided summaries are reduced to a single sentence explaining the main topic from the article.

FanPage and IlPost. These corpora are collected from the homonym Italian news websites. Both contain news articles associated with a title and a short description. The title and the description are concatenated to form the document summary while each original article constitutes an input document.

Prompting Strategies

We explored several prompt templates to test the capabilities of LLMs to different summarization styles.

Abstractive Prompts. The abstractive prompt template defines our baseline, as it only tasks the model with the generation of a concise summary. A simple variant of this template instructs the model with the generation of a summary of approximately m sentences, where m is the number of sentences in the gold summary. Finally, we define a task specification (ts) variant for the Italian datasets, since the ground truth summaries are the concatenation of a title and a short description.

Extractive Prompts. These templates only instruct the model to extract specific sentences from the source text to form a summary.

Chain of Density Prompting. Chain of Density (CoD) [2] prompting instructs the model to iteratively prompt the model to generate increasing entity-dense summaries. First, the model generates a summary. Then, it automatically and iteratively identifies missing entities to fuse into denser revisions without increasing the overall output length.

Chain of Extractions. Similarly to the *extract-then-generate* technique [240], this template merges the advantages of extractive and abstractive methods. The model is tasked with two sequential instructions. First, the model must extract meaningful sentences from the input document, thus removing the risk of hallucinations. Then, extractive summary must be transformed into an abstractive one. We name this approach Chain of Extractions (CoE) since the model is requested to perform all operations in a single forward pass.

Models, Costs, and Limitations

To assess LLMs ability to correctly generate abstractive summaries, we rely on proprietary models. This choice is dictated by several factors. First, proprietary models have achieved significantly better results than open-source counterparts on a variety of tasks. Second, open-source models are limited in context length, overall output quality, and controllability [239, 249]. Finally, open-source model deployment results in significant costs and under-utilization. The models choice was limited to two versions of GPT-3.5 and the recently released GPT-4. For GPT-3.5, we employed gpt-3.5-1106 and gpt-3.5-instruct. As far as GPT-4 is concerned, we only utilized gpt-4-turbo-preview.

The experiments described in this chapter are conducted at the early stages of LLM serving. For this reason, the price of proprietary APIs was quite high with respect to today's standard and hardware for fast inference was quite limited. The cost estimates for the FanPage dataset are reported in Table 3.10 and led to the selection of a subset of experiments. Specifically, we decided upon using all prompt templates up to Extractive. As far as CoD is concerned, it was only used in its standard form. Finally, for CoE we decided to use it already with sentence limit instructions and, whenever possible, task specifications.

Evaluation Metrics

To assess model performance, we employ a suite of metrics targeting different summary dimensions, drawing from syntactic overlap and semantic similarity.

Prompt	Tokens (M)		Cost ($)		
Template	in	out	gpt-3.5-turbo-1106	gpt-3.5-turbo-instruct	gpt-4-turbo-preview
Abstractive	4.9	0.7	6.30	8.75	70.08
+ TS	5.1	0.7	5.56	9.13	70.62
+ SL	4.9	0.7	6.36	8.83	70.59
+ TS + SL	5.2	0.7	6.62	9.23	73.29
Extractive	5.0	0.7	6.25	8.79	70.12
CoD	9.0	0.7	19.17	23.68	242.59
CoE	6.0	0.7	20.56	23.56	278.49

Table 3.10: Cost estimates for the FanPage dataset.

While LM-based metrics can provide deeper insight, these are applied solely to the English datasets due to the limited availability of multi-lingual checkpoints.

Syntactic Overlap. We utilize ROUGE [118], BLEU [161], and TER [193] for syntactic evaluation. These n-gram-based systems quantify the lexical overlap between the system output and the reference summary. While ROUGE is standard for summarization, we include BLEU and TER, typically used in translation, to provide a broader perspective on precision and edit distance, particularly for the Italian datasets.

Semantic Similarity. To capture meaning beyond exact lexical matches, we employ entailment-based metrics. Specifically, we use BERTScore [246], instantiating a SciBERT [18] checkpoint, and BLEURT [186].

LM-based Evaluation. We utilize FactCC [101] and QAFactEval [62] to measure the factual alignment between the generated summary and the source document. These metrics are critical for evaluating whether abstractive methods are hallucinating information. Moreover, we employ BLANC [212] to estimate the general quality of the summaries without human intervention. Due to the uni-lingual nature of these models' training datasets, LM-based metrics for the Italian datasets are not provided.

Natural Language Generation Statistics. We utilize the NLTK [23] library to calculate the number of tokens and Spacy to determine entities. We compute *entity density* as the ratio of entities to tokens. *Abstractiveness* is measured as the average

model	template	BLEU	R-1	R-2	R-L	R-LS	TER	BS	BRT	BLANC	FactCC	QAFactEval
Proprietary Models												
gpt-3.5-1106	abs	7.47	35.61	13.26	22.58	22.98	141.88	73.5	-38.25	15.18	50.90	3.73
	abs_{sl}	7.74	36.28	13.65	23.13	23.54	138.83	73.57	**-37.57**	15.24	20.26	3.66
	ext	**9.39**	**37.35**	**15.27**	**24.32**	**24.76**	**132.67**	**73.75**	-44.22	**17.45**	**58.49**	**4.10**
	CoD	6.57	25.74	8.55	16.94	17.22	142.59	70.02	-83.19	13.86	31.14	3.75
	CoE	7.84	35.42	13.85	22.91	23.35	151.87	71.2	-75.8	16.99	54.44	3.95
gpt-3.5-instruct	abs_{sl}	7.3	35.76	12.74	22.57	23.04	132.07	73.56	-39.94	11.91	58.15	3.53
gpt-4	abs_{sl}	5.17	32.91	11	20.05	20.43	174.56	72.54	-38.63	15.33	42.32	3.25
Open-Source Models												
MatchSum [253]		-	44.41	20.86	40.55	-	-	-	-			
BRIO [127]		-	**47.78**	**23.55**	**44.63**	-	-	-	-	**12.17**	**36.99**	**4.14**
T0 [183]		-	-	-	-	-	-	-	-	8.89	20.12	3.98
GPT3-D2 [74]		-	-	-	-	-	-	-	-	9.83	24.28	3.83
Pegasus [242]		-	-	-	-	-	-	-	-	11.37	51.52	4.47

Table 3.11: CNN/DailyMail syntactic and semantic metrics.

square length of extractive fragments [78]. *Content distribution* is assessed as the average position of the fragments within the input document [2]. Additionally, we compute *fusion* as the average number of source sentences aligned with summary sentences [2].

3.3.2 Results and Analysis

English Datasets

Metrics. Table 3.11 and Table 3.12 present the results for CNN/DailyMail and XSum, respectively. On the highly extractive CNN/DailyMail, the extract-

model	template	BLEU	R-1	R-2	R-L	R-LS	TER	BS	BRT	BLANC	FactCC	QAFactEval
Proprietary Models												
gpt-3.5-1106	abs	1.62	19.42	4.96	13.45	13.47	368.73	69.73	-60.03	19.49	53.5	3.79
	abs_{sl}	**2.65**	**24.87**	**6.15**	**17.62**	**17.61**	180.98	**71.86**	**-52.41**	13.76	23.42	3.18
	ext	2.32	21.28	4.34	15.79	15.82	**117.84**	71.25	-67.08	12.08	62.33	**4.18**
	CoD	0.72	14.8	2.75	10.22	10.21	444.92	66.64	-91.81	**26.53**	51.44	4.04
	CoE	1.23	17.46	3.94	12.04	12.03	428.38	68.76	-67.29	23.51	**65.22**	4.08
gpt-3.5-instruct	abs_{sl}	1.87	21.87	5.15	15.18	15.18	273.5	70.68	-59.06	14.94	33.44	3.39
gpt-4	abs_{sl}	2.29	23.81	5.95	16.66	16.66	209.42	71.08	-56.85	13.05	14.33	2.73
Open-Source Models												
MatchSum [253]		-	24.86	4.66	18.41	-	-	-	-	-	-	-
BRIO [127]		-	**49.07**	**25.13**	**40.4**	-	-	-	-	2.3	20.31	1.86
T0 [183]		-	-	-	-	-	-	-	-	2.38	22.19	2.03
GPT3-D2 [74]		-	-	-	-	-	-	-	-	**5.94**	**39.77**	**2.95**
Pegasus [242]		-	-	-	-	-	-	-	-	2.49	24.65	2.00

Table 3.12: XSum syntactic and semantic metrics.

ive template is favored by ROUGE and BLEU scores in the GPT-3.5 variants. Furthermore, providing explicit length instructions (*abssl*) yields a noticeable improvement over the standard *abs* baseline. Conversely, on the highly abstractive XSum dataset, the extractive solution is significantly outperformed by the abstractive *abssl* template. In both datasets, however, clear length constraints consistently enhance summary quality compared to unconstrained generation.

Concerning the most complex prompting strategies, CoD consistently underperforms on syntactic metrics across both datasets. CoE shows mixed results: it performs on par with abs_{sl} on CNN/DailyMail but lags significantly behind on XSum. Among the proprietary models, the legacy *gpt-3.5-instruct* consistently under-performs compared to *gpt-3.5-1106*. Surprisingly, *gpt-4* also falls short of *gpt-3.5-1106* across nearly all syntactic metrics in this context. Finally, while the fine-tuned state-of-the-art models [253, 127] achieve significantly higher ROUGE scores than the LLMs, a different picture emerges when analyzing semantic quality.

When evaluating semantic consistency and factuality, the results diverge from the syntactic trends. On CNN/DailyMail, the extractive approach extends to be superior on factuality, surpassing the fine-tuned benchmarks and achieving about the same result on QAFactEval. On XSum, while BRIO dominates syntactic metrics, it performs poorly on factuality, scoring lowest on QAFactEval and BLANC. In contrast, the GPT models maintain high factuality scores despite lower ROUGE overlaps. Notably, while CoD and CoE struggled with ROUGE, they excel here: CoD achieves the highest BLANC score and CoE achieves the highest FactCC.

Overall, the results show that while fine-tuned models excel at mimicking the syntax of the source documents, LLMs are generally better with factual consistency and semantic fidelity.

Summarization Statistics. Table 3.13 and Table 3.14 indicate that model architecture is the primary driver of summarization quality, with GPT-4 demonstrating superior abstraction and fusion compared to GPT-3.5 variants. Specifically, GPT-4 achieves the highest degrees of abstraction and fusion, reflecting a shift from verbatim copying to novel synthesis. Prompting strategies introduce distinct trade-offs: the extractive template (*ext*) maximizes entity retention but sacrifices fluency and coverage, particularly on XSum. Conversely, the CoD template yields the widest content distribution but proves highly unstable. It exhibits extreme variance in abstractiveness and fails to adhere to the natural brevity constraints of the XSum dataset, producing summaries significantly longer than standard baselines. Ultimately, while prompt engineering alters structural metrics, superior generation

model	template	tokens	entities	entity density	abstractiveness ↓	fusion	content distribution
gpt-3.5-1106	abs	100.08 (26.45)	7.63 (3.68)	0.08 (0.04)	3.57 (2.39)	1.43 (0.4)	0.44 (0.11)
	abs_{sl}	98.31 (15.82)	7.77 (3.26)	0.08 (0.04)	3.29 (1.76)	1.5 (0.4)	0.41 (0.13)
	ext	100.37 (17.21)	11.33 (3.64)	0.11 (0.04)	9.24 (8.53)	0.75 (0.23)	0.32 (0.15)
	CoD	82.96 (95.72)	9.3 (8.0)	0.14 (0.07)	15.98 (65.57)	1.76 (1.41)	0.57 (0.23)
	CoE	107.19 (42.28)	8.98 (5.12)	0.09 (0.04)	5.77 (6.77)	1.48 (0.41)	0.37 (0.14)
gpt-3.5-instruct	abs_{sl}	92.79 (18.67)	7.17 (3.61)	0.08 (0.04)	2.61 (1.25)	1.41 (0.35)	0.43 (0.12)
gpt-4	abs_{sl}	123.92 (14.9)	10.21 (4.14)	0.08 (0.03)	1.64 (0.6)	1.74 (0.44)	0.42 (0.11)

Table 3.13: CNN/DailyMail statistics. ↓ indicates that lower is better.

capabilities in GPT-4 provide the most consistent balance of abstractiveness.

Italian Datasets

Table 3.15 and Table 3.16 report the results on the Italian datasets. A distinct performance gap is observed between the fine-tuned, supervised models and the Large Language Models in this specific linguistic context. The supervised state-of-the-art models demonstrate a clear superiority in surface-level matching. mBART achieves the highest scores across all major metrics for both datasets (e.g., reaching 38.91 R-1 on IlPost), followed closely by the Italian-specific IT5. This underscores the efficacy of in-domain and language-specific fine-tuning over zero-shot or few-shot prompting for strict syntactic reconstruction. In contrast, models trained on English-centric data, such as Pegasus-XSum and Pegasus-CNN/DailyMail, perform poorly, highlighting the difficulty of cross-lingual transfer without specific adaptation.

Regarding the LLMs, prompt engineering plays a crucial role in narrowing the gap with supervised models. The combined strategy abs_{ts} proves to be the most effective configuration for gpt-3.5-1106, achieving the highest BLEU and ROUGE scores among the GPT variants on both datasets. It significantly outperforms the basic *abs* and length-constrained $abs_{ts;sl}$ baselines. Consistent

model	template	tokens	entities	entity density	abstractiveness ↓	fusion	content distribution
gpt-3.5-1106	abs	91.66 (25.97)	7.62 (3.72)	0.09 (0.04)	3.54 (2.4)	1.41 (0.36)	0.47 (0.12)
	abs_{sl}	45.91 (11.53)	4.54 (2.38)	0.1 (0.05)	2.51 (1.53)	2.51 (0.99)	0.33 (0.19)
	ext	29.81 (7.48)	4.44 (1.86)	0.15 (0.06)	8.35 (7.38)	0.73 (0.34)	0.16 (0.16)
	CoD	109.28 (99.73)	11.74 (9.47)	0.13 (0.06)	20.57 (47.69)	1.85 (1.69)	0.57 (0.2)
	CoE	105.91 (45.63)	9.36 (5.79)	0.09 (0.04)	11.2 (24.08)	1.38 (0.36)	0.43 (0.14)
gpt-3.5-instruct	abs_{sl}	68.03 (26.25)	6.33 (3.6)	0.1 (0.05)	2.29 (1.18)	2.2 (1.04)	0.42 (0.17)
gpt-4	abs_{sl}	53.41 (11.61)	5.53 (2.55)	0.1 (0.04)	1.27 (0.67)	2.77 (1.09)	0.35 (018)

Table 3.14: XSum statistics. ↓ indicates that lower is better.

model	template	BLEU	R-1	R-2	R-L	R-LS	TER	BS	BRT
Proprietary Models									
gpt-3.5-1106	abs	5.17	28.39	11.06	19.46	19.43	209.57	70.85	-39.27
	abs_{sl}	5.91	29.95	11.80	20.66	20.65	167.76	71.34	-38.50
	abs_{ts}	5.79	28.19	11.47	20.85	20.85	201.53	71.26	-41.76
	$abs_{ts;sl}$	**7.03**	**30.29**	**12.67**	**23.07**	**23.07**	144.60	72.11	-40.44
	ext	5.77	29.10	11.52	20.31	20.30	178.21	71.04	-40.54
	CoD	4.26	23.36	8.48	16.60	16.59	189.82	**69.19**	-55.19
	CoE	6.02	27.86	11.02	21.47	21.48	158.35	71.03	-43.41
gpt-3.5-instruct	abs_{sl}	4.63	27.20	10.37	18.44	18.42	233.43	70.54	-38.99
	$abs_{ts;sl}$	5.37	29.06	11.20	19.93	19.94	189.26	71.09	**-38.01**
gpt-4	abs_{sl}	4.03	26.72	9.20	17.81	17.82	206.66	70.01	-41.32
	$abs_{ts;sl}$	5.72	29.68	11.53	22.28	22.30	**142.23**	71.39	-42.79
Open-Source Models									
IT5 [107]		-	33.78	16.29	27.48	30.23	-	-	-
mBART [107]		-	**38.91**	**21.38**	**32.05**	**35.07**	-	-	-
Pegasus-XSum [107]		-	21.03	6.63	16.10	16.07	-	-	-
Pegasus-CNN/DailyMail [107]		-	23.96	7.72	16.81	16.81	-	-	-

Table 3.15: IlPost syntactic metrics.

model	template	BLEU	R-1	R-2	R-L	R-LS	TER	BS	BRT
Proprietary Models									
gpt-3.5-1106	abs	7.06	33.26	12.68	21.73	21.74	131.96	71.84	**-36.85**
	abs_{sl}	7.12	33.28	12.51	21.58	21.59	130.08	71.70	-37.39
	abs_{ts}	7.32	32.85	12.86	22.29	22.28	127.17	71.71	-43.32
	$abs_{ts;sl}$	**7.96**	**33.46**	**13.33**	**22.83**	**22.79**	112.9	**71.97**	-43.45
	ext	7.87	32.93	12.65	22.09	22.08	113.44	71.85	-38.68
	CoD	5.51	26.65	9.43	17.64	17.65	130.72	69.49	-54.21
	CoE	7.72	30.45	11.62	21.75	21.73	**104.47**	70.85	-46.58
gpt-3.5-instruct	abs_{sl}	6.29	32.25	11.98	20.65	20.62	150.96	71.46	-37.23
	$abs_{ts;sl}$	6.75	32.75	12.14	21.14	21.13	137.96	71.68	-37.35
gpt-4	abs_{sl}	5.15	30.81	10.66	19.18	19.19	168.23	70.85	-38.97
	$abs_{ts;sl}$	6.22	31.93	11.70	21.23	21.22	132.01	71.23	-44.17
Open-Source Models									
IT5 [107]		-	33.83	15.46	24.90	28.31	-	-	-
mBART [107]		-	**36.50**	**17.44**	**26.17**	**30.26**	-	-	-
Pegasus-XSum [107]		-	20.10	6.49	14.78	14.76	-	-	-
Pegasus-CNN/DailyMail [107]		-	26.82	9.02	18.10	18.10	-	-	-

Table 3.16: FanPage syntactic metrics.

with the English datasets, CoD consistently under-performs on syntactic metrics, yielding the lowest ROUGE scores among the GPT-3.5 configurations. CoE performs competitively, particularly on FanPage where it achieves the best TER, but it generally trails behind the $abs_{ts;sl}$ strategy.

3.4 Conclusions

In this chapter, we presented a comprehensive evaluation of text summarization through LLMs. By analyzing the performance across diverse datasets, critical patterns emerged regarding the efficacy of prompting strategies, and the differ-

ences between syntactic and semantic quality. First, the experimental results highlight the divergence between lexical overlap and semantic fidelity. While fine-tuned models like BRIO dominate n-gram-based metrics, they lack in factual consistency. In contrast, proprietary LLMs demonstrate superior performance in entailment-based and factuality metrics, suggesting that fine-tuned models excel at mimicking stylistic patterns while LLMs posses robust abilities to generate factually consistent synopses. Second, our evaluation underscores the importance of language-specific fine-tuning. While zero-shot Large Language Models perform competitively, fine-tuned models like mBART and IT5 substantially outperform all other models on syntactic metrics. Third, we observed that increasingly complex prompting strategies do not inherently yield superior performance. Simple prompt engineering, constraining the model to control length and defining the task, consistently provide the best balance between quality and stability. Conversely, complex reasoning chains, such as CoD proved unstable, often failing to adhere to length constraints and exhibiting high variance in abstractiveness. CoE showed promise in enhancing factuality. Instead, it incurred higher computational costs without consistently outperforming optimized standard prompts in syntactic quality.

While fine-tuned architectures remain State-of-The-Art for extractive and highly standardized summaries, LLMs offer a satisfactory alternative characterized by high factuality, impressive abstraction, and satisfactory robustness. The shift in performance metrics suggests that future research should prioritize semantic and factual evaluation over rigid lexical overlap. In conclusion, LLMs rank as the preferred engine for reliable, open-domain, abstractive summarization.

3.5 A Short Evaluation of Aspect-Based Summarization

Our last evaluation moves beyond generic summaries and focuses on Aspect-based Summarization, prioritizing user-specific requests and keywords. While generic summarization is mature, aspect-based summaries remain complex, requiring conditioning on specific topics. To investigate this task, we employ tailored dataset to investigate whether prompting techniques can improve the outcome.

In this section, we evaluate LLM capabilities in aspect-based summarization with various models and techniques. Our goal, is the identification of the optimal approach to generate summaries tailored for a well-defined target user. Our experiments aim to address two research needs. First, while LLMs possess extraordinary capabilities for generic abstractive summarization, their ability to control the generated output is under-explored. Second, recent studies revealed that smaller models can achieve superior performance without fine-tuning, when equipped with simple prompting techniques. This would allow fast deployment in new domains without the need to fine-tune smaller models, which results expensive in terms of dataset annotation.

3.5.1 Selected Dataset

For our experimental evaluation we selected two datasets. **AclSum** [203] is a summarization datasets sourced from prominent conferences. The dataset comprises 250 NLP research papers and aims at the generation of summaries for three topics: challenge, approach, and outcome. The second dataset is **NewTS** [15], derived from CNN/DailyMail. The dataset comprises about 3,000 documents paired with topics. The topics are represented by sets of words and sentences derived from the dataset using Latent Dirichlet Allocation (LDA) [26]. The datasets statistics are reported in Table 3.17.

3.5.2 Prompting Strategies, Evaluation, and Fine-Tuning

Since we already evaluated summary quality for single document summarization under various settings, we are aware that providing clear instructions about the summary length and content systematically improves the summary quality. This time, instead of relying solely on prompt engineering, we evaluate LLMs for few-shot summary generation and compare them against prompt engineering. On AclSum, where summaries are written in a single sentence, we develop three

Dataset	Docs			Tokens		Aspects
	train	val	test	document	summary	
AclSum [203]	100	50	100	914.7		3
Challenge					22.5	
Approach					22.7	
Outcome					21.3	
NewTS [15]	2400	-	600	602.0	74.0	50

Table 3.17: Aspect-base summarization datasets statistics.

strategies: (i) generic summarization, (ii) single-sentence summarization, and (iii) summarization from entity signals, where the model is asked to generate summaries containing certain entities from the source document. On NewTS, where topic words and topic sentences are provided, we employ the following strategies on top of the ones previously employed for AclSum: (i) topic-words conditioned summarization, and (ii) topic-sentences conditioned summarization.

For evaluations we select two models currently used by Altilia. The first is GPT-3.5. The second is LLaMa3, currently deployed in production is Altilia machines. Finally, as far as the evaluation metrics are concerned, we select ROUGE and BERTScore due to the good behavior exhibited in previous experiments.

3.5.3 Experimental Evaluations

We report the metrics from the selected models on AclSum on Table 3.18. A consistent trend observed across both models and prompting strategies is the benefit of few-shot learning. Increasing the number of in-context examples generally leads to improved generation quality. GPT-3.5 shows steady improvements as k increases. For example, in topic-based summarization, the Overall R-1 score rises from 20.05 (0-shot) to 23.69 (4-shot). On the other hand, LLaMa3 demonstrates a much more dramatic response to in-context examples. While its 0-shot performance is often lower than GPT-3.5, it rapidly overtakes the latter with just a few examples. In the topic-based setting, LLaMa3's Overall R-1 score jumps from 14.62 (0-shot) to 38.02 (4-shot), significantly outperforming GPT-3.5's best configuration. The results highlight a trade-off between zero-shot robustness and few-shot adaptability. GPT-3.5 generally exhibits superior performance in the zero-shot setting. In topic-based summarization, it achieves an Overall R-1 of 20.05 compared to LLaMa3's 14.62. Similarly, for entity-based prompting, GPT-3.5 starts at 15.78, surpassing LLaMa3's 12.10. This suggests that GPT-3.5 may have

	Overall				Challenge				Approach				Outcome			
Model	R-1	R-2	R-L	BS	R-1	R-2	R-L	BS	R-1	R-2	R-L	BS	R-1	R-2	R-L	BS
GPT-3.5																
topic																
0-shot	20.05	7.43	14.94	63.03	16.74	5.27	11.96	59.98	21.93	9.13	16.94	65.10	21.46	7.95	15.96	64.01
1-shot	18.73	6.99	14.02	62.50	15.54	4.45	11.13	59.38	20.15	8.90	15.74	64.41	20.59	7.61	15.21	63.71
2-shot	20.54	7.47	15.35	63.14	16.26	4.69	11.61	59.41	22.70	9.53	17.50	65.48	22.75	8.25	16.89	64.52
4-shot	23.69	8.85	17.74	64.50	19.32	5.63	13.62	60.92	25.76	10.88	20.08	66.48	26.13	10.05	19.46	66.09
entities																
0-shot	15.78	7.41	13.22	57.12	19.93	3.71	14.56	59.52	23.85	10.37	19.74	61.66	19.14	9.19	15.79	58.32
1-shot	26.87	9.36	19.66	65.34	21.08	5.70	14.68	61.51	29.79	12.23	22.81	67.75	29.71	10.18	21.51	66.75
2-shot	28.29	10.19	21.29	66.16	21.53	6.01	15.32	62.05	32.33	13.58	25.91	69.12	31.01	11.07	22.65	67.31
4-shot	28.90	10.35	21.56	66.46	22.53	6.23	15.87	62.70	32.64	13.68	25.57	69.05	31.56	11.16	23.30	67.62
LLaMa3																
topic																
0-shot	14.62	6.16	11.34	59.47	12.30	4.27	9.16	56.47	15.72	7.09	12.40	61.24	15.88	7.04	12.47	60.71
1-shot	23.93	10.67	18.67	64.28	19.10	7.73	13.66	60.75	24.62	12.86	20.29	65.89	27.71	11.26	22.14	66.21
2-shot	35.55	15.29	27.62	68.79	32.36	12.30	23.66	67.11	36.26	18.30	30.30	70.00	37.23	14.72	28.49	69.25
4-shot	38.02	17.52	31.67	69.94	30.90	13.38	24.37	67.01	41.84	21.90	36.36	72.71	40.39	16.84	33.12	70.08
entities																
0-shot	12.10	4.91	10.08	55.50	10.80	3.18	6.82	54.78	15.36	6.86	12.10	59.15	15.10	6.80	12.18	58.74
1-shot	16.18	7.64	13.56	58.65	4.57	2.44	3.85	52.65	24.32	10.69	20.17	63.44	19.56	9.42	16.17	59.85
2-shot	34.59	12.96	26.69	68.21	29.64	10.21	21.60	65.74	37.97	16.22	30.86	70.90	35.19	12.46	27.13	67.99
4-shot	33.76	12.97	25.82	68.07	24.89	7.96	17.18	63.89	37.24	15.76	30.80	71.20	38.74	14.92	29.11	69.13

Table 3.18: Aspect-base summarization metrics on AclSum.

better instruction-following capabilities when no exemplars are provided. LLaMa3 proves to be the stronger model when provided with sufficient context. At 4-shot, LLaMa3 consistently achieves the highest scores across all metrics. Notably, on the Approach aspect for topic summarization, LLaMa3 (4-shot) reaches an R-1 of 41.84, whereas GPT-3.5 peaks at 25.76.

A significant deviation from our previous experiments is the inverse relationship between the number of in-context examples and model performance. Both models achieve their highest performance in the 0-shot setting across all modalities. The introduction of few-shot examples appears to introduce noise, distracting the models from the specified constraints. While performance recovers while k increases, it never surpasses the zero-shot baseline. There is not evident performance gap between modalities, suggesting that the models are robust to the specific definition of topics. The performance gap between topic-based and entity-based summarization suggests that this dataset heavily relies on entities to convey meaning.

The difference in few-shot performance between the AclSum and NewTS datasets can be attributed to the structural distinctiveness of their data distribution. AclSum is characterized by structural homogeneity, containing only three broad topics with consistent templates, since the documents are taken from the scientific literature. Conversely, NewTS comprises 50 distinct topics with significant intra-topic variance. Even when few-shot examples are drawn from the same topic, the

Setting	R-1	R-2	R-L	BS
topic				
0-shot	33.66	9.85	20.61	61.97
1-shot	31.43	9.48	19.15	61.04
2-shot	32.05	9.77	19.70	61.19
4-shot	32.46	10.50	19.91	61.41
topic words				
0-shot	34.47	10.24	21.02	62.43
1-shot	31.77	9.83	19.49	61.18
2-shot	32.41	9.93	19.87	61.30
4-shot	32.15	9.88	19.93	61.27
topic sentences				
0-shot	34.12	9.84	21.09	62.16
1-shot	32.02	10.02	19.44	61.25
2-shot	32.22	10.02	19.85	61.28
4-shot	32.40	10.14	20.08	61.44
entities				
0-shot	41.32	16.22	26.43	65.83
1-shot	39.66	15.08	24.54	64.89
2-shot	39.99	15.68	25.28	65.08
4-shot	39.89	15.65	25.22	65.05

Table 3.19: GPT-3.5 results on NewTS.

Setting	R-1	R-2	R-L	BS
topic				
0-shot	34.94	10.81	21.67	62.16
1-shot	30.80	9.16	18.74	60.08
2-shot	31.13	9.33	19.29	60.32
4-shot	32.23	10.18	20.72	61.23
topic words				
0-shot	35.92	11.08	22.02	62.28
1-shot	31.87	9.98	19.52	60.81
2-shot	32.73	10.26	20.28	61.17
4-shot	32.32	10.15	20.72	61.25
topic sentences				
0-shot	34.27	10.38	20.80	61.50
1-shot	31.75	10.36	19.69	60.93
2-shot	32.82	10.52	20.38	61.17
4-shot	32.09	10.17	20.64	61.16
entities				
0-shot	42.67	17.00	26.73	65.81
1-shot	39.98	15.80	24.79	64.69
2-shot	40.06	15.83	25.37	64.94
4-shot	39.38	15.15	24.65	64.59

Table 3.20: LLaMa3 results on NewTS.

content and structure of the summary can be widely different between samples. Consequently, providing in-context examples in NewTS likely introduces noise. The model overfits to the structure of the few-shot examples rather than attending the unique semantic constraints of the input instructions. This suggests that few-shot learning is beneficial for standardized tasks and zero-shot inference is more robust for tasks with high structural heterogeneity.

On a side note, in Table 3.21 we report entity inclusion percentages computed through exact match on the two datasets. Similarly to the trends observed in Table 3.18, Table 3.19, and Table 3.20, we observe significant improvements in AclSum while recording decreased performances in NewTS, corroborating the hypothesis that the NewTS dataset contains a large amount of noise.

3.5.4 Conclusions

In this section, we evaluated the capabilities of LLMs in aspect-based summarization. Specifically, we compared proprietary large-scale models against locally deployed alternatives. Regarding our first research question, our analysis reveals that the optimal prompting strategy is heavily dependent on the structural homogeneity of the data. In standardized domains, like AclSum, few-shot learning serves as a powerful alignment mechanism, providing the model with consistent

Setting	Included Entities (%)	
	NewTS	AclSum
GPT-3.5		
zero-shot	94.26	82.15
1-shot	91.98	86.40
2-shot	91.36	89.75
4-shot	88.54	93.10
LLaMa3		
zero-shot	92.27	78.50
1-shot	87.24	84.12
2-shot	82.63	88.90
4-shot	76.92	91.45

Table 3.21: Aspect-based summarization metrics comparing NewTS and AclSum.

templates. Conversely when dealing with high variance domains, datasets like NewTS introduce semantic noise, reducing the effectiveness of few-shot learning. In this case, zero-shot techniques prove the be more robust, as the model avoids overfitting to unrelated structures. Regarding our second research question, results suggest that recent open-source models, like LLaMa3, are viable and often superior to GPT-3.5 The reported experiments highlight a trade-off between robustness and adaptability. Specifically, the proprietary solution exhibits greater robustness, making it preferable when in-context examples are not available.

Smaller, locally deployed, models can effectively replace larger commercial models for specialized, well-defined tasks. However, their deployment strategy must be tailored to the data. They excel as few-shot learners in structured tasks but require careful prompt design to avoid noise in unstructured domains.

3.6 Conclusions

In this chapter, we systematically investigated the capabilities and limitations of Large Language Models and Multi-modal Large Language Models across three complex data-to-text and text-to-text generation tasks: Table-to-Text generation, Chart-to-Table extraction, and Single/Aspect-Based Text Summarization. By analyzing these models across diverse data modalities, several unifying insights have emerged regarding model scale, adaptation strategies, and real-world industrial applicability.

First, our evaluations consistently demonstrate that large proprietary models establish a robust upper bound for zero-shot performance and semantic reasoning. Whether extracting underlying data from complex visual charts or generating

factually consistent abstractive summaries, these models exhibit superior spatial, numerical, and semantic fidelity out-of-the-box. Conversely, smaller open-source models initially struggle with zero-shot hallucinations and structural omissions. However, we proved that this performance gap is not insurmountable. Through task-specific fine-tuning or optimized in-context learning, open-source architectures can achieve highly competitive, and sometimes state-of-the-art, results on targeted distributions.

Second, the efficacy of adaptation techniques—namely fine-tuning and prompt engineering—is heavily dependent on the structural homogeneity of the target data. In highly standardized domains, such as scientific table descriptions or homogeneous aspect-based summarization, few-shot prompting and PEFT serve as powerful alignment mechanisms. However, when applied to domains with high structural variance and noise, few-shot examples often introduce semantic noise, causing models to overfit to irrelevant structural templates. In these high-variance scenarios, zero-shot inference with strict, instruction-based constraints proves to be significantly more robust.

Finally, our empirical findings highlight a critical discrepancy between curated benchmark performance and real-world deployment readiness. While modern MLLMs and LLMs excel on standardized datasets, their application to complex, noisy industrial environments, such as extracting data from multi-axis ESG charts or summarizing massive, unstructured tables—exposes persistent vulnerabilities. Models continue to suffer from data translation errors, an inability to accurately infer continuous values from sparse visual labels, and a tendency to omit granular numerical details in favor of stylistic mimicry. Furthermore, the high computational overhead and hardware requirements associated with locally deploying and fine-tuning open-source models often offset their theoretical cost advantages over proprietary APIs.

Ultimately, this chapter demonstrates that while generative AI provides powerful, general-purpose engines for complex information extraction and summarization, there is no universal ”one-size-fits-all” solution. The successful integration of these technologies into reliable, automated business intelligence pipelines requires a careful orchestration of model scale, data-aware prompting strategies, and rigorous human validation to mitigate the inherent risks of modality hallucinations and factual inconsistencies.

Chapter 4

Multi-Modal Summarization with Multi-Modal Outputs

Text summarization has evolved significantly, first through Transformer-based models like BART [109] and T5 [175], and then with autoregressive Large Language Model like GPT [1, 28]. Yet, the task is still bounded to text, while documents have evolved to be substantially multi-modal. *Multi-Modal Summarization* is an emerging task involving the synthesis of information from diverse modalities such as text, images, tables, audio, and video [93, 119]. While traditional summarization works by processing and generating text [157],multi-modal summarization integrates heterogeneous data that convey information through different means. The integration of fundamentally different modalities requires sophisticated alignment techniques to preserve the input's semantic coherence while exploiting the complementarity of each modality [209]. Multi-modal summarization inherits the issues of classical summarization like factual consistency [141], controllability [80] and evaluation [61].

The evaluation issue is particularly exacerbated by the presence of multi-modal inputs, due to the lack of methodologies accounting for heterogeneous input modalities. Existing metrics, as the ones previously mentioned in this manuscript, are text-based and fail to assess multi-modal quality adequately. Overlap-based metrics primarily assess lexical overlap and semantic-similarity metrics are exclusively tuned on textual inputs. Recently proposed LLM-based metrics like G-Eval [125] fail to consider visual grounding and image-text complementarity, both core

dimensions in multi-modal summarization. Additionally, the field lacks methodologies for measuring the informativeness and relevance of non-textual components in generated summaries. Finally, the scarcity of annotated multi-modal datasets limits both supervised training and robust evaluation.

Another persistent issue is modality imbalance, e.g. the tendency of models to over-rely on textual inputs while under-utilizing visual and other non-textual cues [258]. For instance, models for news summarization often neglect key images [258], resulting in summaries that miss salient visual information, or they might rely solely on the text modality. This is exacerbated by the computational constraints that prevent simultaneous processing of long texts and multiple images [252]. Finally, modality coherence remains difficult to maintain, especially when the narrative interleaves textual and visual content.

In real-world applications, users increasingly interact with complex financial documents that embed multi-modal content in tables, images, and charts. Users require concise yet rich summaries narrating a context derived from textual paragraphs as well as rich images and tables. Moreover, research suggests that enriching summaries with multiple modalities can increase user satisfaction by up to 12.4% [256]. In this context, Multi-Modal Summarization with Multi-Modal Outputs (MSMO) [256] is particularly demanding as it requires retrieving key knowledge from heterogeneous inputs while producing coherent summaries.

Through this work, we present a modular pipeline for MSMO, designed for document-level inputs. Our system exploit accurate layout detection, links textual and visual elements, and produces factually consistent summaries integrating charts and tables in the final output. We further investigate how structured inputs influence LLM-based summarization, how it affects the quality of LLM-based evaluation, and how well LLMs manage the inclusion of multi-modal content in generated summaries.

4.1 Literature Review

The Multi-Modal Summarization task consists in the generation of concise and meaningful summaries from multiple modalities such as text, images and videos. Information alignment from different modalities usually occurs in a shared embedding space. Multi-modal enhancements are distinguished between *supplementary enhancements*, that reinforce the facts presented in the central modality, and *complementary enhancements*, that complete the information from the central modality [93]. Efforts in multi-modal summarization have been made for times series [3],

movies [60], asynchronous multimedia sources [257], charts [176], and visual summaries and pictorial narratives [20, 218]. Recently, Vision Language Models (VLMs) have been explored for healthcare-specific applications [73] and images and videos [119, 214].

Oldest approaches for multi-modal summarization include both deep neural networks and rule based strategies. ILP was used primarily for extractive summarization [7, 66] but a Joint Integer Linear Programming framework was proposed to extract necessary sentences, images, and videos from news datasets using clustering of pre-trained joint embeddings [92]. Submodular functions' properties were used to solve the multi-modal summarization task. [207] used coverage, novelty, and significance as the submodular functions to extract the most relevant documents for timeline generation in social media event. [114] used a linear combination of submodular functions under budget constraints to create extractive summaries composed of sentences, images, videos, and audio. [148] used a weighted sum of submodular functions to generate summaries comprising texts and images. Finally, graph based techniques have been adopted in extractive summarization frameworks. [148] used a graph based approach to generate text-image summaries and [114] used a guided LexRank approach to identify text saliency in multi-modal inputs. However, all these approaches are extractive in nature.

Summarization of multi-modal documents [258] and generation of multi-modal outputs [256] are limited in the scientific literature, leaving a gap dictated by data scarcity and excessive annotation costs associated with large-scale dataset construction. Instead research effort has been known to concentrate on QA tasks for visually rich documents and form understanding utilizing Transformer-based models like LayoutLMv3 [89] and VL-T5 [41]. Recently, MLLMs have been advancing to tackle the challenged related to document parsing [25, 99, 133, 225], equipped with the ability to parse document pages directly into structured, machine readable formats. Unfortunately, the multi-modal summarization challenge is not explicitly treated with this models.

4.2 Proposed Framework

Our approach to MSMO is a modular pipeline constituted by two parts: (i) a Document Pre-Processing step and (ii) a Controllable Summarization step.

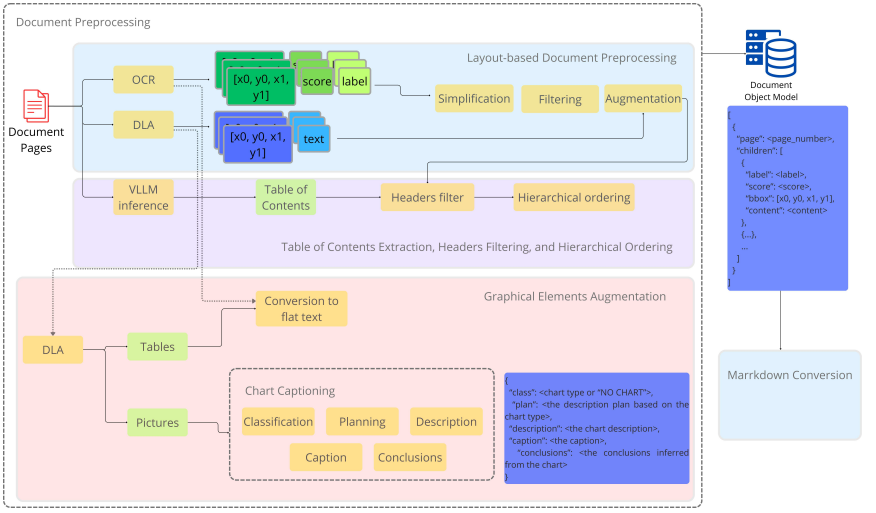


Figure 4.1: MSMO document pre-processing pipeline.

4.2.1 Document Pre-Processing

The goal of our document pre-processing step, shown in Figure 5.1, is bipartite. First, we face the necessity to map every document element to a format consistent with the formats processed by Altilia's technology. Due to the recently deployed heavy text-related optimizations, the chosen form is text. The first goal of the pre-processing step is the transformation of a document into what is referred to as a Document Object Model (DOM). As for HTML files, the DOM is a tree-like hierarchical structure where nodes correspond to document parts and contain comprehensive information about the content of the corresponding document component. The DOM can be represented as a dictionary and can chunked and stored in NoSQL databases for querying. Our pre-processing pipeline works in four steps: (i) Layout-aware Document Pre-Processing, (ii) Table of Contents (TOC) Extraction, Headers Filtering, and Hierarchical Ordering, (iii) Graphical Elements Augmentation, and (iv) Markdown Conversion.

Layout-based Document Pre-Processing

This critical step focuses on the identification and interpretation of spatial and structural components of the document page. Structural components include titles, heading, paragraphs, images, tables, bullet points, and formulas. In the document summarization context, layout understanding enables more accurate semantic segmentation, facilitating the identification of key content and enabling the correct reading order. To detect layout elements, we utilize a Deformable-DETR [259] model, fine-tuned on DocLayNet [166]. This model processed rasterized versions of document pages, locating and classifying the layout elements. After detection, the located elements undergo three processing steps: Simplification, Filtering, and Augmentation.

Simplification. This step is standardizes elements that do not require special handling and those that might introduce noise. For instance the *formula* label is generally irrelevant for summarization algorithms and the frequent misclassifications as text introduce noise that might interfere with downstream pipeline steps.

Filtering. DLA systems are often based on classical object detection and rely on region proposal algorithms to obtain candidate regions, resulting in repeated and overlapping predictions. To eliminate redundancies and noise, a score-based filtering algorithm is employed. The algorithm sorts detected elements by decreasing score and removes those overlapping with higher-rank objects as described in 4.2.1.

Augmentation. To make layout elements more informative, we associate them with their textual content, obtainer through pdfplumber[1] OCR engine. The extracted text is aligned to text boxes measuring their overlap. Finally, each layout element is associated with a reading order consistent with the one obtained through the OCR algorithm. Specifically, the layout elements reading order is consistent with the first associated text box.

Table of Contents Extraction, Headers Filtering, and Hierarchical Ordering.

The DLA model often introduces errors in the DOM representation, often due to misclassifications of regular bodies of text and image captions. To address this

[1] https://pypi.org/project/pdfplumber/

Algorithm 1 Layout Filtering

function $\Theta(b_1, b_2)$
    return Ratio of the overlapping area over the area of b_2
end function
$B \leftarrow \text{DLA}(I)$ ▷ $B = \{(b_i, s_i) \mid b_i \in \mathbb{R}^4,\ s_i \in [0, 1]\}$
$N \leftarrow |B|$
Reorder B into $B' = \{(b_1, s_1), (b_2, s_2), ..., (b_N, s_N)\}$ such that $s_1 \leq s_2 \leq ... \leq s_N$
Initialize $L \leftarrow \emptyset$
$\tau \leftarrow c$ ▷ $\tau \in [0, 1]$
for $i = 1$ **to** N **do**
    $O \leftarrow \emptyset$
    for all $(b, s) \in L$ **do**
        $a \leftarrow \Theta(b_i, b)$
        if $a \geq \tau$ **then**
            $L \leftarrow L \cup \{(b, s)\}$
        end if
    end for
    if $L = \emptyset$ **then**
        $L \leftarrow L \cup \{(b_i, s_i)\}$
    end if
end for
return L

issue, and reduce structural misalignment in the final markdown representation, we leverage GPT-4o to generate the document's TOC from the first k pages of the document, using the prompt template in Figure 4.2. The table of contents serves as a reference, utilized to filter headers from the document pages using an exact match criterion. This cross-referencing step significantly improves fidelity in the document structure.

Graphical Elements Augmentation

The considered documents are lengthy and filled with multi-modal elements. However, LLMs and MLLMs are optimized for lengthy text, not numerous images. This step's goal is to enhance the document's textual representation with rich textual descriptions from visual elements. This addresses two key challenges. First, *Chart Captions Misalignment* is often found in parsed document pages, with captions classified as either paragraphs or headers, or associated with the wrong image. Second, *Uninformative Captions* are often present in document images, as they are mere complements to the graphical information. However, this prevents machine understanding when dealing with uni-modal inputs. To overcome these

Prompt Template

```
System: You are an expert in document analysis. Your task is
to extract the table of contents from a document.
The input consists of the pages of a document, and the
expected output is the document's table of contents.

Do the following:
1. Identify the table of contents in the document.
2. Report the table of contents in its hierarchical format.

Here is an example of the output format.
### TABLE OF CONTENTS
1. Title
1.1 Subtitle
1.2 Subtitle
1.2.1 Sub-subtitle
...

User:
<IMAGES>
```

Figure 4.2: Prompt template for table of contents extraction.

limitations, we utilize GPT-4o to generate descriptions from document charts and images. Resource allocation and project milestones defined by the industrial partner precluded a detailed analysis of this design choice. However, we can report satisfactory descriptions in general. Since LLMs are known to correctly interpret structured tables [158], tables are simply flattened.

DOM Construction

The processed layout elements are concatenated into a Document Object Model. The construction of the DOM is pivotal for the structured representation of the document into machine readable format. This is represented in a tree-like structure with the root representing the document and child nodes representing logical sections and structural components. The DOM is driven by the previously established hierarchical ordering. Each tree node contains metadata corresponding to the element type, the spatial coordinates, and the page number. Nodes classified as text are stored as OCR strings while graphical elements store both the reference to the original image and the rich description generated during the graphical elements augmentation stage. The DOM uses a nested logic where some nodes are parents

to others, following the inferred logical structure of the document. If the table of contents is found in the document, it dictates the document hierarchy, otherwise a heuristic algorithm assigns elements to headers when they are located in the bottom right of a header.

The DOM represents a compatible structure with Altilia's downstream service, since it allows querying, retrieving, and modality focused retrieving (e.g. collecting a certain section or elements from a certain modality) instead of feeding the model with the entire document. Finally, the DOM serves as direct input for the markdown conversion, ensuring that the structured text fed to the model preserves the original document's structural integrity.

To create the markdown representation, headers are concatenated with multiple # corresponding to their logical level in the document; for instance # represents a title while ## represents a subtitle. To account for inputs multi-modality, tables and charts are inserted in the markdown DOM, enclosed in special token sequences.

4.2.2 Controllable Summarization

Controllable summarization is achieved by letting the user writing instructions for a distinct input. A generic prompt template, utilized for all documents, is unlikely to achieve effective controllable summarization. For this reason, we define document categories and associate them with custom templates to extract relevant information from the source document and structure them into a fluent and coherent summary. The inclusion of multi-modal information is not enforced directly but outsourced to the model which is fed with chart captions and flattened tables inserted in the document markdown. The creation of the final summary goes through three steps: (i) Document Classification, (ii) Controllable Generation, and (iii) Multi-Modal Augmentation.

Document Classification

A desiderata for this approach is the ease of adaptation to new domains and documents. For this reason, the classification module is instantiated with a LLM. The model is fed with the markdown of the first few pages of the document and a predefined list of document categories and descriptions and it is expected to output the correct document class.

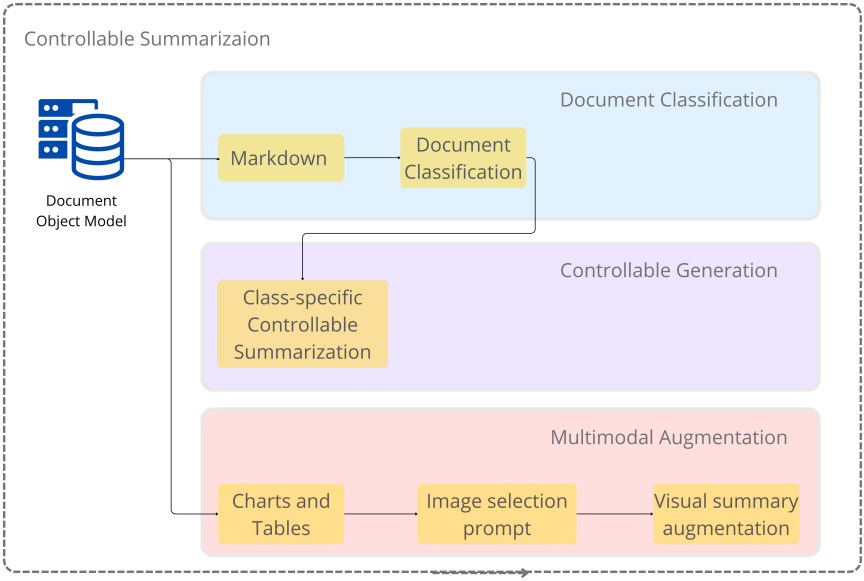


Figure 4.3: MSMO summarization pipeline.

Controllable Generation

This step requires the model to accurately interpret the user intent and adhere to semantic and structural constraints. This is particularly critical for summarization of financial document, due to their length and semantically dense content. To address these issues we leverage a LLM in conjunction with CoT class-specific prompt templates tailored to each document class. The prompt templates specify (i) the markdown structure, (ii) the generation instructions, and (iii) the output format. The *markdown structure* informs the model of the semantic conventions used in the input document. For example, section headers and sub-headers are interpreted using varying numbers of #, while tables and images are represented by designed token patterns. The *generation instructions* provide a step-by-step guide in generating content aligned with user intent. Finally, the *output format example* ensures consistency by illustrating a well-formed output. This prompt design enables fine-grained control over both content and format, ensuring reliable

generations that can be easily post-processed and recomposed in the final summary.

Multi-Modal Augmentation

While the textual portion of the summary is generated abstractively, by rearranging and paraphrasing the information contained in the input document, the visual components require a different strategy due to current architectural and data limitations. Ideally, the multi-modal summarizer should be able to generate both text and visuals. However, two constrains prevent us from utilizing a fully generative approach. First, current MLLMs have made significant steps forward in multi-modal understanding but remain strictly text-centric in multi-modal capabilities and are unable to output pixel-level data directly. Second, while a separate diffusion model could be pipelined to generate images based on text descriptions, this introduces a high risk of visual hallucinations and a high computational burden due to the deployment of a second machine. Moreover, the underlying data plot remains inaccessible.

Since visual elements are not assumed to differ between the source and generated documents, we implement an extractive visual approach. Instead of retrieving images using a bi-encoder setup and computing the similarity between the image and query embedding, we utilize an LLM to identify images relevant to the generated summary sections. This has advantages with respect to the retriever approach. First, an effective retriever utilization would require queries optimized to rank first optimal visual elements, which can't be done at inference time. Second, bi-encoders are functional for obtaining relevant knowledge withing the top-k retrieved elements, leaving us with the issue of identifying the top-1 most relevant element. Finally, a dense retriever achieves high semantic similarity, which might be suboptimal when trying to avoid redundant information.

To overcome this limitations, we employ *LLM-Driven Visual Selection*. This approach leverages LLMs to solve a high-granularity classification task. Each section of the generated summary is fed to model alongside the visual elements descriptions and flattened tables. This approach offers two advantages advantages. First, LLMs can align images to specific summary paragraphs with high granularity. Second, the model can evaluate the images content against text, ensuring that the selected visual elements actively support the summary rather than sharing similar content. In practice, the model is fed with the generated summary and a list of visual elements represented by an ID and their content description. Then the model is tasked with the alignment of a list of visual elements with the summary sections as exemplified in Figure 4.4.

Prompt Template

System: You are an image selection agent. Your task is to select images and tables for a document. The inputs are the summary of a document and a list of images and tables with their descriptions. The expected output is the list of selected items.
Each image and table is identified by an ID and a description.
The summary contains the following elements:
- Headings. These are delimited by "**"
- Paragraphs. Each heading is follwed by one or more paragraphs.

Do the following:
1. Carefully read the summary.
2. Identify and list the section headings from the summary.
3. Carefully read the image descriptions.
4. Identify the most relevant images for the summary. You may select up to {m} images for the summary. For each selected image, explain the selection criterion used.
5. Place each image in the appropriate section.

Below is an example of the output format:
Section Titles
- section 1
- section 2
...

Selected Images
- Image ID: <IMAGE ID>
- Selection Criterion: The criterion used to select the image.
- Target Section Title: The title of the section where the selected image should be placed.
- Image ID: <IMAGE ID>
- Selection Criterion: The criterion used to select the image.
- Target Section Title: The title of the section where the selected image should be placed.

User: Here is the summary (delimited by ```): ```{summary}```
Here is the list of images: {images_list}

Figure 4.4: Prompt template for image selection.

4.2.3 Technical Implementation

The proposed framework was operationalized as a Python-based *skill*. In Altilia, a skill is a function that utilizes various components of the AltiliaIntelligentAutomation platform in a deterministic manner. Since the multimodal summarization of long financial documents for this case study did not impose strict constraints on Time to First Token (TTFT), the architectural design prioritizes modularity, processing depth, and asynchronous execution over token streaming. The pipeline relies on OpenAI's proprietary API (GPT-4o) to handle the generative burden, and triton deployed models for OCR and DLA.

The following sequence represents the execution flow of the implemented system:

- **Model Deployment and Ingestion.** The DLA model is deployed and served via the Triton inference server, which enables scalable inference. Upon document ingestion, the skill routes rasterized document images to the model to extract spacial coordinates and structural labels and bounding boxes.
- **Parallel Extraction and DOM Construction.** Following layout detection, and depending on the availability of OCR in the input document, a native PDF parser or the Azure OCR service extracts the text from each rasterized document page. To minimize latency, the pipeline utilizes asynchronous computing. While the DOM is being constructed, the skill dispatches asynchronous requests to the MLLM the generate rich textual representation for all extracted charts and images.
- **Summary Generation and Visual Alignment.** Once the DOM is fully compiled and all multi-modal elements are captioned, the document markdown representation is injected into the MLLM to synthesize the primary summary. This is then programmatically decomposed into sections and a second API call is executed to perform LLM-driven visual selection, associating relevant visual elements to the generated sections.
- **Section Decomposition and Attribution.** In the final phase, each logical section is fed to the MLLM to associate sentences from each section to passages from the source documents, ensuring strict factual traceability.

4.3 Case Study

The system described in the previous section was proposed in the context of a project requested in a real-world operational environment. This case study was developed in collaboration with an undisclosed client of Altilia, representing an important financial institution active in the Italian market. The objective was the automation of summarization of long financial document, transitioning from manual analysis to an automated multi-modal pipeline. For this project, the client needed to process documents that vary significantly in their structural composition. The primary challenge was not the generation of simple summaries, rather the generation of summaries complemented with multi-modal outputs. Moreover, the system was required to synthesize narrative summaries aggregating information from text as well as tabular and chart data.

This project is characterized by an exacerbation of constraints typical of Altilia's clients. Specifically, the entire work is structured around limited data availability and the strict requirement of factual correctness across all modalities. Given the scarcity of data, fine-tuning smaller language models was deemed unfeasible. Furthermore, the system's intermittent usage pattern rendered a dedicated deployment cost-inefficient. Consequently, leveraging the non-sensitive nature of the input documents, we utilized large-scale proprietary models to maximize performance without the overhead of custom infrastructure.

In this context, it was possible for us to experiment with structured inputs configurations for the generation of structured summaries from long financial documents. The next sections provide a detailed description of our experimental settings.

4.3.1 Experimental Settings

First present the client's dataset, followed by a description of the structured input configurations. Next, we outline the evaluation metrics employed in our analysis. Finally, we discuss the results and their implications.

Dataset

We utilize a private dataset consisting of 42 documents. The documents were collected from three major credit rating agencies, Fitch, Moodys, and S&P, and include rating reports that assess the financial health and creditworthiness of various organizations and industries. The dataset is multilingual and the documents

		source document		summary		compression ratios	
Source	**# docs**	**tokens**	**visual elements**	**tokens**	**visual elements**	**tokens**	**visual elements**
S&P	15	2516.33	36.07	526.27	1.60	0.41	0.06
Moodys	15	5422.33	78.40	529.47	1.07	0.15	0.02
Fitch	12	24479.93	108.87	575.07	1.00	0.04	0.02
Overall	42	10806.2	74.44	543.6	1.22	0.20	0.03

Table 4.1: Multimodal Summarization Dataset statistics.

are redacted either in English or Italian. Domain experts from the client provided us with summaries tailored specifically for the designated rating reports. A qualitative screening of the provided synopses identified critical quality issues concerning the inclusion of information from sources other than the indicated documents, with a verifiability rate of 95%. For this reason, these artifacts were only utilized to compare models and humans ability to include visual elements in the final summary. The final corpus statistics are available in Table 4.1, with each column showing the average statistics for number of tokens and visual elements (i.e. tables and charts).

Structured Input Configurations

To obtain the best possible summaries, we explore the effects of three input configurations: Flat Text (FT), Markdown (MD), and Hierarchical Markdown (MDH). The baseline, FT, only uses the text extracted from the PDF using the pymupdf library. In this configuration, the LLM input is constituted by a concatenation of instructions and the raw document text. All markdown variants adopt markdown-style headers using the # symbol. However, only the hierarchical configurations utilize multiple # to indicate the distance from the document root. Each variant includes images, tables, and charts. In MD and MDH documents, the visual elements are inserted according to the document's linear reading order. All other settings are consistent with what described in the previous sections. These variants serve different purposes: MD offers simplicity while MDH captures structural richness.

Evaluation Metrics

There doesn't exist a single evaluation metric for Multi-Modal Summarization with Multi-Modal Outputs. As a consequence, we rely on multiple metrics to

evaluate our summaries: FineSurE [194], G-Eval [125], and VisG-Eval.

FineSurE. This is a LLM-based summarization evaluation protocol. The technique consists in using a LLM to identify misalignment between the input text and the document summary. FineSurE evaluates the generated output on three dimensions: faithfulness, completeness, and conciseness. However, due to the inaccuracies of our ground truth summaries, we are only evaluating the generated synopses on faithfulness with respect to the input document. Each generated output is transformed into key facts and each fact is compared to the source document. Faithfulness is computed as the percentage of correct facts.

G-Eval. This is a GPT-4-based prompting protocol for assessing the quality of generated summaries, with a demonstrated strong correlation to human evaluators' judgments. GEval utilizes AI-generated prompts (Auto CoT) to guide human-like assessments. We adapt the G-Eval protocol to the collected rating reports following the standard G-Eval protocol. We use GPT-4 [1] to create a new set of instructions for the evaluation of the financial summaries generated by our summarization approach. Then, we employ GPT-4o [91] to execute the instructions and generate the 5-points likert scale evaluations with a voting mechanism.

VisG-Eval. We use GPT-4o to assess the effectiveness of image selection in our methodology. Using the G-Eval protocol, we generate two sets of instructions: one for image relevance, which measures how well images align with the summary's content, and one for image novelty, which gauges how much new information the images add. Relevance scores are generated with a CoT prompt while novelty scores are generated with the classical G-Eval protocol, more information in the appendix. We refer to this evaluation approach as VisG-Eval. These metrics indirectly assess the inclusion of supplementary and complementary enhancements. Each image caption is paired with the summary, and GPT-4o evaluates the relevance and novelty on a 5-point Likert scale.

4.3.2 Results

We report the FineSurE evaluation scores in Table 4.2. Since generated summaries are going to include information distilled from multi-modal source, we utilize the MDH documents for reference. Notably, we observe a generally higher factual error rate from the summaries generated from the unstructured documents. In

	S&P			Moodys			Fitch			Overall		
	FT	**MD**	**MDH**	**FT**	**MD**	**MDH**	**FT**	**MD**	**MDH**	**FT**	**MD**	**MDH**
Faithfulness	94.1	96.1	97.3	98.0	99.8	99.3	94.4	98.1	98.3	95.5	98.0*	98.3*

Table 4.2: FineSurE metrics for multiple summarization. The values marked with * are significantly larger than the baseline.

order to assert the validity of the improvements with respect to the baseline, we implement a 0.05 α-level paired t-test. Due to the reduced dataset dimension, we are only able to successfully obtain significant differences in the Overall setting, that uses all 42 documents to compute the significance of the test. The values marked with "*" in Table 4.2 are significantly larger than the FT baseline.

Table 4.3 presents the G-Eval scores for reference-free evaluations of our summarization approaches. For the S&P source, which contains the shortest and simplest documents, the FT approach performs well but is outpaced by the MDH approach. The FT modality outperforms MD in coherence and fluency, likely due to challenges in distinguishing section headers and image captions without a table of contents, leading to inconsistencies in the markdown version. The MDH input modality excels with the structured title format in markdown, achieving the highest performance. On Moodys, the MD approach shows consistent improvements. On Fitch, the MD approach performs better in all dimensions except coherence. Overall, MD and MDH strategies outperform FT, highlighting the advantage of structured inputs in summarization. Again, we are able to use a paired t-test to assert the significance of the differences in the generated summaries using a 0.05 α-level. The test only shows some significance in the Overall column of Table 4.3 due to the greater sample size. Only the Consistency dimension shows a significance level and only in the MDH setting. As with FineSurE, this suggests that summaries generated using the MDH modality exhibit fewer factual errors compared to those produced using flat text alone.

Finally, Table 4.4 contains the evaluation scores obtained through VisG-Eval, which assesses the quality of the multi-modal elements inserted into the summaries in terms of Relevance and Novelty. Overall, the proposed approach demonstrates superior capabilities in selecting and generating multi-modal content compared to the original summaries. Specifically, the generated summaries achieve higher Relevance scores across the aggregated datasets, indicating that the model is effective at identifying tables and charts that are semantically aligned with the narrative of

the generated summary. As far as Novelty is concerned, the approach consistently minimizes information redundancy. The generated summaries achieve an overall scores of 2.91 which surpasses the 2.73 obtained by ground truth summaries. This suggests that the model successfully selects elements that contribute to the narrative by adding new information rather than repeating data already present in the summary text. Collectively, the scores validate the approach effectiveness in synthesizing multi-modal financial summaries.

4.3.3 VisG-Eval - Correlation with Human Feedback

Two annotators were tasked with the evaluation of the images attached to the summaries by the system. Specifically, they were given the same instructions provided to VisG-Eval, e.g. to assign a likert score for both the relevance and novelty dimensions to each image. We compute the correlation between annotator feedback with Spaerman (ρ) correlation and Kendal τ The correlations between human annotators are high for both relevance and novelty, with $\rho > 0.6$ and $\tau > 0.5$ in both cases. Since the annotators are deemed reliable, we average their scores and compute correlations with LLM judgments with and without CoT reasoning for the generation of the likert score. The results, shows in Table 4.5, highlight that the reasoning approach correlated better for relevance and the simple approach correlates better for novelty.

4.3.4 The Effect of further Cleaning the Document Markdown

From now on, we perform experiments exclusively on 9 documents from Fitch since they are the only ones actually containing a table of contents that we can use to perform headers filtering. The next experiments are supposed to provide further insights into the usage of properly formatted markdown documents in

	S&P			Moodys			Fitch			Overall		
	FT	**MD**	**MDH**	**FT**	**MD**	**MDH**	**FT**	**MD**	**MDH**	**FT**	**MD**	**MDH**
Coherence	3.74	3.7	3.79	3.5	3.53	3.46	2.77	2.8	3.09	3.36	3.36	3.46
Consistency	3.42	3.49	3.66	3.36	3.46	3.5	2.66	3.2	3.07	3.16	3.39	3.42*
Fluency	1.93	1.87	2.02	1.98	2.22	2.09	2.07	3.29	2.15	1.99	2.16	2.09
Relevance	3.17	3.31	3.4	3.01	3.27	2.95	2.82	3.04	2.82	3.0	3.21	3.06

Table 4.3: G-Eval metrics for multiple summarization settings. The values marked with * are significantly larger than the baseline.

	S&P		Moodys		Fitxh		Overall	
	gen	**gt**	**gen**	**gt**	**gen**	**gt**	**gen**	**gt**
Relevance	3.75	3.01	4.12	4.33	3.87	3.16	3.91	3.5
Novelty	3.07	2.95	2.97	2.96	2.55	2.61	2.91	2.73

Table 4.4: VisG-Eval scores.

summarization settings. For this reason, we are unable to include the remaining documents, that contain nonfilterable noise.

Overall, our approach of transforming the document into its markdown version inserts numerous errors in the final markdown: wrongly positioned headers, misclassified layout elements, and captions mixed with headers and text. In order to test the hypothesis that further cleaning the document improves the LLM document understanding capabilities, we further clean the provided documents starting from the MDH version. First, we utilize layout-based heuristics to remove redundant captions and titles for charts and tables when positioned too close to the corresponding visual elements based on bounding box proximity. This helps reduce duplication or noise that often arises from OCR artifacts or layout quirks. Second, we utilize the reading order information provided by our parsing strategy to change the position of the layout elements description and position them at the end of each section. This prevents disruption in the text flow of the document and allows the evaluation of the model capabilities of introducing information from layout elements with informativeness criteria instead of reading order criteria.

Table 4.6 shows the FineSurE Faithfulness metrics obtained using the original markdown version (first row) and the cleaned markdown version (MDH_{clean}) as reference respectively. The table serves two purposes. First, the metrics highlight that using the corrected version of the document allows for improvements in the Faithfulness of the generated summary. In fact, the Faithfulness score of MDH_{clean} in the first row is a full 100, against the 98.3 previously obtained.

		rel		nov	
		ρ	τ	ρ	τ
IAA		0.612	0.539	0.676	0.578
CHF	CoT	0.452	0.306	0.182	0.142
	- CoT	0.008	0.048	0.324	0.209

Table 4.5: Comparison of G-Eval, VisG-Eval, and structural metrics.

		FT	MD	MDH$_{clean}$
Faithfulness	**MDH**	94.3	98.3	100
	MDH$_{clean}$	95.1	95.0	96.1

Table 4.6: FineSurE metrics for multiple summarization settings with a cleaned markdown version.

Second, the table shows the great limitation of LLM-based evaluations of long document summaries. Specifically, the source document format matters and greatly influences the value of the evaluation metrics. FT improves by more than 10 points while both MD and MDH see their scores reduced by almost 4 points.

Table 4.7 reports the G-Eval scores using MDH and MDH$_{clean}$ strategies. Similarly to what was observed for FineSurE, the input structure affects the scores, as seen in the FT approach, where scores change significantly between the first and second markdown versions. This is expected but undesired. Ranking orders and score gaps vary between methods. The FT method is no longer the most consistent, even though it was the most consistent with the first markdown version. Surprisingly, MD and MDH methods are the most consistent, suggesting that a cleaner markdown allows better document understanding. Fluency shows slight variation, and relevance scores change in both order and values.

4.3.5 The Effect of Structured Inputs on Structure Understanding

To assess the model's ability to incorporate structural information, we extract section headers from documents with a table of contents. Using section-by-section summaries, we evaluate the model's comprehension through precision and recall.

		FT	MD	MDH$_{clean}$
MDH	**Coherence**	3.05	2.92	2.74
	Consistency	2.32	2.42	2.67
	Fluency	1.82	1.64	1.63
	Relevance	2.58	2.75	2.46
MDH$_{clean}$	**Coherence**	3.79	3.85	3.7
	Consistency	3.48	3.9	3.73
	Fluency	1.82	1.59	1.7
	Relevance	3.05	3.29	3.46

Table 4.7: G-Eval metrics for Fitch documents after the creation of a new markdown structure.

	FT	MDH
Precision	0.40	0.67
Recall	0.84	0.77
F1	0.54	0.71

Table 4.8: Structural information evaluation.

We apply a simple regex to match summary titles with those in the table of contents. Inferred titles that are not present in the table of content are classified as false positives while table of contents titles that are not present in the final summary are classified as false negatives. To ensure fair comparison, we remove any text before the first title token, eliminating the table of contents at the start of documents that could introduce bias.

As per Table 4.8, the FT approach generates significantly more false positives due to its lack of awareness about the document structure. As a result, the model frequently misclassified rows as titles based on their position and syntactic characteristics. Additionally, the FT approach exhibits lower recall, a consequence of its tendency to generate an excessive number of titles. Overall, structured input yields higher precision but lower recall. Surprisingly, despite clear instructions and the simplicity of the task, the model fails to correctly identify and report section titles. This is particularly unexpected given that the only document rows beginning with "#" correspond to section headers.

The Effect of Ablating Markdown Components

To assess the contribution of each pipeline component, we perform an ablation study on Fitch documents with tables of contents. This setup allows us to isolate the impact of the TOC, which is not present on other documents, on summary quality. We remove individual modules from the full system (MDH_{clean}) and report the results in Table 4.9. Removing the graphical elements (ge) from the input markdown leads to a notable improvement in fluency (+0.47), but slightly decreases the relevance (-0.13) of the textual summary. Moreover, consistency doesn't vary and the difference in coherence is negligible. The order of visual elements (image order) also only has the stronger effect on fluency (+0.45), due to the fact that the correct images and tables are positioned under the correct paragraph, and weaker effects on coherence, consistency, and relevance. Reintegrating the text close to charts and tables has the greatest effect on fluency (+0.55) and

	GEval				Vis-GEval		$\text{struct}_{sections}$		
	coh	**con**	**flu**	**rel**	**relevance**	**novelty**	**p**	**r**	**f1**
MDH_{clean}	3.7_{-}	3.73_{-}	1.7_{-}	3.46_{-}	3.65_{-}	2.8_{-}	0.5_{-}	0.76_{-}	0.6_{-}
- ge	$3.72_{+.02}$	3.73_{-}	$2.17_{+.47}$	$3.33_{-.13}$	$3.93_{+.28}$	$2.55_{-.25}$	0.5_{-}	$0.82_{+.06}$	$0.62_{+.02}$
- image order	$3.46_{-.24}$	$3.62_{-.11}$	$2.15_{+.45}$	$3.6_{+.14}$	$3.56_{-.09}$	$3.15_{+.35}$	$0.54_{+.04}$	$0.84_{+.08}$	$0.66_{+.06}$
- filtering	$3.6_{-.1}$	$3.52_{-.21}$	$2.25_{+.55}$	$3.4_{-.06}$	$3.43_{-.22}$	$2.53_{-.27}$	$0.52_{+.02}$	$0.84_{+.08}$	$0.64_{+.04}$
- toc	$3.72_{+.02}$	$2.71_{-1.02}$	$1.78_{+.08}$	$3.42_{-.04}$	$3.82_{+.17}$	$2.6_{-.2}$	$0.54_{+.04}$	$0.77_{+.01}$	$0.63_{+.03}$

Table 4.9: Ablation of markdown components.

has minor impacts on the remaining GEval dimensions. Finally, it appears that not filtering the section headers using the TOC has the greatest effect on consistency (-1.02) and minor effects on the remaining GEval dimensions.

On Vis-GEval, we only observe a few differences with ge (+0.28 on relevance and -0.25 on novelty), image order (+0.35 on novelty), filtering (-0.22 on relevance and -0.27 on novelty), and toc (+0.17 on relevance and -0.2 on novelty). Finally, relaxing pipeline components has little to no impact on document structure understanding.

4.4 Conclusions

In this chapter, we presented a modular framework for multimodal summarization with multimodal outputs. The proposed system was tailored for long financial documents rich of tables and charts. We addressed critical challenged related to modality imbalance, alignment, and factual consistency.

Our experimental results demonstrate that the structured representation of the input document is crucial for the quality of the generated summaries. We observed that structured inputs significantly outperform flat text representations. The inclusion of document structure reduced hallucination rates, improving faithfulness of the generated summaries. Both the FineSurE and G-Eval metrics highlighted the effectiveness of the approach in terms of factual accuracy. However, further experiments highlighted the limitations of current evaluation approaches, lacking standardization and being heavily dependent on the input format.

Furthermore, evaluation of MLLMs abilities to correctly generate structured summaries highlight the limited ability to generate summaries comprehensive of all the document sections. Specifically, a MLLM tends to infer too many sections when using only textual inputs and too few when dealing with structured inputs. Ablations show that tampering with the layout elements and headers can improve

the model understanding of the document sections.

Despite these advances, it is crucial to acknowledge the rapid evolution of the document processing and LLM landscape. While our modular pipeline effectively mitigated the context window and resolution limitations of earlier models by treating modalities sequentially, alternative natively multi-modal systems have recently emerged. regarding document parsing, the Qwen3-VL series [4] offers robust solutions for both parsing and chart processing. With improved localization capabilities, these models can parse documents to generate bounding boxes and classes for all layout elements. Conversely, focusing on efficiency, IBM recently released GraniteDocling [154]. This open-source chat LVLM can parse documents and localize objects with fewer than 300M parameters—an order of magnitude smaller than its competitors.

In the LLM landscape, recent proprietary advancements are exemplified by the release of GPT-5.2 and Gemini-3. These models generally offer superior capabilities and enhanced summarization performance. Conversely, open-source development has been prioritizing efficiency. Mixture-of-Experts (MoE) architectures [57, 65] achieve high performance with significantly lower computational costs, while dynamic quantization techniques [53, 231] further reduce this burden by at least half. However, none of these models currently support the effective concatenation of multiple images with extensive bodies of text. Crucially, there is no existing approach capable of simultaneously processing high-quality images and text from long documents.

Consequently, future research should address two critical limitations. The first is mitigating the performance degradation observed when models process multiple image and text inputs concurrently. The second is the development of robust datasets and evaluation metrics designed specifically to assess multimodal summarization tasks that generate multimodal outputs.

Chapter 5

Robust Evaluation Strategies for RAG.

In this chapter, we address a critical bottleneck in the deployment of advanced RAG systems, i.e. their reliable assessment. In the context of this manuscript, Retrieval Augmented Generation systems represent the key mechanism to ground the summary knowledge into verifiable evidence. However, this systems' utility is limited by their ability to accurately evaluate their performance, especially in real world scenarios where ground truth data is often unavailable. In this chapter, we primarily focus on the effectiveness of LLM-derived metrics by evaluating how well automated metrics approximate human perception of answer relevance and correctness.

To achieve this, we present a four-steps framework: (i) *ingestion* converts the documents into semantic embeddings for efficient similarity search, (ii) *retrieval* identifies the most relevant text chunks based on a user query, (iii) *generation* uses in-context learning to generate an answer through LLMs, and (iv) *evaluation* uses both reference-based and reference-free metrics. Through extensive experiments and annotations on a proprietary and a public dataset, we investigate the reliability of metrics such as BEM and RAGAS. Then, through the analysis of spearman correlation coefficients against human evaluation, we demonstrate that ground truth metrics show moderately strong alignment with human judgment, while reference-free metrics face significant challenges in capturing the nuances of answer quality.

5.1 Literature Review

RAG

Retrieval Augmented Generation (RAG) is an architectural paradigm that combines *retrieval* and LLMs [110, 206]. Large Language Models knowledge is limited by their training paradigm, making them knowledgeable about whatever was utilized for pre-training and fine-tuning. Due to the inherent time discreteness of LLM training data, models knowledge has time-dependent cutoffs. Moreover, due to the generic nature of such data, LLMs are prone to hallucinations. RAG solves these issues by sourcing relevant passages from an external corpus to provide up-to-date grounded information. Early RAG implementations used to focus on Open-Domain Question Answering (OpenQA) by jointly pre-training the retriever and the generator [79].

The core of RAG architectures is divided in two approaches. The first lets the generator attend each retrieved document independently and then marginalizes over the generated outputs. The second lets the generator attend the concatenated sequence of documents, letting the model concurrently draw knowledge from all documents [191]. Recently, the shifting research paradigm resulted in the development in advanced RAG techniques. Multi-hop RAG requires multi-hop retrieving and reasoning [206], Adaptive RAG seeks to increase efficiency by determining if retrieval is required for a given query or user query can be answered from the model internal states [250], and Retrieval-Augmented Fine-Tuning aims at ignoring irrelevant retrieved documents while providing responses based on the relevant ones [245].

RAG Evaluation Systems

Traditionally, QA and RAG systems were evaluated using n-gram overlap metrics like Exact Match and Token F1 [177]. Unfortunately, metrics that rely on n-grams often fail to recognize equivalent answers using different wordings, lack semantic context, and mostly rely on annotated datasets [29]. To overcome these limitations, researchers are developing reference-free evaluation frameworks [68]. RAGAS [59] uses LLMs to evaluate answers of three dimensions identifying issues with grounding and relevance to the question and context. ARES [182] utilizes synthetic datasets to train small LLMs as judges tailored to specific domains and uses Prediction-Powered Inference (PPI) to correct model errors and provide statistical confidence intervals for its rankings. BEM [29] uses a BERT-based classifier trained on human judgments to measure answer equivalence. Finally, some

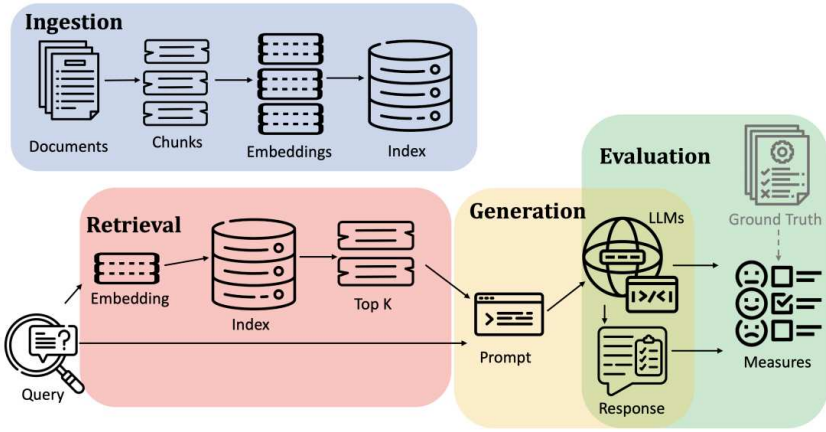


Figure 5.1: A simplified graphical representation of the implemented RAG system.

approaches prefer to decompose the generated answer into factual claims and use them as retrieval queries to verify them against the source corpus [90, 98].

The evolution of Retrieval Augmented Generation (RAG) has led to the development of a variety of implementations that integrate document retrieval and Large Language Models. Early research focused on the foundational aspect of feeding LLMs with retrieved passages to reduce hallucinations in conversational systems. However, more recent research have introduced complex strategies for ingestion, including chunking strategies and semantic embedding techniques to optimize how information is indexed and retrieved.

5.2 Experimental Settings

In order to correctly validate RAG evaluation metrics, we define a standardized RAG pipeline and select datasets and metrics.

5.2.1 RAG Pipeline

We designed and implemented an end-to-end RAG pipeline. Specifically, our architecture is composed of three sequential phases: (i) Ingestion and Indexing, (ii) Semantic Retrieval, and (iii) In-Context Learning.

Ingestion and Indexing

The first step consists in the creation of manageable chunks. This is achieved by segmenting the documents into smaller parts of 1024 characters. Due to the mostly textual part of the considered datasets, no document understanding modules are implemented. Instead, documents are processed through OCR. Then, raw text is chunked into passages of 1024 characters with a stride of 128. Since the evaluation of bi-encoders is not relevant for our work, we rely on OpenAI's *text-embedding-ada-002*[1] to embed textual chunks and queries and we utilize Milvus[2] to store the generated embeddings.

Sematic Retrieval

Given a query, the model used to embed chunks was also used to embed the query. The query vector is used to execute similarity search into the embedding database. Top-k is fixed to 10 and cosine similarity is used to search for the best chunks to answer the query.

Answer-Generation

In order to answer the question, GPT-4 is instanced as the reader. The LLM is fed with the concatenated retrieved chunks and instructed to answer the provided query. In order to account for unanswerable questions, the model is given the instruction to explicitly state whether is the provided material is not sufficient to answer the query. The utilized prompt template is available in Figure 5.2.

5.2.2 Datasets

To assess LLM-based evaluation strategies, we employ two datasets from the narrative and financial domains. NarrativeQA [100] contains 1102 English documents divided into books and movie scripts, associated with 32k question-answer pairs.

[1] https://openai.com/blog/new-and-improved-embedding-model
[2] https://milvus.io/it

Prompt Template

System: You are a chat-bot having a conversation with a human. Given the following extracted parts of a long document and a question, create a final answer.
If you don't know the answer, just say that you don't know, don't try to make up an answer.

Context: {CONTEXT}

Chat History: {CHAT_HISTORY}

User: {user_query}

Figure 5.2: The prompt template utilized to generate answers with GPT-4.

From this dataset, we randomly sample 50 book-related and 50 movie-related questions, spanning 41 unique books and 42 unique movie scripts. The dimension of the sample was restricted to account for budget constraints associated with project. From the financial side, we utilize a private dataset (FinAM) containing 50 Italian question-answer pairs. The corpus covers Italian asset management documents on topics including investment strategies, risk management, and regulatory compliance. The question are complex, often requiring information from multiple paragraph containing detailed conversational-style answers.

5.2.3 Evaluation Metrics

We evaluate two dimensions of the generated answers: answer relevance and answer correctness. *Answer relevance* evaluates whether the generated answer is relevant with respect to a given query and retrieved passage. *Answer correctness* measures an answer's factual correctness with respect to the retrieved passages.

To evaluate the quality of the generated answers we utilize three recently proposed LLM-based frameworks: TruLens, RAGAS [59], and BEM [29]. *TruLens* is an open-source library used to evaluate, track, and debug LLM-based applications. This operates on LLM-based feedback functions to measure context relevance, groundedness, and answer relevance. *RAGAS* focuses on semantic accuracy using "LLM-as-a-judge" to score a retriever's ability to retrieve data and generate faithful, relevant answers. Both TruLens and RAGAS use a Large Language Model to evaluate wether the generated answer is correct and relevant with respect to the

ground truth answer and user query.

RAGAS measures answer correctness through the extraction of factual statements from both the predicted and gold answers. Then, statements are flagged as true positives if present in both lists, false positives if only available in the generated answer, and false negatives if only present in the ground truth answer. Answer correctness is presented in the form of an F1 score. Answer relevance, on the other hand, is computed using the LLM to generate multiple synthetic questions to the generated answer and then computing the average cosine similarity between the embedded original query and generated queries. TruLens answer relevance is measured directly through the model, assigning a continuous score in $[0, 1]$.

Finally, the BEM score [29] is a semantic metrics based on a fine-tuned BERT model rather than an LLM. BEM is optimized for answer equivalence and is used to evaluate whether the generated answer is equivalent to the ground truth.

5.2.4 Correlation with Human Judgment

To study the correlation with human judgment, 4 human annotators were tasked with the evaluation of relevance and correctness of the generated answers. Annotators were provided the query and the generated and ground truth answers and were asked to assess the generated answers with a likert score with 1 being equivalent to incorrect or irrelevant answers and 5 corresponding to completely correct and factually relevant answers. Upon completion, the scores were collected and compared. In case of discrepancies, annotators were asked to discuss on the provided answer and reach consensus, alleviating the individual bias and increasing the reliability of the maual evaluation.

5.3 Experimental Results

We report the correlations between human and model-based evaluations in Table 7.6. About answer relevance (AR), TruLens achieves the most significant results with an average correlation of more than 40% with GPT-3.5 and greater than 30% on GPT-4. Overall, GPT-3.5 results are superior on NarrativeQA for both subsamples. Interestingly, the prompt template from both TruLens and RAGAS are optimized for GPT-3.5. On the Italian dataset, where templates were adapted to the Italian language and not merely translated, GPT-4 achieves superior results on both frameworks. Even for answer correctness (AC), GPT-3.5 achieves slightly superior results on both the publicly available datasets while GPT-4 achieves signi-

Metric	Correlation with Human Judgment			
	NarQA$_{books}$	**NarQA$_{movies}$**	**FinAM**	**Avg**
GPT-3.5				
AR TruLens	0.436	0.565	0.178	0.423
AR RAGAS [59]	0.234	0.483	0.153	0.323
AC RAGAS [59]	0.718	**0.792**	0.053	0.536
GPT-4				
AR TruLens	0.420	0.213	0.280	0.314
AR RAGAS [59]	0.150	0.411	0.230	0.287
AC RAGAS [59]	0.670	0.781	**0.531**	**0.653**
Open-Source				
BEM [29]	**0.735**	0.704	0.208	0.627

Table 5.1: Precision (*P*) and recall (*R*) of the bounding boxes generated for the grounded answer generation task.

ficantly superior metrics on FinAM. Finally, BEM, which is optimized for answer correctness, demonstrates robust performance on English datasets. However, it proves ineffective on FinAM due to the language discrepancy between the corpus and the training data.

5.4 Conclusions

In this chapter, we addressed the critical issue of reliably assessing RAG systems generated answers with and without ground truth data by implementing and end-to-end RAG pipeline and utilizing both proprietary financial data (FinAM) and public narrative datasets (NarrativeQA). Our investigation yielded several key insights regarding the efficacy of "LLM-as-a-judge" and semantic metrics. First, correlation with human judgment appears to be strong on reference-based metrics, where LLMs are able to determine the correctness of the generated answer. However, on reference-free metrics, the correlation with human judgment is significantly reduced. Second, the LLM evaluation frameworks are sensitive to the utilized model and input language. Specifically, we observed significantly better results on the public corpus while our proprietary datasets was more difficult to evaluate for all models and frameworks. In parallel, the reduced performance on the Italian dataset suggest that advanced models are requisite for non-English, specialized domains.

Ultimately, this study highlights that while automated evaluation frameworks are promising, they are not yet a complete substitute for human verification, particularly in specialized languages and domains. The limitations of current reference-free metrics underline the necessity for systems that not only generate answers but also provide transparent evidence for their claims. This motivates the research presented in subsequent chapters, which moves beyond simple evaluation toward explainability and context attribution to enhance user trust.

Chapter 6

Context-Attribution

Nowadays, most AI assistants are powered by LLMs capable of handling multiple tasks. Retrieval Augmented Generation ehances factual accuracy and reduces hallucinations by incorporating external knowledge sources during generation. However, the chance of generating hallucinated or non-factual text [44, 190] is not null and forces the demand for attributable answers in order to enhance trust toward these tools.

Context-Attribution, the task of linking LLM-generated sentences with portions of the input context [142, 241], is particularly relevant in RAG systems [110]. Evidence suggests that corroborative context-attribution, which identifies evidence at support of the generated statements [46], is essential for making LLM-generated content more transparent, trustworthy, and verifiable [136]. By linking the generated outputs to the retrieved content, context-attribution supports fact-checking, grounded summarization, and verifiable question answering. It also enables users to assess the quality and relevance of the sources behind each claim, making it a key component for responsible AI deployment.

Recent advances in LLMs have significantly boosted the potential of context-attribution methods achieving strong results with fine-tuning [142, 130, 8, 241] and zero-shot prompting [69, 237] even on open domain questions [150]. Proprietary LLMs such as GPT-4 [1], Claude 3.7[1], or Gemini[2] have demonstrated strong performance in tasks involving context-attribution. Their advanced reasoning capabilities, understanding of complex context relationships, and access to vast

[1] https://www.anthropic.com/claude/sonnet
[2] https://gemini.google.com/

internal representations make them effective at associating generated content with relevant source passages. However, LLMs have notable limitations when used for post-generation context-attribution. First, proprietary LLMs achieve high accuracy but are associated with significant costs at scale due to expensive APIs and data-center-grade GPUs. Additionally, the nature of these models restricts adaptability to specific domains. Smaller, open-source LLMs offer advantages in this sense but often exhibit limited performance, especially in nuanced tasks such as identifying contributive sources or distinguishing between relevant but non-causal content.

Smaller Language Models represent efficient alternatives, offering advantages in terms of costs and control, being cheaper to train and deploy. Cross-encoders, common in reranking [179] jointly encode the query and a passage. Unlike bi-encoders, which compute separate embeddings for each, cross-encoders perform unified encoding, identifying complex dependencies and returning fine-grained relevance scores. Cross-encoders are particularly suitable for nuanced tasks like context-attribution and offer fast inference when applied to a limited set of candidates. Their ease of fine-tuning and controllability further empowers researchers to adapt models swiftly without the need for retraining multi-billion parameters models.

This study explores the application of LLMs and cross-encoders in post-generation context-attribution tasks. We examine the performance gap between proprietary and open-source LLMs as well as frozen and fine-tuned cross-encoders. Our findings indicate that, with limited hyperparameter tuning, cross-encoders can perform comparably to LLMs for post-generation context-attribution, while open-source LLMs consistently under-perform. Our contribution is three-fold: (i) we demonstrate how LLMs can be employed for post-generation context-attribution, (ii) we introduce a novel application of cross-encoders as context-attributors, providing a model agnostic solution not explored in prior works, and (iii) we present the first comparison of context-attribution performance across proprietary LLMs, open-source LLMs, and cross-encoders, for both in-line and answer-level citations.

6.1 Literature Review

Context attribution remains an open challenge, with recent efforts extending beyond textual grounding to visual contexts. On the textual side, Gao *et al.* [69] introduced ALCE, an automatic benchmark for evaluating LLMs ability to generate text with verifiable citations, providing reproducible metrics for fluency, correctness, and citation quality to mitigate hallucinations. Ye *et al.* [234] proposed AGREE, a learning-based framework that fine-tunes LLMs for grounded and citation-accurate responses using automatically constructed data, coupled with test-time adaptation to iteratively retrieve additional evidence, thus enhancing factual reliability. ContextCite [46] uses the log-probability drop to identify relevant text chunks in source documents and tunes a linear model to retrieve the top-k relevant sources.

6.1.1 LLM-based approaches

First attempts at context-attribution were mostly LLM-based and applied to open-domain question answering. WebGPT [150] relies on a text-based web browsing environment and uses a LLM to trigger information extraction from portions of the page but it is constrained to generate answers and in-line citations in a single forward pass. RARR [67] uses Google search to identify evidence in support of the information contained in a generated answer and uses the retrieved evidence to revise the answer but does not provide fine-grained attributions.

Among zero-shot solutions, [69] tested post-hoc and rag methods to identify the optimal approach of generating text with citations but only considers LLMs and retrievers for in-line context-attribution, [67] used context-attribution after open-book generation for answer revision but only uses bi-encoders and doesn't evaluate their attribution performance, [163] repurposed LLMs as classifiers to identify the citations for the LLM generated output.

[142] trains Gopher [173] with reinforcement learning to generate answers with quotes, [130] attempts a watermarking strategy to link the generated content with passages from the training data, CaLM [8] employs factual consistency models to filter synthetically generated data and uses focused learning to concentrate the backpropagated information around the generated answer rather than the input passages, LongCite [241] uses a retriever approach to identify citations for synthetically generated answers and uses focused learning for tuning, and TruthReader [113] trains LLMs to perform during-generation in-line citations in a RAG pipeline, however, they all rely on expensive LLM fine-tuning strategies by

generating citations and answers in a single step.

6.1.2 Encoders-based approaches

Cross-encoders [179] can be derived from any LM that allows the concatenation of two strings. Similarly to bi-encoders, cross-encoders are used to rank text passages in relation to a query. Differently from bi-encoders, cross-encoders do not output embeddings for the input query and text passage but a score representing a task-dependent probability. Classical cross-encoders are built on top of bidirectional Language Models by adding a classification head and trained with contrastive learning [226, 247] however, some cross-encoders are built on top of LLMs [112].

Relying solely on the generation capabilities of LLMs requires the model to have citation generation capacities. This ability is not guaranteed—especially for smaller LLMs—and often breaks down in specialized domains or complex tasks where accurate context-attribution becomes even more challenging. To address this, [149] introduces retriever-based methods for the context-attribution task, though the comparison is limited to embedding similarity models used as baselines against LLM-generated sources. [27] concatenates the question and answer, and applies dense retrieval to identify the most relevant supporting passages though only uses cross-encoders for evaluation, not attribution. [46] uses a probabilistic framework and employs an LLM to generate the answer along with its probability, iteratively masking portions of the context and observing changes in log probability. A linear model is fit to estimate the log probability shifts and locate the most influential sources. However, the approach lacks a formal evaluation of discriminative capabilities, a key requirement in production settings. Finally, [163] repurposed LLMs as cross-encoders by appending a classification head, enabling the model to function explicitly as a context-attribution classifier.

6.2 Experimental Setup

6.2.1 Datasets

We evaluate post-generation context-attribution using 4 datasets. The first is a proprietary dataset composed of legal documents, annotated in collaboration with an undisclosed client. The remaining are derived from well established datasets for QA and RAG benchmarks: TREC-RAG [169], ASQA [195], and ELI5 [64]. The datasets statistics are reported in Table 6.1.

Proprietary Corpus. Our proprietary corpus consists of legal documents provided by an undisclosed client. It includes Italian financial regulations and internal corporate materials which reflect the company's operational knowledge and regulations (including emails and internal communications). The dataset is made of approximately 37,000 pages, varying significantly in length and content. Annotation is partially conducted by the client's legal experts using a custom web-based application connected to a RAG system.

The RAG system integrates open-source retrievers, an open-source reranker, and a combination of open-source and proprietary LLMs. Specifically, the retriever is implemented using reciprocal rank fusion [47] based on BM25 [180] and a fine-tuned version of E5[3] [215]. The reranker is implemented through GTE [117] while the reader was based on GPT-4o and LLaMa3.1-8B [75] depending on the settings picked by the user. The chat interface enables the user to prompt the model with a query and provide feedbacks to the answer and the top-k retrieved passages. Passages are created from the document pages with a maximum length of 4096 characters and consecutive passages are merged.

The client's team was tasked with prompting the system, evaluating the quality of the generated answers, and identifying the relevance the retrieved passages with a boolean classification. In a second step, our annotators assessed whether each retrieved passage entailed the answer generated by the RAG pipeline, refining the previous annotations. In total, 50 examples were refined by our annotators: 25 were allocated for evaluation and early stopping, 13 for hyperparameter tuning, and 12 for testing. Each example is composed of a question, an answer, and up to 5 retrieved passages.

TREC-RAG. The TREC-RAG dataset is originally designed to evaluate factoid QA systems, while the RAG variant repurposed the corpus for retrieval-based LLM generation by pairing questions with relevant evidence passages. In its ori-

[3]https://huggingface.co/intfloat/multilingual-e5-large

Dataset	Passages	Tokens	Passages per question	Split Size			
				train	eval	hyp	test
Proprietary	37,016	333.79	4.45	27,000	25	13	12
TREC-RAG [169]	6,856	220.79	8.56	6,000	25	25	30
ASQA [195]	487,643	584.70	10.00	33,693	25	25	200
ELI5 [64]	89,115	114.47	10.00	86,904	25	25	200

Table 6.1: Statistics of the datasets used for context-attribution.

ginal construction, the retrieval corpus consists of Wikipedia articles, segmented into passages of fixed length. We adopt a subset[4] from the Ragnarok framework [169] that contains 100 retrieved passages for each question and the top-20 associated reranked passages. The subset contains annotations about the relevance of each passage with respect to the query, which are inconsistent with our task of evaluating consistency with respect to the generated answer. The dataset answers are originally generated with GPT-4. From each question-answer pair, we keep the top-20 passages for annotations and inference and use the remaining 80 for synthetic data generation and training.

Due to budget limitations, we limit passage annotation. First, we sample 25 question-answer pairs for early stopping, 25 for hyperparameter tuning, and 30 for testing. Then, we use a bi-encoder[5] to obtain passage-level embeddings and automate k-means clustering with a maximum of 10 clusters, selecting the optimal number using the Average Silhouette Width [181] method.

ASQA. ASQA was created for long-form factoid question answering focusing on ambiguous questions that require nuanced and multi-faceted responses. ALCE [69] randomly samples 1000 question-answer pairs from the original version of the dataset and manually curates them. Then, it uses the 2018 Wikipedia dump for the construction of the corpus by chunking Wikipedia articles into 100-words passages and uses an off-the-shelf bi-encoder to retrieve 100 passages for in-context learning and context-attribution. We utilize the same Wikipedia dump and associate each question-answer pair with the articles whose passages were retrieved for ALCE. Each article is split into 800-words long chunks with a 200-words stride. Initially, we split the dataset into training, evaluation, and testing ensuring that no Wikipedia article appears in more than one split. Then, due to budget limitations, we only annotate 50 question-answer pairs from the evaluation set: 25 for early stopping and 25 for hyperparameters tuning. Finally, we sample 200 examples from the test set for Natural Language Inference (NLI) metrics. We use an off-the-shelf cross-encoder[6] [247] to rank passages using question-passage pairs. The top-10 passages are retained for testing in order to mimic the task created with the client.

ELI5. This is a long-form question answering dataset derived from Reddit and it is not associated with a retrieval corpus. With the same approach used for ASQA,

[4]https://github.com/castorini/ragnarok_data/tree/main/rag24/retrieve_results/RANK_ZEPHYR

[5]https://huggingface.co/cross-encoder/ms-marco-MiniLM-L6-v2

[6]https://huggingface.co/Alibaba-NLP/gte-multilingual-reranker-base

ALCE provides a manually curated version of the dataset. For the construction of the retrieval corpus, it relies on a dump of Sphere [168] and uses an off-the-shelf retriever to identify 100 relevant passages related to each question. We utilize the answers provided with the original dataset and we adopt the 100 passages retrieved in ALCE as context for context-attribution ensuring that each article of the corpus appears in a single split. We then divide the dataset into training, evaluation, and test sets. From the evaluation set, we sample and annotate 25 question-answer pairs for early stopping and 25 for hyperparameter tuning. Similarly, we sample 200 examples from the test set for natural language inference (NLI) metrics. As for ASQA, we employ the same cross-encoder to rank the passages associated with each question and keep the first 10 for evaluation purposes.

6.2.2 Annotation Process

Annotators were given the following instructions: (i) read the query, the answer, and the passage, (ii) determine if the answer was positively, negatively, or not entailed by the passage. We merged the last two classes to focus on entailment, thus shifting the problem from NLI to binary classification. As far as inter-annotator agreement is concerned, we accounted for it by having annotators resolve conflicts. After the annotation process, they evaluated their own annotations together and identified three cases requiring conflict resolution.

6.2.3 Selected Models

We employ LLMs and Cross-Encoders to solve the attribution problem. The following sections detail the reasons behind our model choices.

LLMs

We employ a small open-source and a proprietary model. For open-source models, we rely on a quantized version of LLaMa3.1-8B, used as the reader in the annotation process of our private dataset. The model is deployed in a machine with a single A100 GPU with 40GB of vRAM and all inferences are done through VLLM [105]. For the proprietary counterpart, we rely on GPT-4o, motivated by its exceptional reported capabilities and its already established utilization in Altilia's other projects. The models are prompted with two templates: the *baseline template* and the *CoT template*. The same templates are used for the two models.

Baseline Template

```
User: {instructions}
You must respond only with "Yes" or "No".

QUESTION: {question}
PASSAGE: {passage}
ANSWER: {answer}

Does the PASSAGE contain the information needed to produce
the ANSWER? Answer only with "Yes" or "No".
```

Figure 6.1: Baseline attribution prompt template.

Moreover, structured output generation using Pydantic[7] facilitated the attribution with GPT-4o but was found to be prohibitively slow on open-source models. Attempts of using LLaMa3.1 without Pydantic led to unstable behavior, yielding only a limited number of successful CoT generations. Consequently, only the baseline template is reported with LLaMa3.1.

Baseline Template. This consists in the simple concatenation of generation instructions and the question-answer-passage triple as illustrated in Figure 6.1. The generation instructions force the machine into the role of domain expert, explain the nature of the input, and introduce the entailment task. The model is tasked with generating only `"Yes"` or `"No"`.

CoT Template. For each triple, the CoT template concatenates structured generation instructions with the triple to infer entailment as illustrated in Figure 6.2. The system instructions describe the task and the input, the structured generation instruction task the model with the generation of a JSON containing *"common information identification"*, *"reasoning"*, and *"entailment"*. The task is threefold, the model must do three operations in sequence: (i) identify the common information between the answer and the passage, (ii) reason about information entailment, and (iii) return "ENTAILED" or "NOT ENTAILED".

[7] https://docs.pydantic.dev/latest/

CoT Template

System: {instructions}
{Structured generation instructions}

User: Document:
[DOCUMENT_START]
{document}
[DOCUMENT_END]
Question: {question}
Answer: {answer}

Now prepare to do the following:

- Report the common information between the document and the answer.
- Explain whether the answer was created starting from the document or not.

Figure 6.2: CoT prompt template example.

Cross-encoders.

To evaluate cross-encoders' capabilities in post-generation context-attribution, we utilize gte-multilingual-reranker-base[8] with approximately 500 million parameters as a lightweight baseline competitor to LLMs. We hypothesize that such a small model is unable to tackle the complex semantics of long passages to infer entailment, for this reason we identify three strategies for using the cross-encoder: *answer-passage*, *sentence-sentence*, and *sliding-window*. Sentence tokenization is done through langid[9] and nltk[10]. With all strategies, if a sliding window is identified as a citing source, the entire passage is labeled as a citing source.

Answer-Passage (ap). This strategy consists in feeding the cross-encoder with the full answer and passage. Then, the model outputs the probability that the answer is entailed in the passage. Inferences are ranked by descending probability and the top-k passages are retained as the attributed context. This strategy is computationally inexpensive, as it requires a single forward pass, but might fail to capture the complex semantics of the passage when the context is saturated.

[8] https://huggingface.co/Alibaba-NLP/gte-multilingual-reranker-base
[9] https://pypi.org/project/langid/
[10] https://www.nltk.org/

Sentence-Sentence (ss). The sentence-sentence strategy couples each sentence from the answer with each sentence from the associated passages and outputs the probability of entailment for each pair. This strategy is expected to capture more granular semantics but it requires a high number of inferences rendering the computational overhead quite expensive. Moreover, this approach might fail to capture long dependencies.

Sliding-Window (sw). Finally, the **sliding-window (sw)** strategy utilizes a sliding window of w_a sentences for the answer and w_p for the passages. Passages and answers are chunked accordingly with a stride of a single sentence. Then, each couple of answer and passage chunk is fed to the cross-encoder to obtain the entailment probability. Through this approach, we expect to identify the optimal granularity level but requires tuning of the sliding windows hyperparameters.

Hyperparameters Tuning. Due to the excessively low number of annotated pairs, we expect the hyperparameters tuning set not to be fully representative of the test set. For this reason, we limit $2 \leq w_a \leq 4$ and $w_a \leq w_p \leq 8$. This greatly reduces the number of inferences required for a single hyperparameters search and is consistent the hypothesis that (i) the passages sliding windows benefit from longer chunks due to long information dependencies and (ii) cross-encoders benefit from the extra context of longer answer chunks to disambiguate false positives. On the proprietary dataset and TREC-RAG we are able to compare model predictions with human annotations, while we can only run NLI-based context-attribution evaluation for the other datasets. Moreover, for all datasets we analyze how performance varies with the score thresholds.

6.2.4 Fine-tuning Settings

We fine-tune the cross-encoder on a single Tesla T4 GPU. We set the initial learning rate to 2e-7 and utilize linear reductions until the end of the epochs. Fine-tuning is run for a single epoch with batch size of 4 and maximum sequence length of 2048 tokens. For early stopping, we compute the F1 score at a threshold of 0.9 on the annotated evaluation set. Grid search is not run since it is beyond the scope of this work. Moreover, the strict operational constraints and the limited availability of computational resources didn't allow a full grid search.

6.2.5 Synthetic data generation.

Due to the limited availability of annotated data, we rely on synthetic data generation to gather evidence with similar distribution to the target examples. We rely on LLaMa3.1-8B and use few-shot learning to generate synthetic question answer pairs. Specifically, we first run k-means on the training passages to cluster them into 10 groups. For each cluster, we retain the example closer to the centroid of each cluster and manually annotate it with a question and an answer. Finally, for each passage of the training set we sample two annotated question-answer-passage pairs and use few-shot learning to generate a new question and answer for the training passage. The few-shot examples solve a bipartite problem. First, they allow for the generation of examples with a distribution similar to the one from human annotations. Second, they enable LLaMa3.1 to generate structured outputs, simplifying answer parsing.

This process utilizes scikit-learn[11] k-means implementation on the embeddings computed through an off-the-shelf bi-encoder model[12]. During the generation process, in order to avoid out of memory errors, we filter all the prompts exceeding the 6k tokens length. A sample of 30 items per dataset is examined and annotators assessed that the queries and answers were meaningful and contextually relevant to the pages and to each other. The manual review did not uncover inconsistencies, reinforcing the reliability and coherence of our synthetic data.

6.2.6 Evaluation Metrics

On the fully annotated test sets of the proprietary dataset and TREC-RAG, we assess model performance with precision, recall, and F1 metrics. For the other datasets, we employ a modified version of citation precision and citation recall [123, 163] that computes information entailment rather than sentence or answer entailment. We do not provide comparisons with other studies' results due to the underlying differences in the considered tasks.

Information Entailment. Let a be the answer obtained in a RAG setting and p_j be the j^{th} passage identified by a retriever. We employ a NLI model to transform each sentence of the answer into a list of mutually exclusive information and then a NLI model to infer the entailment of each piece of information in each passage

[11] `https://scikit-learn.org/stable/`

[12] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L6-v2

p_j of the retrieved context. Formally:

$$I = \{i_1, i_2, ...\} = NLI_{info}(a)$$

$$e_{q,j} = NLI_{entail}(i_q, p_j)$$

where $e_{q,j} \in \{0, 1\}$ is the entailment of information i_q in the passage p_j. We instantiate both NLI_{info} and NLI_{entail} with GPT-4o.

Citation Recall. We assume that each sentence of the answer a is obtained by concatenating and paraphrasing some information $I = \{i_1, i_2, ..., i_n\}$. Let $P = \{p_1, p_2, ..., p_m\}$ be the set of attributing passages. We define citation recall as the proportion of information that is entailed by at least one of the attributing passages. Let $Q = \{1, 2, ..., n\}$ be the indexes of I, $J = \{1, 2, ..., m\}$ be the indexes of P. Let $I_{ent} = \{i_q \mid \forall q \in Q,\ \forall j \in J,\ e_{q,j} = 1\} \subseteq I$ represents the set of information entailed by at least one passage from P.

We define citation recall as

$$r_c = \frac{|I_{ent}|}{|I|}$$

be the answer-level citation recall. Entailments are computed at the sentence-level and aggregated at the answer-level before computing citation recall.

Each information in the answer is considered a true positive if it is entailed by at least one retrieved passage and a false negative otherwise. For context-attribution, we let each information of the answer attend each retrieved passage. Then, the information is classified as a true positive if it is entailed by at least one retrieved passage and a false negative otherwise.

Citation Precision. We define citation precision as the proportion of passages from P that entail at least a portion of the information from a string s. This definition doesn't penalize for repeated information in order to fairly evaluate cross-encoders and LLMs in a scenario in which each passage is processed independently, thus at each inference step the model is not aware of the already identified information. Moreover, authors often include multiple citations to strengthen the perceived groundness of a certain sentence. Let $P_{ent} = \{p_j \mid \forall q \in Q,\ \forall j \in J,\ e_{q,j} = 1\} \subseteq P$ be the set representing all the passages that entail at least a portion of the information conveyed by s.

We define the citation precision as:

$$p_c = \frac{|P_{ent}|}{|P|}$$

For in-line citations, each passage is considered as a true positive if it entails at least one information from the set of sentences it is associated to and a false positive otherwise. For context-attribution, each passage is considered as a true positive if it entails at least one information from the answer and a false positive otherwise. As for citation recall, entailments are aggregated at the answer level before computing citation precision.

6.3 Experimental Results

6.3.1 Context-Attribution

Table 6.2a and Table 6.2b show the strict context-attribution metrics computed on the proprietary dataset and TREC-RAG. The *parameters* column represents the settings identified through hyperparameters tuning. The first three rows of the GTE and GTE$_{tuned}$ models are associated with a single value, which is the average of the metrics obtained with the three settings. This happened due to the models achieving the same performance with various hyperparameter combinations in the hyperparameters tuning set.

GTE and GTE$_{tuned}$ show significant improvements through hyperparameters tuning, with F1 scores or 84.15 and 87.55 on the proprietary dataset and 75.68 and 77.41 on TREC-RAG. In all four cases, the score is the highest relative to the dataset and model. On the proprietary dataset, recall is consistently higher with the *ss* approach due to the answers short nature and the high number of retrieved passages. On TREC-RAG, that requires longer dependencies for correct attribution, GTE doesn't achieve the same results.

Among LLMs, LLaMa3.1 performs poorly on both datasets with both low precision and recall. With GPT-4o, the same template achieves good results on the proprietary dataset, with the highest precision observed, but very low scores on TREC-RAG. The CoT template performs comparably to the fine-tuned cross-encoder counterpart, with similar results on the annotated datasets.

On manually annotated data, cross-encoder can perform comparably to LLMs. Specifically, we highlight that hyperparameter tuning allows the model to maximize performance on the test set. Moreover, fine-tuning the model on the synthetically generated dataset further improves the performance around the same level of

Model	Method	Parameters	p	r	f1
Cross-Encoders					
GTE	sw	3/2/4/0.6	84.76	85.67	84.15
		5/3/5/0.6			
		5/3/5/0.6			
	ss	10/1/1/0.6	76.76	92.00	83.64
	ap	2/full/full/0.5	75.00	72.00	73.47
GTE_{tuned}	sw	3/2/6/0.6	85.88	89.33	**87.55**
		10/2/4/0.8			
		10/3/4/0.8			
	ss	10/1/1/0.7	76.76	92.00	83.64
	ap	4/full/full/0.8	84.62	88.00	86.27
LLMs					
GPT-4o	ap	CoT	80.00	**96.00**	87.27
	ap	baseline	**90.48**	76.00	82.61
LLaMa-3.1	ap	baseline	69.23	72.00	70.59

(a)

Model	Method	Parameters	p	r	f1
Cross-Encoders					
GTE	sw	10/3/5/0.6	74.08	82.60	75.68
	ss	3/1/1/0.7	**80.55**	48.23	58.78
	ap	10/full/full/0.5	70.79	81.73	72.42
GTE_{tuned}	sw	10/4/5/0.9	70.32	90.82	77.41
	ss	10/1/1/0.9	63.11	**98.61**	76.96
	ap	10/full/full/0.9	67.32	80.37	70.05
LLMs					
GPT-4o	ap	CoT	73.98	89.29	**78.74**
	ap	baseline	45.03	29.64	31.17
LLaMa-3.1	ap	baseline	57.04	42.01	42.05

(b)

Model	Method	Parameters	p_c	r_c	f1_c
Cross-Encoders					
GTE	sw	40/2/5/0.6	91.48	69.16	78.77
	ss	20/1/1/0.6	81.38	69.57	75.06
	ap	5/full/full/0.5	88.96	68.27	77.25
GTE_{tuned}	sw	20/2/5/0.7	88.76	**70.05**	78.31
	ss	5/1/1/0.7	85.44	66.66	74.90
	ap	10/full/full/0.8	90.27	69.34	78.47
LLMs					
GPT-4o	ap	CoT	**96.90**	66.73	**79.03**
	ap	baseline	96.09	59.54	73.56
LLaMa	ap	baseline	93.96	52.11	67.14

(c)

Model	Method	Parameters	p_c	r_c	f1_c
Cross-Encoders					
GTE	sw	5/2/6/0.4	51.32	**17.20**	**25.76**
	ss	5/1/1/0.4	47.28	16.91	24.91
	ap	5/full/full/0.4	49.54	14.94	22.96
GTE_{tuned}	sw	5/3/7/0.7	57.96	16.09	25.19
	ss	5/1/1/0.4	49.88	16.97	24.92
	ap	5/full/full/0.6	53.61	15.17	23.65
LLMs					
GPT-4o	ap	CoT	81.16	12.46	21.60
	ap	baseline	**90.08**	5.85	10.99
LLaMa	ap	baseline	71.83	9.12	16.19

(d)

Table 6.2: Attribution metrics for: the proprietary dataset (a), TREC-RAG (b), ASQA (c), and ELI5 (d). Parameters are `top-k/answer sw/passage sw/score thr`.

the best version of GPT-4o. This corroborates our hypothesis that cross-encoder are limited in their abilities to manage dependencies dynamically. Instead, modifying the window size is the most successful approach. Generally, the larges MLLMs are capable of automatically manage dynamic dependencies when equipped with reasoning.

Table 6.2c and Table 6.2d report the context-attribution performance on ASQA and ELI5 respectively, in which test set annotations were not available. In this case, the results highlight that hyperparameter tuning is still the preferred mean to increase model performance. However, we observe limited to no improvements with fine-tuning. Experiments with annotated dataset in Table 6.2a and Table 6.2b already proved that GPT-4o doesn't achieve 100% accuracy on entail-

ment, introducing some noise in NLI and, as a consequence, in NLI evaluations. Hyperparameter tuning is executed on manually annotated data while metrics are computed by mean of a Large Language Model, explaining the difference in trend observed between the two sets of experiments.

Among LLMs, GPT-4o consistency performs better that all other approaches when equipped with CoT reasoning. Again, the *baseline* template under-performs with respect to all other approaches.

6.3.2 In-line citations

We compare cross-encoders and LLMs for in-line citations in a standardized setting. We infer information entailment on 8 sentences long passages with a stride of 2 sentences. Moreover, we fix top-k to 20 and we modify the LLM prompts to utilize a single sentence from the answer and no question information. Since we don't possess sentence-level annotations, we only utilize Natural Language Inference to compute metrics.

Figure 6.3 shows the in-line citations performance on all datasets. The tuned model systematically exhibits a smoother behavior with respect to the frozen counterpart and always outperforms its frozen version and LLaMa3.1. Both versions of GPT-4o templates have similar behavior and are rarely outperformed by other models. Regardless of the fact that cross-encoders were not trained for sentence-level entailment, tuning on synthetically generated data is found to slightly improve attribution capabilities. The comparison is in favor of Large Language Model (LLM)s, capable of high scores with and without complex prompting strategies when dealing with in-line citations.

6.3.3 Answer-Level Context-Attribution

Figure 6.4 shows F1 scores for the *ap* setting with varying score thresholds on all passages. Since we possess manual annotations at the passage level on the proprietary dataset and TREC-RAG, we only utilize NLI on the remaining corpus.

As for in-line citations, the model fine-tuned on synthetic data always outperforms the frozen model by at least a small margin and always shows a smoother behavior, while the frozen model can achieve higher scores only locally. Interestingly, the tuned model exhibits a lower degree of diminishing returns on TREC-RAG and ELI5. GPT-4o with CoT has strong performance on all datasets, always ranking in the top-3, but ELI5. While the baseline GPT-4o can achieve satisfactory results, LLaMa3.1 always ranks among the worst models for the task.

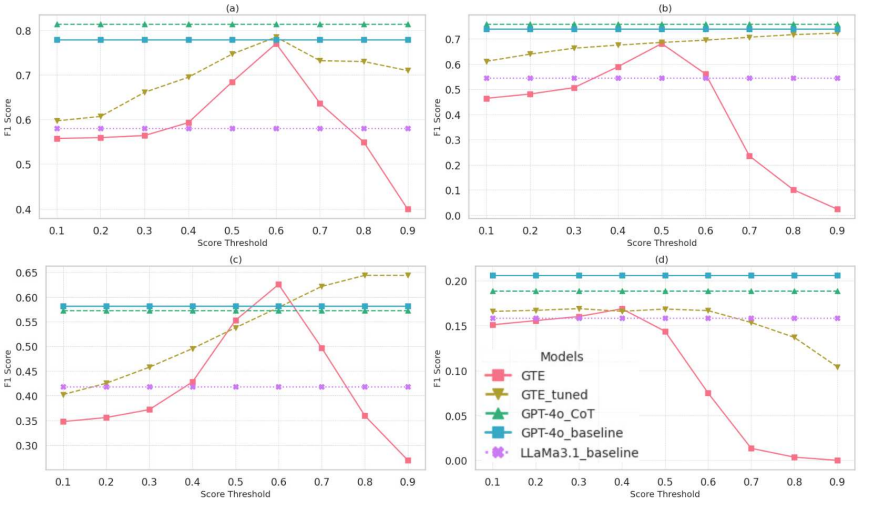


Figure 6.3: In-line citations metrics for each model on the proprietary dataset (a), TREC-RAG (b), ASQA (c), and ELI5 (d).

Open source LLMs do not compare favorably to either proprietary counterparts or cross-encoders, underscoring their limited capabilities at managing dynamic dependencies. Finally, we observe that hyperparameters tuning always allows the frozen model to obtain strong performance and occasionally surpasses proprietary models when computing attribution scores with NLI.

6.3.4 Error Analysis

We conduct manual evaluation of 30 randomly sampled items per dataset. Our analysis indicates that: (i) pretrained retrievers often misclassify similar-looking passages, (ii) tuned retrievers improve on this but still misclassify passages occasionally, (iii) LLaMa relies heavily on surface cues, sometimes misled by key entities, (iv) GPT-4o may overlook marginally mentioned but crucial information, and (v) CoT can still make reasoning errors, especially with implicit subject shifts.

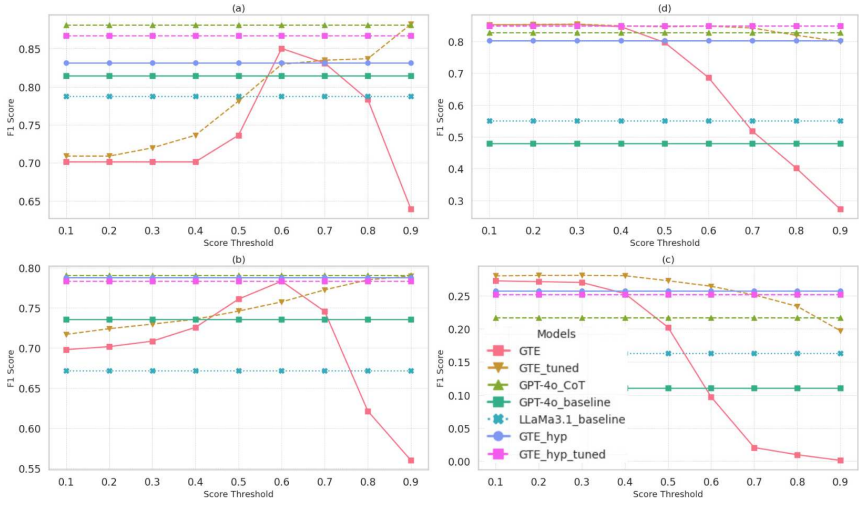


Figure 6.4: Full-answer context-attribution performance on proprietary dataset (a), TREC-RAG (b), ASQA (c), and ELI5 (d).

6.3.5 Cost Estimate

In our experiments, LLM- and reranker-based approaches differ in throughput and cost. The tuned cross-encoder averages $2.15_{s/query}$ ($0.6_{s/query}$ optimizing parallelization) yielding $\sim 1674_{queries/h}$. With a Tesla T4 at $0.526_{\$/hour}$ (AWS), this results in $\sim 0.31_{\$/1000queries}$. In contrast, LLM inference (GPT-4o) on 1000 queries with $\sim$4M tokens costs $\sim$10 \$. Parallelism improves LLM throughput but does not close the cost gap and cross-encoder deployment must be calibrated on cost and utilization criteria.

6.4 Technical Implementation

The proposed context-attribution approach was deployed in a production environment. The system is not an end-to-end complete pipeline, rather a microservice activated by the user the get the attribution after the RAG pipeline has completed answer generation. The system is designed to be model agnostic, maintain high

throughput, and dynamic scaling.

The complete attribution pipeline can be divided in three parts:

- **Data Ingestion.** The underlying corpus is processed at ingestion time, where documents are chunked and transformed into embeddings stored within a Milvus vector database. When a user submits a query, the system performs hybrid retrieval limiting the research to the documents selected by the user.
- **RAG.** When the user submits a request, **RRF!** uses a bi-encoder, BM25, and the cross-encoder to retrieve the top-k passages and subsequently feed a LLM with them. An abstraction layer roots the requests to the selected LLM through LiteLLM, allowing the user to interchange the underlying generative model based on user preferences.
- **The Attribution Microservice.** Once the answer is generated, the post-hoc context-attribution phase is triggered. The cross-encoder is deployed using Ray Serve, with autoscaling capabilities. By decoupling the cross-encoder from the main RAG generation pipeline, the architecture scales independently. This ensures that the attribution task, which is computationally intensive, does not bottleneck the retrieval and text generation steps.

6.5 Conclusions

We investigated the effectiveness of semi-supervised and frozen cross-encoders as a lightweight alternative to LLMs for post-generation context-attribution. Through comprehensive experiments across four datasets we demonstrated that fine-tuned cross-encoders can match and sometimes surpass LLMs in coarse- and fine-grained attribution tasks, particularly when tuned on synthetic data.

Our results highlight several key takeaways. First, open source LLMs are not well suited for context-attribution. Second, proprietary models such as GPT-4o provide consistently strong performance for sentence-level attribution but are inconsistent on answer-level attribution, meaning that they become less performing with the increased answer length. Third, prompt engineering mitigates this issue and improves overall attribution performance but it requires a model with structured data generation capabilities. Fourth, cross-encoders offer a scalable and cost-efficient alternative, particularly when computational resources or annotated data are limited. Cross-encoders do not require prompt engineering and

can be fine-tuned with semi-supervised strategies and minimally annotated data. In fact, synthetic data generation proved to be effective to overcome annotation bottlenecks and enhance model performance in low-resource environments.

Overall, this study underscores the practicality and promise of semi-supervised cross-encoders for robust, interpretable, and resource-efficient context-attribution, especially in production-oriented or specialized domains.

Chapter 7

Visual Grounding

The rapid evolution of MLLMs have transformed the landscape of Document Visual Question Answering (DocVQA). Modern large-scale foundational multimodal models [30, 121, 243], built upon Transformer-based architectures [213], have achieved state-of-the-art performance in reasoning and generating textual answers based on visual inputs. However, as these models are employed in critical domains such as legal analysis, scientific research, and financial auditing, the answer accuracy is no longer enough. In document understanding, neural networks pose a significant challenge. While MLLMs can process vast amounts of information, they are prone to hallucinations, providing information that is either incorrect or not present in the source document [147]. Users require the answers to be *verifiable*, making simple QA evaluation superfluous. Visual Answer Grounding (VAG) refers to the task of identifying the evidence that supports a given answer in a Visual Question Answering (VQA) setting [11, 31, 222, 260].

Despite advancements in computer vision, there remains a gap between handling natural images and document images. MLLMs have become proficient at object detection in generic scenes. However, the same models struggle when tasked with locating answers within the dense layout of a document page [202]. The semantic complexity of text and the combination with visual structural elements presents a reasoning challenge that general-purpose vision encoders are unable to solve effectively in zero-shot settings. Even the largest state-of-the-art models fail to identify the spatial coordinates of the evidence supporting their own answers.

In the context of *document understanding*, answer grounding was initially

explored at the textual level by prompting Large Language Models (LLMs) [156] to generate verifiable citations supporting their answers, either through prompt engineering [69] or supervised fine-tuning [234]. The growing capabilities of MLLMs have subsequently enabled a natural extension of this task to the visual domain, giving rise to Visual Answer Grounding [51, 134, 135], a more intuitive and rapidly verifiable approach, yet one that remains an open challenge. Evidence from non document-related domains suggest that MLLMs exhibit strong answer accuracy but limited grounding capabilities [202].

Recent works on VAG demonstrate that fine-tuning foundation MLLMs on Document VQA datasets annotated with grounding information significantly improves attribution performance [134] compared to non-specialized models [16, 5, 36, 37, 255]. However, most existing benchmark datasets [51, 122, 134, 135, 204, 205, 220] primarily focus on the VQA task itself, providing limited or no explicit link between answers and supporting visual evidence. High-quality grounding annotations are essential not only for assessing the alignment between model responses and the supporting content, but also for training more reliable models and mitigating hallucination phenomena [147].

The primary reason preventing the development of grounding-aware models is data scarcity. While the number of DocVQA datasets is not low [204, 220], they mostly focus on text-generation tasks, providing tools to evaluate models that generate text without explicit linking to visual evidence. Constructing such datasets remains a non-trivial challenge. Manual annotation is prohibitively costly and labor-intensive, requiring humans to identify and verify the visual evidence that supports each answer [205]. Automatic approaches can reduce human effort but may introduce errors or hallucinated grounding, where the linked visual content does not faithfully correspond to the answer [69, 136, 146]. Furthermore, automated annotation often relies on paywall-protected APIs, increasing financial costs and limits large-scale dataset generation [51].

Building on these motivations, we develop two complementary components. First, we introduce **DocAttriBench** (DAB), a large-scale Document VQA dataset with VAG designed to provide higher-quality and more comprehensive annotations. DAB enables both fine-grained evaluation of VQA models attribution capabilities and effective fine-tuning of MLLMs using high-quality grounded data. To construct this dataset, we propose **Ma**sked-based **P**er**pl**exity-derived A**tt**ribution (**MAPPET**), a novel VAG framework capable of performing two key tasks: (i) localizing the spatial coordinates of the visual region that supports a given answer, and (ii) filtering out low-quality samples according to a custom metric and threshold.

Based on existing VQA datasets, DocAttriBench comprises 238,720 document images spanning diverse domains such as scientific articles, business reports, and digital slides, along with 295,075 sets of image, question, answer, and bounding box, covering visual elements like text paragraphs, tables, and figures. First, for each combination of document, question, and answer, a document layout analysis model extracts all the structural elements positions. Then, MAPPET iteratively masks layout elements and computes a perplexity-based score on the masked image. The attribution score is defined as the difference between the masked and unmasked image scores, defining the grounding region as the one achieving the highest value. Figure 7.1 depicts both the MAPPET task and DocAttriBench.

Extensive experiments confirm the reliability of MAPPET and the usefulness of DocAttriBench as a ground-centric benchmark. Human evaluation shows that MAPPET produces high-quality attributions, with accuracy exceeding 95% for some benchmarks after automatic filtering, validating its ability to discard noisy and ambiguous examples. Through the evaluation of sixteen state of the art MLLMs, we observe that zero-shot grounding is consistently weak, with the grounding score rarely surpassing 35% for the largest models ($>$30B) and often falling below 10% for the smaller ones ($\leq$3B). Finally, fine-tuned variants outperform their zero-shot counterparts by a large margin, even surpassing larger models in grounding accuracy, demonstrating the proposed dataset and attribution pipelines provides sufficient signal for supervision.

The remainder of this chapter is divided in x parts. First, we delve into the approaches that are currently available in the scientific literature. Then, we describe MAPPET and DocAttriBench. Finally, we explore the effectiveness of DAB for fine-tuning and evaluations of Visual Answer Grounding.

7.1 Literature Review

7.1.1 Document Visual Understanding

Document Visual Understanding aims to jointly reason over textual, visual, and layout cues in scanned or digital-born documents [199, 22]. Recently, large-scale multimodal models [243, 30] have become the leading approach in document visual understanding, jointly modeling visual layouts, text, and spatial relations within a unified multimodal space. Powered by large-scale image-text pretraining and transformer architectures, they enable rich cross-modal reasoning. Recent advances focus on OCR-free models [232, 236], which infer textual and structural semantics directly from images. These operate on high-resolution images, developing various strategies to manage the resulting computational load in order to improve document comprehension capabilities. End-to-end approaches such as Donut [99], PaLI-X [35], and Qwen2-VL [216] aim to process full-resolution document images directly, yet typically rely on image downscaling to maintain computational tractability. In contrast, tile-based models like UReader [233] and InternVL2 [36] enhance efficiency by partitioning documents into smaller regions processed independently. Hybrid strategies, as adopted in the LLaVa-derived models [111, 121], preserve global coverage by tiling full-resolution inputs while subsequently downsampling the aggregated visual representations.

7.1.2 Document VQA Benchmarks

Several benchmarks have been introduced for visual question answering on documents, but most suffer from limited scale, weak grounding, or low diversity. DocVQA [140] focuses on scanned UCSF documents but offers full annotations only for its 5k validation set and lacks bounding boxes. VisualMRC [205] provides 30k QA pairs over webpage screenshots with OCR and Region of Interest (ROI) annotations, but its construction relies on more than 500 human annotators for document selection, labeling, and QA creation, rendering the process expensive and slow. Moreover, the collected pages are generally short and structurally simple. LongDocURL [51] extends to multi-page documents but has misaligned boxes and limited verification. It contains only 2.3k QA pairs and relies on proprietary models, including GPT-4o for document-type classification and QA generation, as well as commercial tools for PDF parsing and layout extraction. DoclingMatix [155] offers synthetic instruction-response pairs at scale (2.4M images) without grounding annotations or page-level attributions. MMLongBench-Doc [135] tar-

Error type	Quantity
Hallucination	10%
Red Box	4%
Wrong Box	23%
Overall	33%

Table 7.1: Error rates for the source dataset PaperVISA.

gets long-context multi-modal reasoning across 135 PDF documents with 1k expert-annotated questions involving text, tables, and figures. While it introduces cross-page and unanswerable questions, the dataset remains small and provides no spatial grounding.

Structured resources like the VISA family [134] provide large-scale VQA datasets with varying grounding quality: Wiki-VISA (90k samples) and FineWeb-VISA (60k) construct question-answer pairs from the content of structural page elements; Paper-VISA (100k) constructs element-level grounding prompting MLLMs with boxes-overlayed images. Moreover, FineWeb-VISA lacks human verification. Other efforts, such as SlideVQA[204] cover 14k QA pairs across 52k slide images, supporting single-hop, multi-hop, and numerical reasoning (25.5% arithmetic), but does not include bounding boxes and provides limited reasoning supervision. VisualWebBench [122] WebQA split contains only 314 webpage QA samples without bounding box annotations.

7.2 PaperVISA Error Analysis

Fueling this research step, is the need to automatically annotated a large-scale dataset containing questions, answers, and annotated ROIs. The closest solution is used to prepare PaperVISA [134]. The original system is a three-steps approach. First, an off-the-shelf DLA model identifies structural elements from the document page. Second, an element is sampled and its bounding box overlayed on the document image with a red bounding box. In this section, we report the results of a careful inspection of 100 single-page examples sampled from the PaperVISA dataset. Detected error rates are reported in Table 7.1.

Among the inspected examples, 10% contain hallucinated answers (*Hallucination*) either referring to content not available on the document page or reporting incorrect numbers, figures, and names; 4% of the questions refer to the red bounding box overlayed on top of the image during the dataset generation process (*Red*

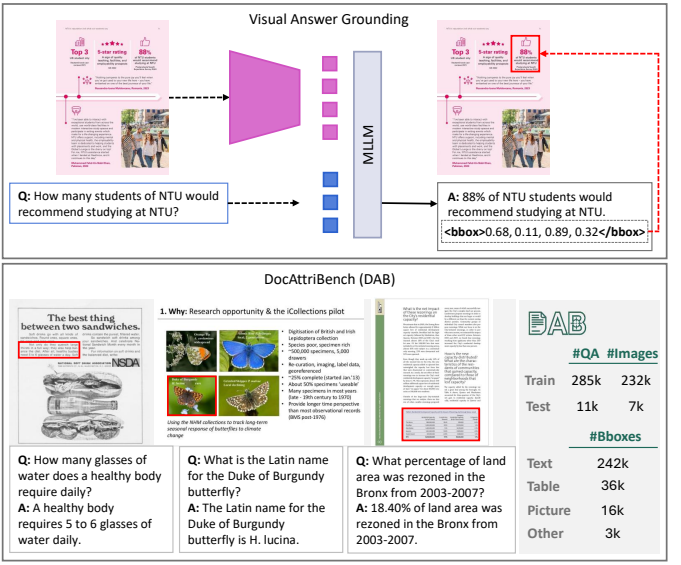


Figure 7.1: **(Top)** The Visual Answer Grounding (VAG) task, which our method advances. Specifically, a Multimodal Large Language Model (MLLM) answers an image query with text and a bounding box of the evidence region. **(Bottom)** Examples from our dataset, DocAttriBench (DAB) with dataset statistics: number of question-answer pairs (#QA), unique images, and bounding boxes per type (*i.e.*, text, table, picture, and other).

Box), which is unavailable at inference time and creates unsolvable examples; 23% of the sample contains wrong grounding (*Wrong Box*), *e.g.* the generated question-answer pair is not aligned with the bounding box that was overlayed to the document image. Finally, 33% of the inspected samples contains at least one of the above mentioned errors. While the hallucinated examples can still be used for training and evaluation of post-hoc grounding approaches, about 23% of the dataset contains wrong answer-box associations, introducing noise that makes the approach less attuned to both model training and evaluation.

7.3 DocAttriBench

In this section, we provide a comprehensive overview of the core features of DAB, with an emphasis on how the dataset is constructed. We first outline the statistics of the large-scale dataset produced by our automatic annotation pipeline. Next, we introduce the proposed Mask-based Perplexity-Derived Attribution (MAPPET) method. We then describe the procedure used to select public benchmarks that serve as the foundation of our dataset. Finally, we detail the full automatic annotation pipeline that enables scalable and consistent dataset creation.

7.3.1 Mask-based Perplexity-Derived Attribution

To automatically extract region-answer associations, we propose MAPPET, which is based on a perplexity-derived score designed to quantify the contribution of visual regions to a language model confidence in generating an answer. Perplexity [94] measures the probability of generating a sentence through a LLM. Let $T_n = \{t_1, t_2, ..., t_{n-1}\}$ denote a set of answer tokens and $I = \{i_i, i_2, ..., i_m\}$ be the set of image pixels. Given a MLLM, $P(t_k|T_k, I)$ is the conditional probability of generating token t_k given the previous tokens and the input image I. We omit the question and other context tokens in the formulation. The image-conditioned perplexity is defined as

$$\rho_I = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(t_i|T_i, I)\right), \tag{7.1}$$

where lower perplexity is associated with higher model confidence.

Let $Q \subset I$ be a region of pixels from I and let $I_Q = I/Q$ be the masked variant of I, obtained by occluding the pixels in Q. The attribution score of the region Q is defined as

$$\Delta_Q = \log(\rho_{I_Q}) - \log(\rho_I). \tag{7.2}$$

Intuitively, if the region Q contains information essential for the model to generate the answer, its removal increases perplexity resulting in a positive attribution score. In contrast, uninformative and irrelevant regions yield near-zero or negative scores.

7.3.2 Dataset Collection

To construct DAB, we aggregate data from eight publicly available document understanding benchmarks. The selected datasets satisfy two criteria aligned with

Source	Train		Test	
	#Docs	#Q&A	#Docs	#Q&A
DoclingMatix [220]	97,443	135,481	-	-
DocVQA [140]	1,114	4,070	123	467
VisualMRC [205]	7,640	17,066	2,143	4,857
VISA [134]	119,881	119,881	2,779	2,779
SlideVQA [204]	5,567	8,043	981	1,234
VisualWebBench [122]	-	-	108	233
LongDocURL [51]	-	-	688	688
MMLongBench-Doc [135]	-	-	253	276
DocAttriBench (DAB)	231,645	284,541	7075	10,534

Table 7.2: Statistics from the repurposed datasets.

our benchmark objective. First, all datasets provide document-oriented content, consisting of either scanned/rendered PDFs or document page images. This ensures domain consistency and focuses the benchmark on visual-text reasoning within structured documents. Second, each dataset adopts a question-answering format and contains single-page instances or multi-page examples with page-answer alignment, which makes spatial grounding possible.

Specifically, we construct DocAttriBench starting from eight public datasets: DoclingMatix [220], DocVQA [140], VisualMRC [205], LongDocURL [51], MMLongBenchDoc [135], SlideVQA [204], VisualWebBench [122], Wiki-VISA, Paper-VISA, and FineWeb-VISA [134]. Among them, VISA, VisualMRC, and LongDocURL already contain some form of grounding annotations, which we repurposed for DAB. DoclingMatix and FineWeb-VISA lack official test sets and are therefore used only for training. In contrast, LongDocURL, MMLongBenchDoc, and VisualWebBench are evaluation-oriented benchmarks and are used exclusively for testing. All other datasets follow their original train/test splits. The complete statistics and splits of our dataset are presented in Table 7.2.

7.3.3 Automatic Annotation

Our automatic annotation pipeline aims to extract the document regions that are most relevant for the generation of a given answer. First, we filter examples from existing benchmarks, discarding multiple attributions. Second, we extract structural elements from the document pages using a document layout analysis model. In parallel, we abstract existing answers using a MLLM. Then, we use

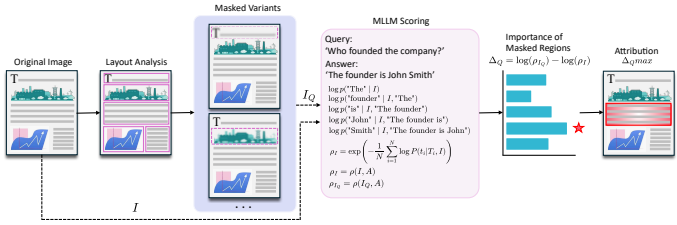


Figure 7.2: Illustration of the MAPPET visual attribution pipeline.

MAPPET to identify the most relevant structural elements with respect to the abstractive answer. Finally, a thresholding mechanism discards examples with low confidence scores. A graphical overview of our automatic annotation pipeline is provided in Figure 7.2.

Examples Filtering. Examples Filtering The first stage of our pipeline focuses on discarding low-quality samples. To ensure reliable attribution performance, we retain only QA examples linked to a single document page, discarding all instances associated with multiple or no pages.

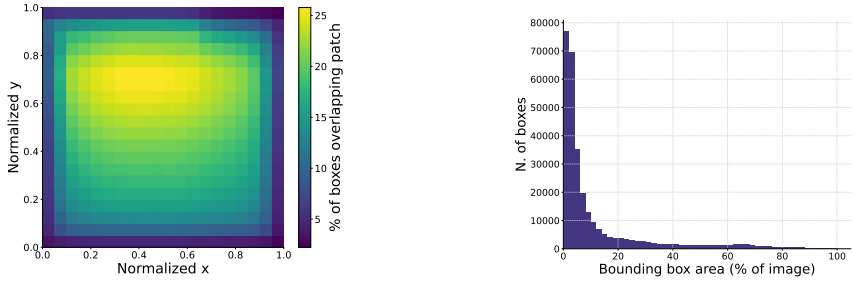
Document Layout Analysis. Accurate detection of structural elements is for our pipeline. To enable grounding through MAPPET, the dataset must provide consistently sized candidate layout regions. However, most benchmarks either lack explicit structural coordinates or exhibit inconsistent annotation granularity. To this end, we employ Docling [13], a state-of-the-art off-the-shelf document layout analysis module, that predicts both bounding boxes and the semantic categories of structural elements. We then normalize the output categories to ensure consistency across all datasets. Specifically, all text regions are merged into a unified *text* category, figures and charts are merged into the *picture* category, and tables are kept in the *table* category. Among the benchmarks, VisualMRC includes an *other* category containing otherwise unclassifiable examples. We retain this category to avoid introducing noise into the class taxonomy. Furthermore, in the VisualMRC datasets, where overlapping annotations are present for reducible elements (*e.g.* table cells and nested tables), we preserve only the outermost bounding box to maintain annotation coherence.

	DocVQA			VisualMRC			VISA			SlideVQA			VisualWebB			LongDocURL			MMLongB-Doc		
Model	Acc	Acc_F	%Filt	Acc	Acc_F	%Filt	Acc	Acc_F	%Filt	Acc	Acc_F	%Filt	Acc	Acc_F	%Filt	Acc	Acc_F	%Filt	Acc	Acc_F	%Filt
DeepSeek-7B	70.0	87.0	71.0	72.7	**94.1**	**94.1**	20.7	55.1	42.9	63.6	91.9	90.3	56.6	88.9	86.7	58.0	86.0	67.4	46.8	88.6	77.1
Qwen3-VL-8B	82.0	90.7	81.4	54.5	87.7	87.7	46.0	83.7	74.8	45.5	82.2	80.0	71.7	**96.9**	**92.2**	65.0	**95.8**	**79.2**	54.3	**95.2**	85.7
Qwen2.5-VL-7B (ext)	**95.0**	95.8	86.5	**84.8**	91.2	89	**72.5**	84.4	78.7	77.1	87.5	87.5	**82.8**	90.5	84.5	**83.0**	90.8	71.3	63.4	82.1	76.1
Qwen2.5-VL-7B (abs)	93.0	**98.9**	**88.0**	74.5	91.8	91.8	63.2	84.2	**79.4**	**80.6**	**93.6**	**92.3**	78.8	94.6	87.8	70.0	95.2	74.6	**66.0**	89.7	**86.2**
Qwen2.5-VL-32B	90.9	95.6	86.8	67.3	92.2	92.2	57.8	**88.1**	78.8	63.3	91.9	90.3	66.3	89.6	83.6	78.0	97.1	78.6	61.7	89.1	78.3

Table 7.3: MAPPET annotation quality. *Acc* denotes overall annotation accuracy; Acc_F is accuracy on MAPPET-filtered annotations; *%Filt* reports accuracy when hallucinated content is automatically marked incorrect.

Answer Abstraction. Concurrently with document layout analysis, we perform answer abstraction to ensure stylistic and semantic consistency across datasets. LLMs are inherently proficient in generating discursive answers rather than extractive ones. However, among the selected sources, most contain extractive answers, which are misaligned with the expressive style expected from LLMs. To harmonize these formats, we employ Qwen2.5-VL-7B [16] instructing it to abstract otherwise extractive answers while preserving their semantic content. The model selection criteria and the prompt template utilized for answer abstraction are available in the supplementary material. To assess the models ability to generate correct abstractive answers from the extractive counterpart, we sample 200 examples from the LongDocURL and DocVQA datasets and perform manual annotation. Three annotators were first instructed to identify semantic inconsistencies between the extractive and abstractive answers. Then, the annotators collectively re-evaluated and corrected their annotations to reach hunanimous agreement. In total, 4.5% of the generated answers were hallucinated, consisting mostly of slight titles and names modifications. Overall, only 1.5% of the generated answers were completely semantically dissimilar from the extractive counterparts.

Attribution. Our third stage is attribution, which locates the visual evidence supporting each answer. This step is pivotal for constructing a comprehensive dataset suitable for grounding evaluation. Specifically, for every example in the benchmark dataset and for each sentence in the abstracted answers, we use MAPPET to compute the attribution score of each structural element. The elements are ranked in descending order of relevance and the top-ranked element is selected as the attributed evidence source. In case of multi-sentence answers, each new sentence is conditioned on every token up to the previous answer sentence.



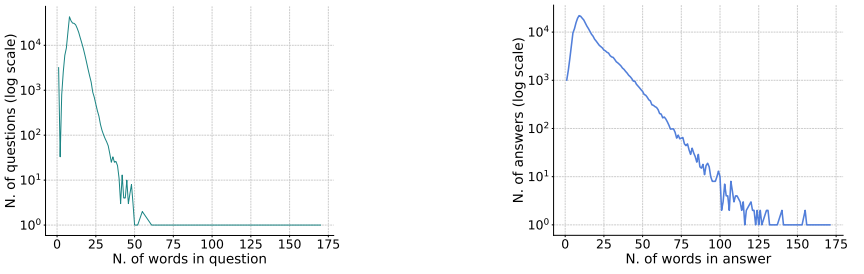
(a) (b)

Figure 7.3: Statistics for DocAttriBench. In order: distribution of images aspect ratios (height/width) for the train split (a), and test split(b).

Filtering. As final step, we filter the examples where the system fails to identify a reliable attribution. This filtering step addresses three primary cases. First, some answers might correspond to information redundantly appearing in multiple locations within the page, preventing unique grounding. Second, some sentences might depend on evidence scattered across several regions. Third, some answers might not be answerable with the document page. To handle these scenarios, we employ a two-steps filtering strategy. We first discard all samples whose top attribution score is lower than a fixed confidence threshold. Then, we remove cases where the score gap between the top-two ranked regions is smaller than a predefined margin. For benchmarks already containing the evidence source, we compute the region attribution score and retain the example if the score is greater than the threshold. This strategy ensures that the retained examples exhibit strong evidence alignment and are not ambiguous or attributable to multiple source. For all the above mentioned cases, we set the threshold to the same value ($\tau = 0.1$).

7.3.4 Dataset Details

The final dataset, DocAttriBench, automatically constructed through MAPPET, comprises 238,720 documents across diverse domains and 295,075 examples. Each example includes an image, a question, an answer, and annotated evidence regions covering text, tables, figures, and other visual elements. The dataset is split into training and test sets with no overlapping images across partitions. The large-scale training dataset enables fine-tuning of MLLMs for VAG, demonstrating

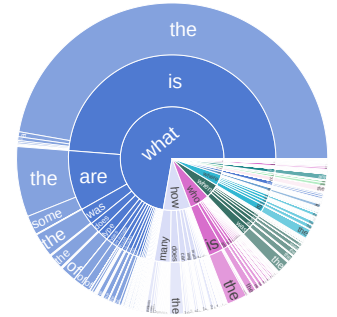


(a) (b)

Figure 7.4: Statistics for DocAttriBench. In order: word count distribution for questions (a) and answers (b).

the effectiveness of the proposed pipeline in generating high-quality supervision signals. The test set serves as a standardized benchmark to evaluate existing and future MLLMs with grounding capabilities on VAG.

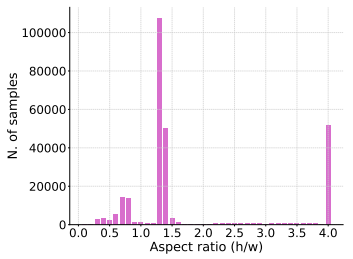
To provide a deeper understanding of the dataset characteristics, we include additional visualizations. Figure 7.4a and Figure 7.4b show the distribution of question and answer lengths, and the table in Figure 7.5b shows the corresponding average token counts. Further, in Figure 7.5a we summarize the most occurring questions grouping them according to their first three words, excluding rare occurrences. The distribution of the images aspect ratios is shown in Figure 7.5c and Figure 7.5d, showing local modes corresponding to slides (< 1), A4-like documents (~ 1.5), and rendered webpages (> 3). We additionally examine the spatial distribution and scale of the annotated regions. First, each image is divided into a 20×20 grid of non overlapping patches. Then, we count how many bounding boxes intersect with each patch. The resulting density map, along with the distribution of normalized region areas, is provided in Figure 7.3a and Figure 7.3b.



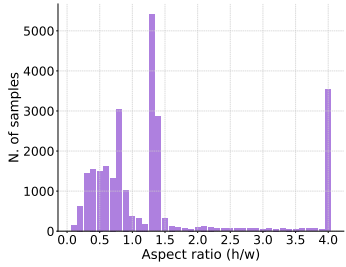
(a)

Split	Avg. Question Tokens	Avg. Answer Tokens
Train	12.34	18.36
Test	11.80	10.43

(b)



(c)



(d)

Figure 7.5: Statistics for DocAttriBench. In order: hierarchical distribution of questions by their first three words (a), average number of tokens per question and answer (b), heatmap of bounding-box coverage (c), and the histogram of bounding-box areas distribution (d).

7.4 Evaluation Protocol

Assessing grounding performance in MLLMs is inherently challenging due to the abstraction of free-form generation. Our evaluation protocol is designed to satisfy two key requirements: (i) it must enable the assessment of free-form text, and (ii) it must jointly evaluate both answer generation and grounding capabilities. To this ends, we design DocAttriBench protocol to assess MLLMs on three tasks: (i) grounded answer generation, where the model must produce an answer along with its visual grounding, (ii) post-hoc grounding, where the model must only localize the supporting evidence for a given answer, and (iii) answer-locating, where the model only responds with coordinates to a given query. All our prompt templates are available in the supplementary material.

7.4.1 Evaluation Metrics

Answer Accuracy. Current MLLMs are optimized for the generation of long-form, open-ended responses rather than concise, extractive answers. Consequently, our evaluation protocol takes into account the variability in response style and length across models. To this end, we design an answer evaluation protocol, inspired from MATHVISTA [129] and MMLongBenchDoc [135], repurposing the latter LLM-based answer extraction module to convert answers into a structured list of information units. To facilitate post-processing, we instruct the model to assign a semantic type to each extracted element. During evaluation, we align the spans from the ground-truth and generated answers by decreasing Average Normalized Levenshtein Similarity (ANLS) [24] for text spans, while we use exact match for numbers. Each text span pair is considered a match if the similarity score exceeds a fixed threshold ($\tau = 0.5$). Finally, we compute answer-level accuracy as the proportion of ground-truth spans that have a matched counterpart in the generated answer.

Box F1. MLLMs with grounding capabilities produce bounding boxes following diverse formatting conventions. Our evaluation protocol accounts for this with adaptable answer-generation prompts and model-specific parsing functions to accurately extract the predicted boxes. Ground truth and predicted regions are matched by descending order of Intersection over Union (IoU) and are considered correct matches if their IoU exceeds a fixed threshold ($\tau = 0.5$). Unmatched ground truth boxes are accounted as false negatives, while unmatched predictions

are false positives. Finally, we compute the F1 score to quantify grounding accuracy.

Overall Answer Correctness. To fully assess the correctness of the generated answers, we combine Answer Accuracy and BoxF1 into an overall answer-level correctness metric. For a given answer, the Overall Answer Correctness is 1 if both Answer Accuracy and BoxF1 exceed a threshold ($\tau = 0.5$).

7.4.2 Evaluating MLLMs on DocAttriBench

Zero-shot MLLMs

To establish a comprehensive zero-shot benchmark on our dataset, we evaluate sixteen representative MLLMs with localization capabilities. All inferences are performed through the VLLM library, an efficient inference engine based on Paged Attention [105].

Qwen Family. From the Qwen family, we include Qwen2.5-VL [16] (in the 3B, 7B, and 32B sizes) and Qwen3-VL [4] (in the 2B, 8B, and 32B variants). Precisely, we are able to fit the full sized models (provided in half precision for this model family) for the variants up to 8B. For the bigger variants, we utilize quantized versions. Specifically, we utilize 4 bits and 8 bits versions for Qwen2.5 and Qwen3 respectively. VLLM is set up with tensor parallel size equal to 2, a maximum number of batched tokens of 256, a maximum context length of 8,000 tokens, prefix caching, temperature of 0.0, and maximum GPU memory utilization of 0.8. All inferences were computed on a single A100 GPU with 64 GB of vRAM.

InternVL Family. From the InternVL series, we select InternVL2.5 [5], InternVL3 [255], and InternVL3.5 [217], each evaluated in 2B, 8B, and 38B variants. Due to unavailability of quantized versions of this models, they are deployed full size. VLLM settings are similar to the ones used for Qwen. However, due to the size of the larger 38B models, 2 A100 GPUs were required to shard the model weights over the GPUs.

VISA. VISA [134] is a document VAG oriented fine-tuned derivative of Qwen2-VL [216] fine-tuned on the omonim dataset. The version we selected was fine-tuned using LoRA starting from the 7B parameters version of Qwen2-VL. Up

Grounded Answer Generation Prompt Template

System: You are an agent excellent at identifying evidence in a document page.
Your job is to answer the user's query based on the provided image and the context.
When you answer, provide evidence bounding boxes in the format <box> x1 y1 x2 y2 </box>.
Add an evidence bounding box at the end of each generated sentence.

If you cannot find the answer, respond with "I don't know".

User: {query}

Assistant: {answer}

Figure 7.6: Grounded answer generation prompt template employed for fine-tuning and inference from the fine-tuned models.

to date, VISA is available model known to be fine-tuned for answer grounding. VISA was deployed using VLLM with the same settings utilized for the Qwen family.

7.4.3 Fine-tuning MLLMs

To assess the applicability of our dataset construction method for developing grounding capable MLLMs, we fine-tune Qwen2.5-VL-7B[1] and Qwen3-VL-8B[2] using LoRA [88] as implemented in the PEFT library [137]. We employ a modified versions of the original inference prompt templates, removing bounding-box dimensions instructions to let the model internalize the task structure through data exposure. The prompt templates used for finetuning are reported in Figure 7.6, Figure 7.7, and Figure 7.8. For each training sample, a task type is randomly sampled to ensure balanced task diversity.

To ensure robustness to image dynamic resolutions, we randomize the input image sizes. Specifically, we first assign a target size for the longest side of each image: 1024 pixels for images whose aspect ratio is at most 3:1 and 2048 pixels otherwise. We then added a random offset between 0 and 500 pixels to the base

[1] Qwen/Qwen2.5-VL-7B-Instruct
[2] Qwen/Qwen3-VL-8B-Instruct

Post-Hoc Grounding Prompt Template

System: You are an agent excellent at identifying evidence in a document page.
Your job is to identify the answer contained in the user's input based on the provided image and the context.
Respond by providing the evidence bounding boxes in the format <box> x1 y1 x2 y2 </box>.
The input is constituted by the user query and the answer to that query.

If you cannot find the answer, respond with the empty box <box> </box>.

User: Query:
{query}

Answer:
{answer}

Assistant: {bounding_box}

Figure 7.7: Post-hoc grounding prompt template employed for fine-tuning and inference from the fine-tuned models.

value. The image is resized ensuring that the longest side matches the resulting target length while preserving the aspect ratio. After this step. we apply each model internal resizing logic before feeding the image to the model.

For grounding supervision, bounding boxes are represented in absolute coordinates for Qwen2.5 and relative coordinates ([0, 1000]) for Qwen3, following their pretraining conventions. Training is conducted for 24 hours on two A100 GPUs with learning rate of 1e-4 and a batch size of 64. We name the models finetuned from Qwen2.5 and Qwen3 respectively MAPPET-7B and MAPPET-8B.

7.5 Experimental Results

We report experimental results to assess both benchmark quality of DocAttriBench and the attribution performance of current MLLMs. We begin with an analysis of MAPPET-generated annotations. Then, we analyze zero-shot results on the selected MLLMs. Finally, we demonstrate the benefits of MAPPET-based annotations showing the fine-tuning results. The human annotated dataset quality is shown in

Answer Localization Prompt Template

System: `You are an agent excellent at identifying evidence in a document page.`
`Your job is to locate the answer to the user's query based on the provided image and the context.`
`Respond by providing the evidence bounding boxes in the format <box> x1 y1 x2 y2 </box>.`

`If you cannot find the answer, respond with the empty box <box> </box>.`

User: {`query`}

Assistant: {`bounding_box`}

Figure 7.8: Answer localization prompt template employed for fine-tuning and inference from the fine-tuned models.

Table 7.3, while Table 7.4 and Table 7.5 show the results on DAB test sets.

7.5.1 Annotation Quality Evaluation

To validate the effectiveness of MAPPET in generating a grounding dataset, we conducted human evaluation to compare the annotations generated by MAPPET with the ones obtained through human effort. We begin by selecting open-source models. Specifically, we evaluate Qwen2.5-VL-7B, Qwen3-VL-7B, DeepSeek-7B and Qwen2.5-VL-32B. The first three models are chosen for their favorable trade-off between computational cost and inference speed, enabling large scale annotation without excessive resource requirements. The inclusion of the 32B variant allows us to examine whether scaling the underlying model yields higher-quality attributions. Empirically, we observe that the 32B model provides inconsistent improvements and we refrain from testing even larger models in our study.

Next, we randomly sample 100 examples from each source benchmark (including 100 from PaperVISA and 100 from WikiVISA) and annotate grounding regions. Three annotators were instructed to identify the evidence boxes for each sampled example and to annotate any form of hallucination in the dataset original answer. In case alternative boxes are available, we retain both. Table 7.3 reports the annotation quality metrics. Annotation accuracy (Acc) measures the portion of correct machine-generated attributions. Across benchmarks, Acc ranges from

	DocVQA			VisualMRC			VISA			SlideVQA			VisualWebB			LongDocURL			MMLongB-Doc			
Model	Acc_{txt}	$F1_{box}$	Acc	Acc_{txt}	$F1_{box}$	Acc	Acc_{txt}	$F1_{box}$	Acc	Acc_{txt}	$F1_{box}$	Acc	Acc_{txt}	$F1_{box}$	Acc	Acc_{txt}	$F1_{box}$	Acc	Acc_{txt}	$F1_{box}$	Acc	**Avg**
Zero-shot MLLMs																						
InternVL2.5-2B	37.5	0.0	0.0	30.0	1.0	0.1	6.6	0.0	0.0	29.6	0.7	0.0	33.5	0.5	0.4	20.1	0.2	0.0	25.7	0.4	0.0	0.1
InternVL3-2B	41.9	4.9	0.4	25.9	8.2	2.1	10.4	0.0	0.0	14.9	23.2	1.6	37.3	3.5	1.3	16.7	4.7	0.0	15.8	8.5	0.7	0.7
InternVL3.5-2B	49.0	19.3	11.7	23.1	2.0	0.9	2.3	0.4	0.0	39.6	0.6	0.3	18.4	9.8	3.9	22.2	1.0	0.6	18.8	1.5	0.7	2.1
Qwen2.5-VL-3B	12.4	4.3	0.2	34.3	11.5	2.7	2.7	0.1	0.0	20.6	17.6	3.1	6.0	7.1	0.4	12.7	6.1	0.6	11.8	11.6	0.0	0.9
Qwen3-VL-2B	29.9	6.8	6.3	34.7	17.0	9.1	27.8	0.4	0.1	29.9	10.4	7.2	33.5	5.1	3.4	14.4	7.2	2.8	15.8	8.1	5.2	4.3
InternVL2.5-8B	59.0	0.0	0.0	65.7	2.8	1.2	36.0	0.0	0.0	64.8	2.8	1.7	60.9	0.9	0.4	36.6	0.0	0.0	45.6	1.2	0.7	0.5
InternVL3-8B	64.9	6.4	3.2	63.9	8.2	4.2	21.9	0.0	0.0	61.9	16.5	9.3	53.2	1.9	1.3	40.7	3.9	1.9	44.1	5.4	1.8	2.6
InternVL3.5-8B	70.3	22.0	16.9	68.1	5.4	2.9	37.6	0.4	0.2	70.5	4.0	2.2	64.0	25.0	17.6	45.5	3.2	1.6	58.5	2.1	1.5	5.3
Qwen2.5-VL-7B	67.0	7.3	2.6	65.6	10.0	2.7	52.5	0.6	0.0	65.1	17.8	8.1	61.4	7.7	3.0	44.6	10.7	3.7	51.5	12.5	3.7	3.0
Qwen3-VL-8B	67.7	14.6	10.8	67.8	24.4	14.3	53.1	4.2	1.9	70.1	9.4	7.0	63.1	16.8	8.6	43.1	19.8	7.0	52.2	14.9	7.7	9.4
InternVL2.5-38B	64.2	1.0	0.4	65.9	6.9	3.8	5.8	0.3	0.1	63.0	10.0	6.3	63.1	2.8	1.7	40.1	3.0	1.3	48.9	6.0	3.7	2.3
InternVL3-38B	59.0	4.6	3.2	66.9	6.4	3.9	44.9	1.8	0.4	66.9	11.4	8.3	60.1	2.1	1.7	38.4	4.0	2.2	55.1	6.8	4.4	3.2
InternVL3.5-38B	69.0	26.4	20.4	67.2	14.9	8.6	42.4	1.4	0.2	69.2	6.2	4.3	**65.2**	30.0	20.2	42.4	11.5	4.9	56.2	7.6	5.2	8.6
Qwen2.5-VL-32B	68.1	9.7	7.8	67.3	19.6	11.6	56.4	1.4	0.6	69.5	11.5	8.9	64.0	28.4	19.3	45.9	9.8	5.8	56.6	14.8	11.8	9.1
Qwen3-VL-32B	**72.9**	41.8	29.3	**68.2**	56.7	36.6	**56.5**	31.0	18.3	**73.2**	30.6	22.4	63.5	31.2	21.0	**51.6**	47.4	24.9	**61.8**	44.3	**29.0**	25.4
Source Attribution MLLMs																						
VISA-7B	60.5	43.2	22.1	62.6	64.3	36.6	52.2	**50.4**	**27.1**	62.2	62.6	34.1	53.6	14.5	8.6	44.1	43.0	16.4	46.0	55.9	21.3	24.3
MAPPET-7B	57.0	66.2	38.6	63.4	75.1	45.8	42.2	43.3	22.4	59.1	**67.8**	**37.9**	56.6	33.7	19.7	46.4	**59.9**	**28.7**	44.5	**61.8**	26.1	30.2
MAPPET-8B	61.8	**70.5**	**43.2**	64.5	**76.1**	**47.7**	37.4	43.2	22.3	61.1	69.0	39.9	59.7	**36.7**	**24.5**	43.1	57.1	27.0	39.7	55.3	23.5	**31.6**

Table 7.4: Evaluation of selected models on DocAttriBench for grounded answer generation. Acc denotes overall annotation accuracy, Acc_{txt} is the answer accuracy, $F1_{box}$ reports the grounding F1 score, and Acc is the overall answer accuracy.

20% to 95% due to heterogeneous underlying models. After applying MAPPET filtering mechanism, we recompute accuracy on the retained subsets (Acc_F) and observe consistent increases (sometimes exceeding 95%), highlighting MAPPET ability to effectively discard noisy and unreliable annotations. Finally, we treat all hallucinated examples as incorrect, even when the predicted region is semantically consistent with the provided answer, and compute the final accuracy (%Filt). This evaluation produces a slight decrease in performance, but it is necessary to highlight the amount of noise present in existing benchmarks. Overall, these results confirm that MAPPET yields high-quality attributions and that its filtering step plays an important role in enhancing dataset reliability.

For Qwen2.5, we further ablate the answer abstraction module by replacing the abstractive answers with an extractive counterpart. The results show comparable performance, with a slight advantage of extractive answers. In contrast, when filtering is employed, the abstractive answers consistently yield higher attribution accuracy, indicating that the abstraction process provides more informative textual signals for MAPPET filtering mechanism.

Visual Answer Grounding Benchmarking

Model	DocVQA		VisualMRC		VISA		SlideVQA		VisualWebB		LongDocURL		MMLongB-Doc			
	$F1^{Q}_{box}$	$F1^{QA}_{box}$	$F1^{Q}_{box}$	$F1^{QA}_{box}$	$F1^{Q}_{box}$	$F1^{QA}_{box}$	$F1^{Q}_{box}$	$F1^{QA}_{box}$	$F1^{Q}_{box}$	$F1^{QA}_{box}$	$F1^{Q}_{box}$	$F1^{QA}_{box}$	$F1^{Q}_{box}$	$F1^{QA}_{box}$	**Avg^{Q}**	**Avg^{QA}**
Zero-shot MLLMs																
InternVL2.5-2B	0.0	0.2	1.1	0.4	0.2	0.0	0.6	0.3	0.0	0.0	0.2	0.2	0.5	0.0	0.4	0.2
InternVL3-2B	8.0	15.7	12.2	13.1	2.3	4.5	13.8	21.0	4.3	6.4	3.4	5.6	6.6	12.0	7.2	11.2
InternVL3.5-2B	11.1	13.8	0.8	2.0	0.2	0.2	0.2	0.3	4.8	11.7	0.2	1.2	1.2	2.5	2.6	4.5
Qwen2.5-VL-3B	6.3	4.5	14.7	14.4	6.0	7.2	25.4	22.6	8.1	8.2	8.2	11.0	13.5	17.3	11.7	12.2
Qwen3-VL-2B	28.2	26.4	27.7	30.4	16.5	11.4	30.9	32.7	15.9	16.5	28.0	26.4	29.6	32.5	25.3	26.1
InternVL2.5-8B	0.22	0.6	7.5	7.7	0.7	1.0	4.0	4.2	0.9	0.9	0.6	1.3	1.1	2.9	2.1	2.7
InternVL3-8B	14.6	15.2	9.9	11.1	3.6	1.8	23.2	16.0	8.4	11.6	8.0	5.7	10.3	8.1	11.1	9.9
InternVL3.5-8B	29.3	44.7	10.1	9.7	3.8	6.1	17.5	15.1	24.9	27.7	9.8	11.4	11.9	14.0	15.3	18.4
Qwen2.5-VL-7B	16.8	19.8	22.1	25.9	10.6	10.9	25.8	32.0	13.0	17.6	16.8	20.4	20.1	27.6	17.9	22.0
Qwen3-VL-8B	17.6	27.4	31.4	41.3	21.3	24.4	12.4	18.7	15.9	24.4	26.5	35.5	22.8	31.8	21.1	29.1
InternVL2.5-38B	10.7	11.4	16.3	15.6	6.8	6.3	27.7	27.8	6.4	6.8	12.8	13.5	29.5	27.0	15.8	15.5
InternVL3-38B	19.3	17.8	16.4	14.7	9.1	8.3	37.8	33.3	3.9	4.3	16.9	17.5	26.6	27.2	18.6	17.6
InternVL3.5-38B	28.6	36.2	25.1	23.2	4.7	4.2	10.4	11.9	28.0	30.7	14.0	13.4	12.1	13.9	17.6	19.1
Qwen2.5-VL-32B	14.8	19.1	27.7	34.6	5.12	7.2	20.4	27.4	26.9	28.8	17.9	29.2	22.4	32.4	19.3	25.5
Qwen3-VL-32B	47.5	51.0	61.0	64.6	38.9	42.7	35.1	39.5	28.2	30.7	44.2	46.9	54.6	51.9	44.2	46.8
Source Attribution MLLMs																
VISA-7B	41.3	44.5	65.4	65.8	**66.2**	**73.5**	62.7	62.8	13.7	14.5	42.7	46.2	**55.1**	**56.5**	49.6	52.0
MAPPET-7B	**57.0**	61.9	**68.7**	**69.8**	40.3	42.0	**67.2**	**68.7**	**26.5**	33.7	**43.2**	**43.8**	52.4	54.8	**50.8**	53.5
MAPPET-8B	54.7	**62.3**	64.0	72.0	31.9	38.4	62.5	68.5	25.6	**42.3**	32.9	42.0	40.0	53.0	44.5	**54.1**

Table 7.5: Evaluation of selected models on DocAttriBench for the answer locating and post-hoc attribution tasks. $F1^{Q}_{box}$ is the answer locating F1 score; $F1^{QA}_{box}$ is the post-hoc F1 score.

Zero-shot Models. Table 7.4 summarizes the zero-shot performance of sixteen representative MLLMs, evaluated on the seven public benchmarks constituting the test set of DocAttriBench. Results are reported in term of answer accuracy (Acc_{txt}), Box F1 ($F1_{box}$), and overall answer correctness (Acc). Table 7.5 focuses on the answer-localization and post-hoc attribution results reporting of grounding F1 scores.

Overall, the latest models from both families achieve the strongest results. However, $F1_{box}$ scores remain consistently low, averaging below 40% for >30B parameter models, below 20% for 7-8B parameters models, and below 10% for the smallest 2-3B parameters models. This highlight how even the best-performing models fail to reliably localize the visual evidence supporting their answers.

The complementary analysis in Table 7.5 supports these findings. When evaluated solely on box generation, models exhibit slightly higher localization precision, even if they failed when asked to generate the answer and evidence location jointly. The best-performing model, Qwen3-VL-32B, achieves almost 47 $F1_{box}$ in post-hoc mode on average, nearly 8 points over the grounded generation setup, while maintaining stable textual accuracy. Finally, in answer-locating mode most models exhibit lower results, even though they exceed $F1_{box}$ from the grounded answer generation task. This highlights the intrinsic difficulty of

end-to-end grounded reasoning, which requires the model to both reason and justify its output within a single forward pass.

Source Attribution Models. We the evaluate models explicitly designed for grounding, including VISA-7B and the fine-tuned versions of Qwen2.5 and Qwen3. As shown in Table 7.4, our models outperform general-purpose MLLMs in visual grounding. In general, models explicitly exposed to grounded data are superior on VAG, surpassing zero-shot counterparts on grounding and average metrics. Moreover, fine-tuning improves metrics with respect to the more powerful 32B and 38B models, even though generally good results must be highlighted from Qwen3-32B. The fine-tuned models achieve good results and an overall answer correctness score greater than 30, almost 5 points above the best alternative without exposure to grounding data. On answer-locating and box-answering tasks, the fine-tuned models exhibit substantial gains over their zero-shot counterparts, with average improvements exceeding 25 points. Notably, MAPPET-7B and MAPPET-8B achieve the highest scores among all evaluated models surpassing large general-purpose MLLMs. These results suggest that expose to DocAttriBench supervision meaningfully enhances grounding quality in MLLMs.

Finally, the textual accuracy (Acc_{txt}) is generally lower on fine-tuned models with respect to zero-shot counterparts. Inspection of the generated answers from both model types reveal two significant differences between our training dataset and the answers generated by zero-shot Qwen2.5 and Qwen3 models. First, our training dataset consists in short, concise, and reasoning-free answers, while zero-shot models have a tendency to produce short reasoning steps, guiding the answer toward the correct values. Second, the reasoning steps can include the precise location of the correct answer value, while our dataset lacks this information and forces the tuned models to avoid the respective step too.

Boxes Precision and Recall. The grounding metrics previously reported focus exclusively on F1 scores. This might raise concerns about models tendencies to generate multiple boxes, lowering precision scores. To clarify this, we report precision and recall for each task in Table 7.6, Table 7.7, and Table 7.8. The numerical results highlight a consistent pattern for Qwen3-VL-2B: the model frequently generates multiple boxes for the grounded answer generation task, inflating false positives and leading to low precision. Qualitative results underscore that these extra boxes often do not correspond to meaningful reasoning, instead, the model produces several regions without an actual answer. This behavior largely

explains the poor performance observed for this model. In contrast, we do not observe substantial drops in either precision or recall for the other models. The low F1 scores reflect challenges in identifying the correct evidence rather than systematic over-generation or a breakdown of the reasoning process.

	DocVQA		**VisualMRC**		**VISA**		**SlideVQA**		**VisualWebB**		**LongDocURL**		**MMLongB-Doc**			
Model	P_{box}	R_{box}	P_{box}	R_{box}	P_{box}	R_{box}	P_{box}	R_{box}	P_{box}	R_{box}	P_{box}	R_{box}	P_{box}	R_{box}	$\mathbf{Avg_P}$	$\mathbf{Avg_R}$
Zero-shot MLLMs																
InternVL2.5-2B	0.0	0.0	1.2	0.8	0.0	0.0	0.7	0.7	0.5	0.4	0.2	0.1	0.4	0.4	0.4	0.4
InternVL3-2B	6.0	4.1	14.1	5.8	0.0	0.0	26.0	21.0	4.2	3.0	6.3	3.7	9.4	7.7	9.4	6.5
InternVL3.5-2B	28.0	14.8	3.5	1.4	2.2	0.2	0.7	0.6	28.0	5.9	1.2	0.9	2.2	1.1	9.4	3.6
Qwen2.5-VL-3B	5.0	3.7	10.9	12.2	0.1	0.1	17.3	17.9	8.0	6.4	6.6	5.6	12.7	10.7	8.7	8.1
Qwen3-VL-2B	4.1	20.6	11.7	31.3	0.3	0.5	6.3	29.8	3.1	16.1	4.2	25.0	4.7	27.6	4.9	21.5
InternVL2.5-8B	0.0	0.0	4.0	2.2	0.0	0.0	3.1	2.5	1.1	0.8	0.0	0.0	1.3	1.1	1.4	0.9
InternVL3-8B	6.8	6.1	9.9	7.0	0.0	0.0	19.5	14.3	2.3	1.7	4.7	3.3	7.1	4.4	7.2	5.2
InternVL3.5-8B	22.5	21.5	6.0	5.0	0.4	0.4	4.3	3.7	25.0	25.0	3.3	3.1	2.0	2.2	9.1	8.7
Qwen2.5-VL-7B	15.6	4.8	26.6	6.2	2.0	0.3	30.6	12.6	15.8	5.1	16.8	7.8	19.4	9.2	18.1	6.6
Qwen3-VL-8B	14.8	14.3	23.7	25.1	4.1	4.3	8.6	10.4	16.7	16.9	18.2	21.8	13.6	16.5	14.3	15.6
InternVL2.5-38B	2.3	0.7	7.2	6.5	2.2	0.2	11.1	9.2	4.2	2.1	3.1	3.0	6.6	5.5	5.3	3.9
InternVL3-38B	4.7	4.6	6.3	6.5	1.8	1.7	10.7	12.2	2.1	2.1	3.6	4.6	6.1	7.7	5.0	5.6
InternVL3.5-38B	26.7	26.0	15.0	14.9	1.4	1.4	6.5	6.0	30.3	29.7	11.4	11.7	7.5	7.7	14.1	13.9
Qwen2.5-VL-32B	9.7	9.8	18.5	20.8	1.2	1.5	9.8	13.8	27.2	29.7	8.4	11.7	12.8	17.6	12.5	15.0
Qwen3-VL-32B	41.5	42.1	55.0	58.5	29.8	32.3	29.3	31.9	31.5	30.9	44.2	51.0	42.8	46.0	39.2	41.8
Source Attribution MLLMs																
VISA-7B	43.2	43.2	64.4	64.2	**50.5**	**50.4**	62.7	62.6	14.6	14.4	43.0	43.0	55.9	55.9	47.7	47.7
MAPPET-7B	66.3	66.2	75.2	75.0	43.4	43.3	67.9	67.8	33.9	33.5	**60.0**	**59.9**	**61.8**	**61.8**	58.3	**58.2**
MAPPET-8B	**70.7**	**70.3**	**76.5**	**75.8**	43.7	42.8	**69.2**	**68.8**	**36.9**	**36.4**	57.5	56.7	55.8	54.8	**58.6**	57.9

Table 7.6: Precision (P) and recall (R) of the bounding boxes generated for the grounded answer generation task.

Model	DocVQA		VisualMRC		VISA		SlideVQA		VisualWebB		LongDocURL		MMLongB-Doc			
	P^{QA}_{box}	R^{QA}_{box}	P^{QA}_{box}	R^{QA}_{box}	P^{QA}_{box}	R^{QA}_{box}	P^{QA}_{box}	R^{QA}_{box}	P^{QA}_{box}	R^{QA}_{box}	P^{QA}_{box}	R^{QA}_{box}	P^{QA}_{box}	R^{QA}_{box}	$\mathbf{Avg}^{QA}_{P}$	$\mathbf{Avg}^{QA}_{R}$
Zero-shot MLLMs																
InternVL2.5-2B	0.2	0.2	0.4	0.4	0.0	0.0	0.3	0.2	0.0	0.0	0.2	0.1	0.0	0.0	0.2	0.1
InternVL3-2B	15.7	15.6	13.2	13.1	0.0	0.0	21.2	20.9	6.5	6.4	5.6	5.5	12.0	12.0	10.6	10.5
InternVL3.5-2B	16.4	12.0	2.4	1.6	0.3	0.3	0.3	0.2	12.5	11.0	1.2	1.2	2.5	2.5	5.1	4.1
Qwen2.5-VL-3B	6.0	3.6	15.8	13.2	0.4	0.3	26.8	19.5	10.3	6.8	13.8	9.2	20.8	14.9	13.4	9.6
Qwen3-VL-2B	25.8	27.0	30.3	30.4	1.3	1.3	31.1	34.4	16.1	16.9	26.2	26.6	29.3	36.6	22.9	24.8
InternVL2.5-8B	0.6	0.6	7.7	7.7	0.1	0.1	4.2	4.2	0.9	0.8	1.3	1.3	2.9	2.9	2.5	2.5
InternVL3-8B	17.1	13.7	11.8	10.4	0.2	0.2	17.3	14.9	12.8	10.6	6.3	5.2	9.1	7.2	10.6	8.9
InternVL3.5-8B	44.8	44.5	9.6	9.7	1.7	1.7	15.0	15.2	27.8	27.5	11.3	11.5	13.8	14.1	17.7	17.8
Qwen2.5-VL-7B	19.9	19.7	26.2	25.5	1.1	1.1	31.9	32.1	17.8	17.4	20.4	20.4	27.7	27.5	20.7	20.5
Qwen3-VL-8B	27.7	27.2	41.4	41.1	6.8	6.7	18.9	18.6	24.6	24.1	35.9	35.2	32.0	31.5	26.7	26.3
InternVL2.5-38B	11.3	11.3	15.6	15.6	1.4	1.4	27.9	27.6	6.9	6.8	13.5	13.5	27.1	26.8	14.8	14.7
InternVL3-38B	17.8	17.8	14.7	14.7	3.2	3.2	33.3	33.2	4.3	4.2	17.5	17.6	27.2	27.2	16.8	16.8
InternVL3.5-38B	37.9	34.7	24.3	22.2	5.1	4.0	14.0	10.3	32.2	29.2	18.5	10.5	18.2	11.2	21.5	17.5
Qwen2.5-VL-32B	19.1	19.1	34.5	34.7	4.4	4.4	27.1	27.6	29.1	28.4	28.8	29.5	32.1	32.6	25.0	25.2
Qwen3-VL-32B	51.3	50.8	64.9	64.3	35.0	34.2	40.0	39.0	31.8	29.7	**47.2**	**46.7**	52.0	51.8	46.0	45.2
Source Attribution MLLMs																
VISA-7B	44.5	44.5	65.8	65.7	**58.6**	**58.6**	62.9	62.7	14.6	14.4	46.2	46.2	**56.5**	**56.5**	49.9	49.8
MAPPET-7B	61.9	61.9	69.8	69.7	32.8	32.8	**68.8**	**68.6**	33.9	33.5	43.8	43.8	54.9	54.7	52.3	52.1
MAPPET-8B	**62.3**	**62.3**	**72.1**	**72.0**	31.2	31.2	68.6	68.4	**42.7**	**42.0**	42.1	42.0	53.1	52.9	**53.2**	**53.0**

Table 7.7: Precision (P) and recall (R) of the bounding boxes generated for the post-hoc attribution task.

Model	DocVQA		VisualMRC		VISA		SlideVQA		VisualWebB		LongDocURL		MMLongB-Doc			
	P^{Q}_{box}	R^{Q}_{box}	P^{Q}_{box}	R^{Q}_{box}	P^{Q}_{box}	R^{Q}_{box}	P^{Q}_{box}	R^{Q}_{box}	P^{Q}_{box}	R^{Q}_{box}	P^{Q}_{box}	R^{Q}_{box}	P^{Q}_{box}	R^{Q}_{box}	$\mathbf{Avg}^{Q}_{P}$	$\mathbf{Avg}^{Q}_{R}$
Zero-shot MLLMs																
InternVL2.5-2B	0.0	0.0	1.5	0.9	0.2	0.2	0.7	0.5	0.0	0.0	0.3	0.1	0.6	0.4	0.5	0.3
InternVL3-2B	8.1	7.9	12.4	12.1	0.0	0.0	14.0	13.7	4.3	4.2	3.5	3.3	6.6	6.5	7.0	6.8
InternVL3.5-2B	13.6	9.4	1.0	0.7	0.5	0.2	0.2	0.2	6.3	3.8	0.2	0.1	1.2	1.1	3.3	2.2
Qwen2.5-VL-3B	10.3	4.5	16.1	13.6	0.1	0.1	33.8	20.3	10.0	6.8	12.0	6.2	18.9	10.5	14.5	8.9
Qwen3-VL-2B	28.4	28.1	27.7	27.7	1.3	1.3	28.9	33.3	16.0	15.7	27.5	28.5	26.0	34.4	22.3	24.1
InternVL2.5-8B	0.2	0.2	7.6	7.4	0.3	0.2	4.1	4.0	0.9	0.8	0.6	0.6	1.2	1.1	2.1	2.0
InternVL3-8B	16.1	13.3	11.8	8.6	1.0	0.1	28.4	19.6	10.9	6.8	9.5	7.0	12.6	8.7	12.9	9.1
InternVL3.5-8B	29.8	28.7	10.2	9.9	2.0	1.9	17.6	17.5	25.2	24.6	9.8	9.7	12.3	11.6	15.3	14.8
Qwen2.5-VL-7B	17.0	16.7	22.5	21.6	1.2	1.1	25.9	25.7	13.3	12.7	16.6	17.0	20.2	19.9	16.7	16.4
Qwen3-VL-8B 8.6	17.8	17.3	31.8	31.1	5.8	5.7	12.6	12.2	16.0	15.7	27.4	25.6	23.1	22.5	19.2	18.6
InternVL2.5-38B	10.7	10.7	16.4	16.3	1.2	1.1	27.8	27.6	6.4	6.4	12.8	12.8	29.7	29.4	15.0	14.9
InternVL3-38B	19.4	19.3	16.5	16.4	3.1	3.1	38.0	37.5	3.9	3.8	17.0	16.9	26.7	26.4	17.8	17.6
InternVL3.5-38B	28.9	28.3	25.7	24.5	4.4	4.1	11.2	9.7	28.4	**27.5**	15.3	12.9	13.2	11.2	18.1	16.9
Qwen2.5-VL-32B	15.0	14.6	27.7	27.6	3.9	3.9	20.4	20.3	27.2	26.7	18.1	17.7	22.8	22.1	19.3	19.0
Qwen3-VL-32B	47.8	47.1	61.4	60.7	26.2	25.6	35.8	34.4	**28.9**	**27.5**	**44.7**	**43.8**	55.0	54.4	42.8	41.9
Source Attribution MLLMs																
VISA-7B	41.3	41.3	65.5	65.3	**49.8**	**49.8**	62.8	62.6	13.7	13.6	42.7	42.7	**55.1**	**55.1**	47.3	47.2
MAPPET-7B	**57.0**	**57.0**	**68.8**	**68.6**	28.1	28.1	**67.3**	**67.1**	26.7	26.3	43.2	43.2	52.5	52.2	**49.1**	**48.9**
MAPPET-8B	54.8	54.6	64.2	63.9	27.9	27.8	62.7	62.3	27.3	24.1	33.0	32.9	40.1	39.9	44.3	43.6

Table 7.8: Precision (P) and recall (R) of the bounding boxes generated for the answer-locating task.

7.6 Limitations

We benchmarked MAPPET against the VISA framework. The VISA model employs a LoRA-based adaptation strategy with efficient module selection and was fine-tuned on a corpus of 300k instances for two epochs. Conversely, when tuning the MAPPET models, we adopted a different architectural intervention. Specifically, we were forced to reduce training times by relying about 50k examples for a single epoch and we relied on tuning both the model Transformer modules and the weight of the Feed Forward Network predicting token probabilities. As a consequence, the lower answer accuracy metrics were expected.

While answer and ROI accuracy are of importance, so are the intermediate reasoning steps. We acknowledge that most of the models utilized in this study rely on automatic intermediate reasoning steps to provide high quality answers. Our approach doesn't account for reasoning. This design choice is not a limitation of the proposed automatic dataset generation approach, rather it is a limitation dictated by the current available data, that we used as guidance. Further experiments should concentrate on strategies to improve the reasoning steps while performing VAG.

7.7 Conclusions

In this work, we introduced DocAttriBench (DAB), the first large-scale benchmark dataset designed to evaluate visual answer grounding in Document VQA, and proposed MAPPET, a scalable Mask-based Perplexity-Derived Attribution framework for automatic, fine-grained source annotation. By leveraging the MAPPET framework we unified multiple existing document understanding benchmarks, creating a dataset that enables both reliable evaluation and effective fine-tuning of grounding-capable multimodal models. Our experiments reveals that state-of-the-art MLLMs achieve strong textual accuracy but struggle to localize the supporting evidence on document pages. Fine-tuning on DocAttriBench substantially narrows this gap, improving overall grounding correctness across models and tasks.

We expect DocAttriBench to serve as a foundation for verifiable, interpretable document understanding and to foster models with explicit visual grounding.

Chapter 8

Conclusions

The goal of the research activities we carried out during my PhD period was strictly related to the Altilia clients' needs. Specifically, we focused on explainable and multi-modal document understanding and summarization systems. This goal has been tackled by following two main research directions which motivate the work presented in the previous chapters.

The first is the generation of summaries from multi-modal sources, which motivated a big part of my studies and the majority of my first works. We started with the study of various approaches and techniques to transform multi-modal inputs in machine readable format, in order to simplify summarization and information access through Altilia's already established means. Following, was the evaluation of LLMs generic and aspect-based summarization capabilities, utilizing open-source and proprietary counterparts as well as zero-shot, few-shot, and fine-tuning techniques. This was particularly relevant, leading us to a simple and effective design choice for our multi-modal summarization algorithms.

The second track, dealt with explainability in LLM-generated outputs, which motivated the last two chapters of this thesis. First, we delved into text-based attribution, providing a mean to solve answer-level attribution effectively and at scale. Moreover, through cross-encoders and dynamic chunking strategies, we designed a model agnostic solution. This enables researchers and engineers to swiftly change the underlying LLM in a RAG pipeline without the need to re-engineer prompts and attribution pipelines. Finally, our visual-grounding work represents the seminal work for the creation of visual grounding datasets at scale, without the need for expensive human supervision.

In the following sections, we summarize our contributions and draw the final conclusions of the results achieved so far.

Multi-Modal Summarization

Our preliminary investigation into the individual components of multi-modal summarization provided critical insights into the capabilities and limitations of current models. First, we demonstrated that Large Language Models possess impressive zero-shot capabilities for the interpretation of tabular data and we found that fine-tuning smaller models can bridge the performance gap with larger proprietary models. However, from an operational standpoint, open-source LLMs are often not advantageous in terms of monetary costs. Then, our systematic evaluation of Chart-to-Table models revealed significant discrepancies between benchmark and real-world performance. While larger model exhibit decent capabilities for standardized datasets, they suffer from data translation errors and extrinsic hallucinations when applied to complex real-world documents. Consequently, we concluded that C2T technologies are not yet mature enough for strict financial reporting, leading to their exclusion from the summarization pipeline. Finally, we observed divergencies between syntactic and semantic quality in text summarization, with fine-tuned models dominating overlap-based metrics and zero-shot LLMs demonstrating superior performance in factuality metrics. We concluded that prompt engineering offers a better balance of stability and quality compared to complex reasoning chains. Even few-shot learning approaches were found to suffer when provided examples are heterogeneous and thus were deemed unnecessary for the summarization task.

For multi-modal summarization, we proposed a modular pipeline for processing financial documents. We found that structure inputs can impact summary quality and significantly change the model's understanding of the provided document. Due to the limitations of generative approaches for charts, we implemented an extractive visual strategy to include multi-modal elements in the generated synopses, ensuring that elements are semantically relevant and provide novel information rather than redundant data. Regardless of the advances, we were fundamentally limited by the unavailability of high quality data and pre-trained open-source models to deal with complex documents. Finally, the imposed restriction of not utilizing RAG technologies was a fundamental limit, defining our design choices.

LLM-Based Evaluation and Context-Attribution

Motivated by the fundamental challenges of ensuring reliability of Retrieval Augmented Generation systems, we implemented an end-to-end pipeline to test LLM-based evaluation of LLM-generated answers. We addressed the critical issue of assessing answer quality with and without ground truth data. To do so, we investigated recently proposed LLM-based metrics on both public narrative datasets and proprietary financial data, revealing current limitations of LLM-based reference-free systems. Our findings show that there persist a significant reference-dependency gap, with reference-based metrics correlating strongly with human judgment but reference-free metrics remaining significantly less reliable. Moreover, performance degrades significantly when moving from general English corpora to highly specialized Italian financial data, suggesting that LLMs reasoning abilities are not uniformly distributed across languages and domains. While automated frameworks are evolving, they currently function as assistive tools rather than autonomous agents for human verification.

To address the need for verifiable AI, we investigated post-generation context-attribution strategies, comparing LLMs against cross-encoder architectures. Our experiments showed that open-source and proprietary LLMs tend to under-perform in post-generation context-attribution environments, achieving less than satisfactory performance. Large Language Model performance can be drastically improved through reasoning techniques. However, this comes with higher costs and longer inference times.

Alternatively, cross-encoder architectures were found to obtain results similar to those achieved by proprietary LLMs equipped with reasoning. Our findings indicate that cross-encoders might have issues with context length and long dependencies. This issue is automatically avoided through LLMs with reasoning, seemingly dealing with long dependencies. As far as cross-encoder are concerned, carefully tuning the context length drastically improves performance. Finally, synthetic data generation was found to be a viable solution to further reduce the gap between cross-encoders and large-scale LLM with reasoning, sometimes surpassing the latter.

Overall, cross-encoders were found to be a viable solution requiring minimal to no engineering for deployment, with cost advantages of up to $30\times$ on full utilization. Moreover, automatically deploying a cross-encoder architecture can be viewed as a model-agnostic solution, allowing researchers and engineers to how-switch LLMs in production without the need to re-engineer the prompt template and without loss of attributing performance. The solution is not data-agnostic

though, always requiring at least some amount of non-annotated data for synthetic data generation and tuning.

Visual Grounding

We addressed Visual Answer Grounding (VAG) in document Visual Question Answering, where models must correctly answer a given query and precisely localize the evidence in support to their response. Our approach involved the creation of a model-agnostic VAG algorithm, solely dependent on the model ability to answer a given query. Mask-based Perplexity-Derived Attribution (MAPPET) is a novel framework based on a perplexity reparameterization. The approach requires an off-the-shelf DLA module utilized as a region proposal algorithm. With the increasing capabilities of document parsing models and the constantly reduced dimensions of such modules, MAPPET can be utilized at inference time to identify the attributing sources.

Due to the computational overhead, we utilized our algorithm to produce a synthetic dataset at scale. Repurposing existing literature datasets, we created DocAttriBench (DAB), a large-scale dataset with almost 300k visual answer grounding examples. The same approach can be used to create synthetic datasets at scale from any document source and it exclusively relies on open-source technologies, reducing the overall dataset production cost only to GPU hours. Moreover, through efficient inference implementations it is possible to further scale the production of such datasets.

Finally, we addressed the VAG issue fine-tuning a model for during-generation and post-hoc visual attribution, significantly improving the attribution performance. Moreover, the dataset constructed through MAPPET was partially utilized for inference, benchmarking existing models with answer grounding capabilities. Our results indicate that current models, while able to localize objects in general images, struggle with precise localization in document-types images, achieving superior performance in answer generation and reasoning.

Chapter 9

Future Work and Limitations

While this thesis has developed robust mechanisms for explainable and multimodal document understanding in an industrial setting, challenges for future research remain. The rapid evolution of MLLMs continues to open new possibilities that we couldn't explore in this work due to constraints like time and technological maturity.

Advancements in Chart-to-Table Technologies. One of the primary problems we addressed is the quality of C2T models. Our evaluation revealed that larger models perform well on standardized benchmarks but suffer from heavy hallucinations when applied to real-world documents. Future work in this field should concentrate on the development of pre-training objectives prioritizing numerical precision and structural fidelity over general visual reasoning in order to obtain smaller artifacts that can be deployed in complex ingestion pipelines without the heavy computational overheads typical of generative technologies.

Integration of Retrieval Augmented Generation (RAG) A fundamental limitation we faced for our summarization pipeline was the strict imposition not to use retrievers. As documents in the financial and legal domains grow in length and complexity, relying solely on the context window becomes increasingly prohibitive. Future iterations of this pipeline should integrate hybrid RAG mechanisms to handle both text and images. This could potentially overcome the limitations of current context windows, improve factual consistency, and limit information loss. Moreover, the integration of retrieval techniques has the potential of speeding up

the pipeline at inference time and creating summaries coming from different but related documents. Finally, while we provided an additional evaluation system for the inclusion of multi-modal information, the research on this topic is generally limited and requires further clarifications.

Scaling Context-Attribution for Long Dependencies. For context-attribution, we observed that cross-encoders offer cost-effective alternatives to proprietary models, while struggling with context length and long dependencies. Moreover, out solution is model-agnostic, while the research community is consistently switching to data-agnostic models. Ideally, research should be able to provide data agnostic solutions as well as large-scale datasets to train and evaluate on the context-attribution task. Moreover, our approach is currently limited to answer-level context-attribution. While LLMs can provide fine-grained citations, cross-encoders are currently limited to coarse-grained attributions, thus creating the need for lightweight solutions capable of providing in-line citations.

Refining Visual Grounding in Document Domains. Our benchmark for Visual Answer Grounding revealed that current models struggle with precise ROI localization in document-type images. This suggests a domain gap that general pre-training doesn't address, probably due to data unavailability. Furthermore, while our MAPPET algorithm allows for synthetic dataset creation, its applicability to multiple attribution was not explored. Optimizing this algorithms for multiple-attributions is necessary to elicit further visual grounding models development.

Cost-Efficient Deployment and Evaluation. Finally, the industrial context of this research stresses the continuous tension between model performance and operational cost. While ever advancing reasoning models improve 1LLM performance, they also consistently increase inference times and costs. Future efforts should prioritize the development of lightweight architectures that can approximate the reasoning capabilities of proprietary models without the associated heavy costs.

Ethical and Regulatory Implications. The applications described in this manuscript are mostly financial and legal in nature, which demands attention to the ethical and regulatory implications of LLM-based solutions. The integration of generative models into corporate pipelines introduces significant risks regarding data privacy, bias, and the black-box nature of LLM-based decision making. With

the introduction of more stringent regulatory frameworks, such as the AI Act, and the ever evolving data protection regulations, future research must address compliance by design. While our work advances model explainability through context-attribution and visual grounding, further efforts are required to guarantee comprehensive auditability across the ingestion and generation steps. Future development should focus on standardized protocols to assess systemic biases in MLLMs, ensuring that the output is factually consistent, verifiable, legally compliant, and equitable. Additionally, the creation of robust frameworks for robust data handling should be prioritized to prevent the leakage of sensitive information during model fine-tuning and inference.0

Bibliography

[1] Josh Achiam et al. Gpt-4 technical report. In: *arXiv preprint arXiv:2303.08774* (2023) (cit. on pp. 43, 75, 89, 105).

[2] Griffin Adams, Alexander R. Fabbri, Faisal Ladhak, Eric Lehman and Noémie Elhadad. From Sparse to Dense: GPT-4 Summarization with Chain of Density Prompting. In: *Proceedings of the 4th New Frontiers in Summarization Workshop*. 2023, 68–74 (cit. on pp. 18, 19, 61, 63).

[3] Saif Ahmad, Paulo C. F. de Oliveira and Khurshid Ahmad. Summarization of Multimodal Information. In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation*. 2004 (cit. on p. 76).

[4] Shuai Bai et al. *Qwen3-VL Technical Report*. 2025 (cit. on pp. 96, 139).

[5] Zhe Chen et al. *Expanding Performance Boundaries of Open-Source Multimodal Models with Model, Data, and Test-Time Scaling*. 2025 (cit. on pp. 43, 126, 139).

[6] Jean-Baptiste Alayrac et al. Flamingo: a Visual Language Model for Few-Shot Learning. In: *Advances in Neural Information Processing Systems*. 2022 (cit. on p. 17).

[7] Rasim Alguliev, Ramiz Aliguliyev and Makrufa Hajirahimova. Multi-document summarization model based on integer linear programming. In: *Intelligent Control and Automation* (2010), 105–111 (cit. on p. 77).

[8] Rami Aly, Zhiqiang Tang, Samson Tan and George Karypis. Learning to Generate Answers with Citations via Factual Consistency Models. In: *Annual Meeting of the Association for Computational Linguistics*. 2024, 11876–11896 (cit. on pp. 22, 105, 107).

[9] Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu and Xuanjing Huang. CoNT: Contrastive Neural Text Generation. In: *Advances in Neural Information Processing Systems*. 2022 (cit. on p. 38).

[10] Chenxin An, Ming Zhong, Zhichao Geng, Jianqiang Yang and Xipeng Qiu. Retrievalsum: A retrieval enhanced framework for abstractive summarization. In: *arXiv preprint arXiv:2109.07943* (2021) (cit. on p. 18).

[11] Stanislaw Antol et al. Vqa: Visual question answering. In: *Proceedings of the IEEE international conference on computer vision*. 2015, 2425–2433 (cit. on p. 125).

[12] Vamsi Aribandi et al. ExT5: Towards Extreme Multi-Task Scaling for Transfer Learning. In: *International Conference on Learning Representations*. 2022 (cit. on p. 36).

[13] Christoph Auer et al. Docling technical report. In: *arXiv preprint arXiv:2408.09869* (2024) (cit. on p. 133).

[14] Dzmitry Bahdanau, Kyunghyun Cho and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In: *International Conference on Learning Representations*. 2015 (cit. on p. 12).

[15] Seyed Ali Bahrainian, Sheridan Feucht and Carsten Eickhoff. NEWTS: A Corpus for News Topic-Focused Summarization. In: *Annual Meeting of the Association for Computational Linguistics*. 2022, 493–503 (cit. on pp. 68, 69).

[16] Shuai Bai et al. Qwen2.5-VL Technical Report. In: *arXiv preprint arXiv:2502.13923* (2025) (cit. on pp. 126, 134, 139).

[17] Satanjeev Banerjee and Alon Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In: *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and Summarization*. 2005, 65–72 (cit. on p. 32).

[18] Iz Beltagy, Kyle Lo and Arman Cohan. SciBERT: A Pretrained Language Model for Scientific Text. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*. 2019, 3613–3618 (cit. on p. 62).

[19] Iz Beltagy, Matthew E Peters and Arman Cohan. Longformer: The long-document transformer. In: *arXiv preprint arXiv:2004.05150* (2020) (cit. on p. 13).

[20] Jingwen Bian, Yang Yang and Tat-Seng Chua. Multimedia summarization for trending topics in microblogs. In: *Proceedings of the ACM international Conference on information & knowledge management*. 2013 (cit. on p. 77).

[21] Junyi Bian, Xiaolei Qin, Wuhe Zou, Mengzuo Huang and Weidong Zhang. Hellama: Llama-based table to text generation by highlighting the important evidence. In: *arXiv preprint arXiv:2311.08896* (2023) (cit. on p. 29).

[22] Galal M. Binmakhashen and Sabri A. Mahmoud. Document Layout Analysis: A Comprehensive Survey. In: *ACM Computing Surveys* (2019) (cit. on p. 128).

[23] Steven Bird. NLTK: The Natural Language Toolkit. In: *International Conference on Computational Linguistics*. 2006 (cit. on p. 62).

[24] Ali Furkan Biten et al. Scene Text Visual Question Answering. In: *International Conference on Computer Vision*. 2019 (cit. on p. 138).

[25] Lukas Blecher, Guillem Cucurull, Thomas Scialom and Robert Stojnic. Nougat: Neural Optical Understanding for Academic Documents. In: (2024) (cit. on p. 77).

[26] David M Blei, Andrew Y Ng and Michael I Jordan. Latent dirichlet allocation. In: *Journal of machine Learning research* (2003), 993–1022 (cit. on p. 68).

[27] Bernd Bohnet et al. Attributed question answering: Evaluation and modeling for attributed large language models. In: *arXiv preprint arXiv:2212.08037* (2022) (cit. on pp. 22, 108).

[28] Tom Brown et al. Language models are few-shot learners. In: *Advances in neural information processing systems* (2020), 1877–1901 (cit. on pp. 14, 15, 29, 30, 75).

[29] Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Börschinger and Tal Schuster. Tomayto, Tomahto. Beyond Token-level Answer Equivalence for Question Answering Evaluation. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Dec. 2022, 291–305 (cit. on pp. 98, 101–103).

[30] Davide Caffagni et al. The Revolution of Multimodal Large Language Models: A Survey. In: *ACL Findings*. 2024 (cit. on pp. 125, 128).

[31] Chongyan Chen, Samreen Anjum and Danna Gurari. Grounding Answers for Visual Questions Asked by Visually Impaired People. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2022 (cit. on p. 125).

[32] Miao Chen et al. Towards Table-to-Text Generation with Pretrained Language Model: A Table Structure Understanding and Text Deliberating Approach. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2022, 8199–8210 (cit. on p. 37).

[33] Wenhu Chen. Large Language Models are few(1)-shot Table Reasoners. In: *Findings of the Association for Computational Linguistics: EACL*. 2023, 1090–1100 (cit. on p. 29).

[34] Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen and William Yang Wang. Logical Natural Language Generation from Open-Domain Tables. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 2020, 7929–7942 (cit. on pp. 16, 29).

[35] Xi Chen et al. Pali-3 vision language models: Smaller, faster, stronger. In: *arXiv preprint arXiv:2310.09199* (2023) (cit. on pp. 44, 128).

[36] Zhe Chen et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. In: *Science China Information Sciences* (2024), 220101 (cit. on pp. 126, 128).

[37] Zhe Chen et al. InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2024 (cit. on pp. 43, 126).

[38] Jianpeng Cheng and Mirella Lapata. Neural Summarization by Extracting Sentences and Words. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016, 484–494 (cit. on p. 19).

[39] Zhoujun Cheng et al. Binding Language Models in Symbolic Languages. In: *International Conference on Learning Representations*. 2023 (cit. on p. 29).

[40] Rewon Child, Scott Gray, Alec Radford and Ilya Sutskever. Generating long sequences with sparse transformers. In: *arXiv preprint arXiv:1904.10509* (2019) (cit. on p. 13).

[41] Jaemin Cho, Jie Lei, Hao Tan and Mohit Bansal. Unifying vision-and-language tasks via text generation. In: *International Conference on Machine Learning*. 2021 (cit. on p. 77).

[42] Aakanksha Chowdhery et al. Palm: Scaling language modeling with pathways. In: *Journal of Machine Learning Research* (2023), 1–113 (cit. on p. 36).

[43] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg and Dario Amodei. Deep reinforcement learning from human preferences. In: *Advances in Neural Information Processing Systems*. 2017, 4299–4307 (cit. on p. 13).

[44] Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim and James R. Glass. Lookback Lens: Detecting and Mitigating Contextual Hallucinations in Large Language Models Using Only Attention Maps. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Nov. 2024, 1419–1436 (cit. on p. 105).

[45] Jordan Clive, Kris Cao and Marek Rei. Control prefixes for parameter-efficient text generation. In: *Proceedings of the Workshop on Natural Language Generation, Evaluation, and Metrics*. 2022, 363–382 (cit. on p. 36).

[46] Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev and Aleksander Madry. ContextCite: Attributing Model Generation to Context. In: *Advances in Neural Information Processing Systems*. 2024 (cit. on pp. 22, 105, 107, 108).

[47] Gordon V Cormack, Charles LA Clarke and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: *Proceedings of the international ACM SIGIR conference on Research and development in information retrieval*. 2009, 758–759 (cit. on p. 109).

[48] Tri Dao. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In: *International Conference on Learning Representations*. 2024 (cit. on p. 13).

[49] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra and Christopher Ré. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In: *Advances in Neural Information Processing Systems*. 2022 (cit. on p. 13).

[50] Luca De Grandisa, Francesco Maria Granataa, Davide Costaa, Antonio Lanzaa and Ermelinda Oroa. Improving Context-Attribution with Semi-Supervised Cross-Encoders. In: (2025) (cit. on p. 8).

[51] Chao Deng et al. LongDocURL: a Comprehensive Multimodal Long Document Benchmark Integrating Understanding, Reasoning, and Locating. In: *Annual Meeting of the Association for Computational Linguistics*. 2025 (cit. on pp. 126, 128, 132).

[52] Xiang Deng, Huan Sun, Alyssa Lees, You Wu and Cong Yu. TURL: Table Understanding through Representation Learning. In: *SIGMOID Records* (2022), 33–40 (cit. on p. 16).

[53] Tim Dettmers, Mike Lewis, Younes Belkada and Luke Zettlemoyer. GPT3.int8(): 8-bit Matrix Multiplication for Transformers at Scale. In: *Advances in Neural Information Processing Systems*. 2022 (cit. on p. 96).

[54] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019, 4171–4186 (cit. on p. 14).

[55] Bhuwan Dhingra, Manaal Faruqui, Ankur P. Parikh, Ming-Wei Chang, Dipanjan Das and William W. Cohen. Handling Divergent Reference Texts when Evaluating Table-to-Text Generation. In: *Procedings of the Annual Meeting of the Association for Computational Linguistics*. 2019, 4884–4895 (cit. on p. 32).

[56] Alexey Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: *International Conference on Learning Representations*. 2021 (cit. on p. 17).

[57] Nan Du et al. Glam: Efficient scaling of language models with mixture-of-experts. In: *International conference on machine learning*. 2022, 5547–5569 (cit. on p. 96).

[58] Jeffrey L Elman. Finding structure in time. In: *Cognitive science* (1990), 179–211 (cit. on p. 11).

[59] Shahul Es, Jithin James, Luis Espinosa Anke and Steven Schockaert. RAGAs: Automated Evaluation of Retrieval Augmented Generation. In: *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Mar. 2024, 150–158 (cit. on pp. 98, 101, 103).

[60] Georgios Evangelopoulos et al. Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention. In: *IEEE Transactions on Multimedia* (2013) (cit. on p. 77).

[61] Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher and Dragomir Radev. Summeval: Re-evaluating summarization evaluation. In: *Transactions of the Association for Computational Linguistics* (2021) (cit. on p. 75).

[62] Alexander R. Fabbri, Chien-Sheng Wu, Wenhao Liu and Caiming Xiong. QAFactEval: Improved QA-Based Factual Consistency Evaluation for Summarization. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2022, 2587–2601 (cit. on p. 62).

[63] Angela Fan, David Grangier and Michael Auli. Controllable Abstractive Summarization. In: *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. July 2018, 45–54 (cit. on p. 19).

[64] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston and Michael Auli. ELI5: Long Form Question Answering. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. July 2019, 3558–3567 (cit. on pp. 108, 109).

[65] William Fedus, Barret Zoph and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. In: *Journal of Machine Learning Research* (2022), 1–39 (cit. on p. 96).

[66] Dimitrios Galanis, Gerasimos Lampouras and Ion Androutsopoulos. Extractive multi-document summarization with integer linear programming and support vector regression. In: *Proceedings of COLING 2012*. 2012, 911–926 (cit. on p. 77).

[67] Luyu Gao et al. Rarr: Researching and revising what language models say, using language models. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2023, 16477–16508 (cit. on pp. 21–23, 107).

[68] Mingqi Gao, Xinyu Hu, Xunjian Yin, Jie Ruan, Xiao Pu and Xiaojun Wan. Llm-based nlg evaluation: Current status and challenges. In: *Computational Linguistics* (2025), 1–27 (cit. on p. 98).

[69] Tianyu Gao, Howard Yen, Jiatong Yu and Danqi Chen. Enabling Large Language Models to Generate Text with Citations. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2023, 6465–6488 (cit. on pp. 22, 105, 107, 110, 126).

[70] Claire Gardent, Anastasia Shimorina, Shashi Narayan and Laura Perez-Beltrachini. Creating Training Corpora for NLG Micro-Planners. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. July 2017, 179–188 (cit. on p. 30).

[71] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats and Yann N Dauphin. Convolutional sequence to sequence learning. In: *International conference on machine learning*. 2017, 1243–1252 (cit. on p. 12).

[72] Carlos Gemmell and Jeffrey Dalton. Generate, transform, answer: Question specific tool synthesis for tabular data. In: *arXiv preprint arXiv:2303.10138* (2023) (cit. on p. 29).

[73] Akash Ghosh, Arkadeep Acharya, Raghav Jain, Sriparna Saha, Aman Chadha and Setu Sinha. Clipsyntel: clip and llm synergy for multimodal question summarization in healthcare. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2024 (cit. on p. 77).

[74] Tanya Goyal, Junyi Jessy Li and Greg Durrett. News summarization and evaluation in the era of gpt-3. In: *arXiv preprint arXiv:2209.12356* (2022) (cit. on pp. 19, 63).

[75] Aaron Grattafiori et al. The llama 3 herd of models. In: *arXiv preprint arXiv:2407.21783* (2024) (cit. on p. 109).

[76] Alex Graves. Generating sequences with recurrent neural networks. In: *arXiv preprint arXiv:1308.0850* (2013) (cit. on p. 12).

[77] Alex Graves, Greg Wayne and Ivo Danihelka. Neural turing machines. In: *arXiv preprint arXiv:1410.5401* (2014) (cit. on p. 12).

[78] Max Grusky, Mor Naaman and Yoav Artzi. Newsroom: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2018, 708–719 (cit. on p. 63).

[79] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat and Ming-Wei Chang. Retrieval Augmented Language Model Pre-Training. In: *Proceedings of the International Conference on Machine Learning*. 2020, 3929–3938 (cit. on p. 98).

[80] Junxian He, Wojciech Kryscinski, Bryan McCann, Nazneen Rajani and Caiming Xiong. CTRLsum: Towards Generic Controllable Text Summarization. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2022, 5879–5915 (cit. on pp. 19, 75).

[81] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang and David A. Sontag. TabLLM: Few-shot Classification of Tabular Data with Large Language Models. In: *International Conference on Artificial Intelligence and Statistics*. 2023, 5549–5581 (cit. on p. 16).

[82] D Hendrycks. Gaussian Error Linear Units (Gelus). In: *arXiv preprint arXiv:1606.08415* (2016) (cit. on p. 13).

[83] Karl Moritz Hermann et al. Teaching machines to read and comprehend. In: *Advances in neural information processing systems* (2015) (cit. on p. 60).

[84] Salah El Hihi and Yoshua Bengio. Hierarchical Recurrent Neural Networks for Long-Term Dependencies. In: *Advances in Neural Information Processing Systems*. 1995, 493–499 (cit. on p. 12).

[85] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. In: *Neural computation* (1997), 1735–1780 (cit. on p. 11).

[86] Jordan Hoffmann et al. Training Compute-Optimal Large Language Models. In: *arXiv preprint arXiv:2203.15556* (2022) (cit. on p. 13).

[87] Anwen Hu et al. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. In: *Findings of the Association for Computational Linguistics*. 2024, 3096–3120 (cit. on p. 44).

[88] Edward J Hu et al. Lora: Low-rank adaptation of large language models. In: *International Conference on Learning Representations* (2022), 3 (cit. on pp. 32, 140).

[89] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu and Furu Wei. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. In: *The ACM International Conference on Multimedia*. 2022, 4083–4091 (cit. on p. 77).

[90] Siqing Huo, Negar Arabzadeh and Charles L. A. Clarke. Retrieving Supporting Evidence for Generative Question Answering. In: *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*. 2023, 11–20 (cit. on p. 99).

[91] Aaron Hurst et al. *Gpt-4o system card*. 2024 (cit. on pp. 43, 89).

[92] Anubhav Jangra, Adam Jatowt, Mohammad Hasanuzzaman and Sriparna Saha. Text-image-video summary generation using joint integer linear programming. In: *European Conference on Information Retrieval*. 2020, 190–198 (cit. on p. 77).

[93] Anubhav Jangra, Sriparna Saha, Adam Jatowt and Mohammed Hasanuzzaman. Multi-Modal Supplementary-Complementary Summarization using Multi-Objective Optimization. In: *The International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2021, 818–828 (cit. on pp. 75, 76).

[94] Fred Jelinek, Robert L Mercer, Lalit R Bahl and James K Baker. Perplexity—a measure of the difficulty of speech recognition tasks. In: *The Journal of the Acoustical Society of America* (1977), S63–S63 (cit. on p. 131).

[95] Ziwei Ji et al. Survey of Hallucination in Natural Language Generation. In: *ACM Computing Surveys* (2023), 248:1–248:38 (cit. on pp. 14, 20).

[96] Jared Kaplan et al. Scaling laws for neural language models. In: *arXiv preprint arXiv:2001.08361* (2020) (cit. on p. 13).

[97] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas and François Fleuret. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. In: *Proceedings of the International Conference on Machine Learning*. 2020, 5156–5165 (cit. on p. 13).

[98] Vasileios Katranidis and Gabor Barany. Faaf: Facts as a function for the evaluation of rag systems. In: *arXiv preprint arXiv:2403.03888* (2024) (cit. on p. 99).

[99] Geewook Kim et al. Donut: Document understanding transformer without ocr. In: *arXiv preprint arXiv:2111.15664* (2021), 2 (cit. on pp. 77, 128).

[100] Tomáš Kočiskỳ et al. The narrativeqa reading comprehension challenge. In: *Transactions of the Association for Computational Linguistics* (2018), 317–328 (cit. on p. 100).

[101] Wojciech Kryscinski, Bryan McCann, Caiming Xiong and Richard Socher. Evaluating the Factual Consistency of Abstractive Text Summarization. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2020, 9332–9346 (cit. on p. 62).

[102] Taku Kudo. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 2018, 66–75 (cit. on p. 13).

[103] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2018, 66–71 (cit. on p. 13).

[104] Litton J Kurisinkel and Nancy F Chen. LLM based multi-document summarization exploiting main-event biased monotone submodular content extraction. In: *arXiv preprint arXiv:2310.03414* (2023) (cit. on pp. 18, 19).

[105] Woosuk Kwon et al. Efficient Memory Management for Large Language Model Serving with PagedAttention. In: *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*. 2023 (cit. on pp. 13, 111, 139).

[106] Philippe Laban et al. SummEdits: Measuring LLM Ability at Factual Reasoning Through The Lens of Summarization. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2023, 9662–9676 (cit. on p. 18).

[107] Nicola Landro, Ignazio Gallo, Riccardo La Grassa and Edoardo Federici. Two new datasets for italian-language abstractive text summarization. In: *Information* (2022), 228 (cit. on pp. 59, 60, 66).

[108] Yann LeCun, Léon Bottou, Yoshua Bengio and Patrick Haffner. Gradient-based learning applied to document recognition. In: *Proceedings of the IEEE* (1998), 2278–2324 (cit. on p. 17).

[109] Mike Lewis et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 2020, 7871–7880 (cit. on p. 75).

[110] Patrick Lewis et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In: *Advances in neural information processing systems* (2020), 9459–9474 (cit. on pp. 98, 105).

[111] Bo Li et al. Llava-onevision: Easy visual task transfer. In: *arXiv preprint arXiv:2408.03326* (2024) (cit. on p. 128).

[112] Chaofan Li, Zheng Liu, Shitao Xiao and Yingxia Shao. Making large language models a better foundation for dense retrieval. In: *arXiv preprint arXiv:2312.15503* (2023) (cit. on p. 108).

[113] Dongfang Li et al. TruthReader: Towards Trustworthy Document Assistant Chatbot with Reliable Attribution. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Nov. 2024, 89–100 (cit. on pp. 22, 107).

[114] Haoran Li, Junnan Zhu, Cong Ma, Jiajun Zhang and Chengqing Zong. Multi-modal summarization for asynchronous collection of text, image, audio and video. In: *Proceedings of the 2017 conference on empirical methods in natural language processing*. 2017, 1092–1102 (cit. on p. 77).

[115] Junnan Li, Dongxu Li, Silvio Savarese and Steven C. H. Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In: *Proceedings of the International Conference on Machine Learning*. 2023, 19730–19742 (cit. on p. 17).

[116] Miao Li, Eduard H. Hovy and Jey Han Lau. Summarizing Multiple Documents with Conversational Structure for Meta-Review Generation. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2023, 7089–7112 (cit. on p. 18).

[117] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. In: *arXiv preprint arXiv:2308.03281* (2023) (cit. on p. 109).

[118] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In: *Text summarization branches out*. 2004, 74–81 (cit. on pp. 32, 62).

[119] Jingyang Lin et al. Videoxum: Cross-modal visual and textural summarization of videos. In: *IEEE Transactions on Multimedia* (2023) (cit. on pp. 75, 77).

[120] Fangyu Liu et al. DePlot: One-shot visual language reasoning by plot-to-table translation. In: *Findings of the Association for Computational Linguistics*. 2023, 10381–10399 (cit. on pp. 42, 44).

[121] Haotian Liu, Chunyuan Li, Yuheng Li and Yong Jae Lee. Improved Baselines with Visual Instruction Tuning. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2024 (cit. on pp. 125, 128).

[122] Junpeng Liu et al. VisualWebBench: How Far Have Multimodal LLMs Evolved in Web Page Understanding and Grounding? In: *arXiv preprint arXiv:2404.05955* (2024) (cit. on pp. 126, 129, 132).

[123] Nelson F. Liu, Tianyi Zhang and Percy Liang. *Evaluating Verifiability in Generative Search Engines*. 2023 (cit. on p. 115).

[124] Yang Liu. Fine-tune BERT for extractive summarization. In: *arXiv preprint arXiv:1903.10318* (2019) (cit. on p. 19).

[125] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu and Chenguang Zhu. G-Eval: NLG Evaluation using Gpt-4 with Better Human Alignment. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2023, 2511–2522 (cit. on pp. 75, 89).

[126] Yang Liu and Mirella Lapata. Text Summarization with Pretrained Encoders. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2019, 3728–3738 (cit. on p. 18).

[127] Yixin Liu, Pengfei Liu, Dragomir R. Radev and Graham Neubig. BRIO: Bringing Order to Abstractive Summarization. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 2022, 2890–2903 (cit. on pp. 63, 64).

[128] Yixin Liu et al. Benchmarking Generation and Evaluation Capabilities of Large Language Models for Instruction Controllable Summarization. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*. June 2024, 4481–4501 (cit. on p. 19).

[129] Pan Lu et al. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In: *arXiv preprint arXiv:2310.02255* (2023) (cit. on p. 138).

[130] Xinyang Lu et al. WASA: WAtermark-based Source Attribution for Large Language Model-Generated Data. In: *Findings of the Association for Computational Linguistics*. 2025, 23791–23824 (cit. on pp. 22, 105, 107).

[131] Xinyuan Lu, Liangming Pan, Qian Liu, Preslav Nakov and Min-Yen Kan. SCITAB: A Challenging Benchmark for Compositional Reasoning and Claim Verification on Scientific Tables. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2023, 7787–7813 (cit. on p. 29).

[132] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. In: *IEEE Transactions on Audio, Speech and Language Processing* (2025) (cit. on p. 14).

[133] Tengchao Lv et al. Kosmos-2.5: A multimodal literate model. In: *arXiv preprint arXiv:2309.11419* (2023) (cit. on p. 77).

[134] Xueguang Ma, Shengyao Zhuang, Bevan Koopman, Guido Zuccon, Wenhu Chen and Jimmy Lin. VISA: Retrieval Augmented Generation with Visual Source Attribution. In: *Annual Meeting of the Association for Computational Linguistics*. 2025 (cit. on pp. 126, 129, 132, 139).

[135] Yubo Ma et al. MMLONGBENCH-DOC: Benchmarking Long-context Document Understanding with Visualizations. In: 2024 (cit. on pp. 126, 128, 132, 138).

[136] Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar and Dan Roth. ExpertQA: Expert-Curated Questions and Attributed Answers. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*. 2024 (cit. on pp. 105, 126).

[137] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul and Benjamin Bossan. *PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods*. `https://github.com/huggingface/peft`. 2022 (cit. on p. 140).

[138] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq R. Joty and Enamul Hoque. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In: *Findings of the Association for Computational Linguistics*. 2022, 2263–2279 (cit. on p. 42).

[139] Ahmed Masry, Megh Thakkar, Aayush Bajaj, Aaryaman Kartha, Enamul Hoque and Shafiq Joty. ChartGemma: Visual Instruction-tuning for Chart Reasoning in the Wild. In: *Proceedings of the International Conference on Computational Linguistics*. 2025, 625–643 (cit. on p. 44).

[140] Minesh Mathew, Dimosthenis Karatzas and CV Jawahar. Docvqa: A dataset for vqa on document images. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2021, 2200–2209 (cit. on pp. 128, 132).

[141] Joshua Maynez, Shashi Narayan, Bernd Bohnet and Ryan T. McDonald. On Faithfulness and Factuality in Abstractive Summarization. In: *Annual Meeting of the Association for Computational Linguistics*. 2020, 1906–1919 (cit. on pp. 20, 75).

[142] Jacob Menick et al. Teaching language models to support answers with verified quotes. In: *arXiv preprint arXiv:2203.11147* (2022) (cit. on pp. 22, 105, 107).

[143] Kaiz Merchant and Yash Pande. NLP Based Latent Semantic Analysis for Legal Text Summarization. In: *International Conference on Advances in Computing, Communications and Informatics*. 2018, 1803–1807 (cit. on p. 19).

[144] Rada Mihalcea and Paul Tarau. TextRank: Bringing Order into Text. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2004, 404–411 (cit. on pp. 18, 19).

[145] Derek Miller. Leveraging BERT for extractive text summarization on lectures. In: *arXiv preprint arXiv:1906.04165* (2019) (cit. on p. 19).

[146] Sewon Min et al. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2023 (cit. on p. 126).

[147] Abhika Mishra et al. Fine-grained Hallucination Detection and Editing for Language Models. In: *First Conference on Language Modeling*. 2024 (cit. on pp. 21, 125, 126).

[148] Natwar Modani et al. Summarizing multimedia content. In: *International Conference on Web Information Systems Engineering*. 2016, 340–348 (cit. on p. 77).

[149] Ryan Muther and David Smith. Citations as Queries: Source Attribution Using Language Models as Rerankers. In: *arXiv preprint arXiv:2306.17322* (2023) (cit. on pp. 22, 23, 108).

[150] Reiichiro Nakano et al. Webgpt: Browser-assisted question-answering with human feedback. In: *arXiv preprint arXiv:2112.09332* (2021) (cit. on pp. 22, 105, 107).

[151] Ramesh Nallapati, Feifei Zhai and Bowen Zhou. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In: *Proceedings of the Conference on Artificial Intelligence*. 2017, 3075–3081 (cit. on p. 19).

[152] Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre and Bing Xiang. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In: *Proceedings of the Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*. 2016, 280–290 (cit. on pp. 59, 60).

[153] Shashi Narayan, Shay B. Cohen and Mirella Lapata. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2018, 1797–1807 (cit. on pp. 59, 60).

[154] Ahmed Nassar et al. *SmolDocling: An ultra-compact vision-language model for end-to-end multi-modal document conversion*. 2025 (cit. on p. 96).

[155] Ahmed Nassar et al. SmolDocling: An ultra-compact vision-language model for end-to-end multi-modal document conversion. In: *arXiv preprint arXiv:2503.11576* (2025) (cit. on p. 128).

[156] Humza Naveed et al. A Comprehensive Overview of Large Language Models. In: *ACM Trans. Intell. Syst. Technol.* (2025) (cit. on p. 126).

[157] Ani Nenkova and Kathleen McKeown. A survey of text summarization techniques. In: *Mining text data* (2012) (cit. on p. 75).

[158] Ermelinda Oro, Luca De Grandis, Francesco Maria Granata and Massimo Ruffolo. Leveraging Large Language Models for Flexible and Robust Table-to-Text Generation. In: *International Conference on Database and Expert Systems Applications*. 2024, 222–227 (cit. on pp. 5, 6, 81).

[159] Ermelinda Oro, Francesco Maria Granata, Antonio Lanza, Amir Bachir, Luca De Grandis and Massimo Ruffolo. Evaluating Retrieval-Augmented Generation for Question Answering with Large Language Models. In: *Ital-IA*. 2024, 12–17 (cit. on p. 7).

[160] Long Ouyang et al. Training language models to follow instructions with human feedback. In: *Advances in Neural Information Processing Systems*. 2022, 27730–27744 (cit. on pp. 13, 16).

[161] Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 2002, 311–318 (cit. on pp. 32, 62).

[162] Ankur P. Parikh et al. ToTTo: A Controlled Table-To-Text Generation Dataset. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2020, 1173–1186 (cit. on p. 30).

[163] Nilay Patel, Shivashankar Subramanian, Siddhant Garg, Pratyay Banerjee and Amita Misra. Towards Improved Multi-Source Attribution for Long-Form Answer Generation. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*. 2024, 3906–3919 (cit. on pp. 22, 107, 108, 115).

[164] Tim Pearce and Jinyeop Song. Reconciling Kaplan and Chinchilla Scaling Laws. In: *Transactions on Machine Learning Research* (2024) (cit. on p. 13).

[165] Maria Soledad Pera and Yiu-Kai Ng. A Naive Bayes Classifier for Web Document Summaries Created by Using Word Similarity and Significant Factors. In: *International Journal on Artificial Intelligence Tools* (2010), 465–486 (cit. on p. 19).

[166] Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar and Peter W. J. Staar. DocLayNet: A Large Human-Annotated Dataset for Document-Layout Segmentation. In: *The ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022, 3743–3751 (cit. on p. 79).

[167] Jason Phang, Yao Zhao and Peter J. Liu. Investigating Efficiently Extending Transformers for Long Input Summarization. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2023, 3946–3961 (cit. on p. 18).

[168] Aleksandra Piktus et al. The web is your oyster-knowledge-intensive NLP against a very large web corpus. In: *arXiv preprint arXiv:2112.09924* (2021) (cit. on p. 111).

[169] Ronak Pradeep et al. Ragnarök: A reusable RAG framework and baselines for TREC 2024 retrieval-augmented generation track. In: *European Conference on Information Retrieval*. 2025, 132–148 (cit. on pp. 108–110).

[170] Xiao Pu, Mingqi Gao and Xiaojun Wan. Summarization is (almost) dead. In: *arXiv preprint arXiv:2309.09558* (2023) (cit. on p. 18).

[171] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever et al. Improving language understanding by generative pre-training. In: (2018) (cit. on pp. 13, 14).

[172] Alec Radford et al. Learning Transferable Visual Models From Natural Language Supervision. In: *Proceedings of the International Conference on Machine Learning*. 2021, 8748–8763 (cit. on p. 16).

[173] Jack W Rae et al. Scaling language models: Methods, analysis & insights from training gopher. In: *arXiv preprint arXiv:2112.11446* (2021) (cit. on p. 107).

[174] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In: *Advances in Neural Information Processing Systems*. 2023 (cit. on p. 13).

[175] Colin Raffel et al. Exploring the limits of transfer learning with a unified text-to-text transformer. In: *Journal of machine learning research* (2020), 1–67 (cit. on pp. 14, 19, 75).

[176] Raian Rahman, Rizvi Hasan, Abdullah Al Farhad, Md Tahmid Rahman Laskar, Md Hamjajul Ashmafee and Abu Raihan Mostofa Kamal. ChartSumm: A Comprehensive Benchmark for Automatic Chart Summarization of Long and Short Summaries. In: *Canadian AI*. 2023 (cit. on p. 77).

[177] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In: *arXiv preprint arXiv:1606.05250* (2016) (cit. on p. 98).

[178] Nedunchelian Ramanujam and Manivannan Kaliappan. An automatic multidocument text summarization approach based on Naive Bayesian classifier using timestamp strategy. In: *The Scientific World Journal* (2016), 1784827 (cit. on p. 19).

[179] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*. 2019, 3980–3990 (cit. on pp. 106, 108).

[180] Stephen Robertson, Hugo Zaragoza et al. The probabilistic relevance framework: BM25 and beyond. In: *Foundations and Trends® in Information Retrieval* (2009), 333–389 (cit. on p. 109).

[181] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. In: *Journal of computational and applied mathematics* (1987), 53–65 (cit. on p. 110).

[182] Jon Saad-Falcon, Omar Khattab, Christopher Potts and Matei Zaharia. ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. June 2024, 338–354 (cit. on p. 98).

[183] Victor Sanh et al. Multitask Prompted Training Enables Zero-Shot Task Generalization. In: *International Conference on Learning Representations*. 2022 (cit. on pp. 16, 63).

[184] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. In: *IEEE transactions on Signal Processing* (1997), 2673–2681 (cit. on p. 12).

[185] Abigail See, Peter J. Liu and Christopher D. Manning. Get To The Point: Summarization with Pointer-Generator Networks. In: *Procedings of the Annual Meeting of the Association for Computational Linguistics*. 2017, 1073–1083 (cit. on pp. 18, 19).

[186] Thibault Sellam, Dipanjan Das and Ankur P. Parikh. BLEURT: Learning Robust Metrics for Text Generation. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 2020, 7881–7892 (cit. on pp. 32, 62).

[187] Rico Sennrich, Barry Haddow and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In: *Procedings of the Annual Meeting of the Association for Computational Linguistics*. 2016 (cit. on p. 13).

[188] Peter Shaw, Jakob Uszkoreit and Ashish Vaswani. Self-Attention with Relative Position Representations. In: *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2018, 464–468 (cit. on p. 12).

[189] Haizhou Shi et al. Continual learning of large language models: A comprehensive survey. In: *ACM Computing Surveys* (2025), 1–42 (cit. on p. 14).

[190] Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer and Wen-tau Yih. Trusting Your Evidence: Hallucinate Less with Context-aware Decoding. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*. June 2024, 783–791 (cit. on p. 105).

[191] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela and Jason Weston. Retrieval Augmentation Reduces Hallucination in Conversation. In: *Findings of the Association for Computational Linguistics*. 2021, 3784–3803 (cit. on p. 98).

[192] Ananya Singha, José Cambronero, Sumit Gulwani, Vu Le and Chris Parnin. Tabular representation, noisy operators, and impacts on table structure understanding tasks in llms. In: *arXiv preprint arXiv:2310.10358* (2023) (cit. on p. 16).

[193] Matthew G. Snover, Bonnie J. Dorr, Richard M. Schwartz, Linnea Micciulla and John Makhoul. A Study of Translation Edit Rate with Targeted Human Annotation. In: *Proceedings of the Conference of the Association for Machine Translation in the Americas: Technical Papers*. 2006, 223–231 (cit. on pp. 32, 62).

[194] Hwanjun Song, Hang Su, Igor Shalyminov, Jason Cai and Saab Mansour. FineSurE: Fine-grained Summarization Evaluation using LLMs. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 2024, 906–922 (cit. on p. 89).

[195] Ivan Stelmakh, Yi Luan, Bhuwan Dhingra and Ming-Wei Chang. ASQA: Factoid Questions Meet Long-Form Answers. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Dec. 2022, 8273–8288 (cit. on pp. 108, 109).

[196] Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. In: *Neurocomputing* (2024), 127063 (cit. on p. 12).

[197] Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang and Nigel Collier. Plan-then-Generate: Controlled Data-to-Text Generation via Planning. In: *Findings of the Association for Computational Linguistics*. 2021, 895–909 (cit. on pp. 30, 38).

[198] Lya Hulliyyatus Suadaa, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura and Hiroya Takamura. Towards Table-to-Text Generation with Numerical Reasoning. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*. 2021, 1451–1465 (cit. on pp. 30, 37).

[199] Nishant Subramani, Alexandre Matton, Malcolm Greaves and Adrian Lam. A survey of deep learning approaches for ocr and document understanding. In: *arXiv preprint arXiv:2011.13534* (2020) (cit. on p. 128).

[200] Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han and Dongmei Zhang. Evaluating and enhancing structural understanding capabilities of large language models on tables via input designs. In: *arXiv preprint arXiv:2305.13062* (2023) (cit. on p. 16).

[201] Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han and Dongmei Zhang. Gpt4table: Can large language models understand structured table data? a benchmark and empirical study. In: *arXiv preprint ArXiv:2305.13062* (2023) (cit. on p. 16).

[202] Dídac Surís, Sachit Menon and Carl Vondrick. ViperGPT: Visual Inference via Python Execution for Reasoning. In: *International Conference on Computer Vision*. 2023 (cit. on pp. 125, 126).

[203] Sotaro Takeshita, Tommaso Green, Ines Reinig, Kai Eckert and Simone Paolo Ponzetto. ACLSum: A New Dataset for Aspect-based Summarization of Scientific Publications. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*. 2024, 6660–6675 (cit. on pp. 68, 69).

[204] Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito and Kuniko Saito. SlideVQA: A Dataset for Document Visual Question Answering on Multiple Images. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2023 (cit. on pp. 126, 129, 132).

[205] Ryota Tanaka, Kyosuke Nishida and Sen Yoshida. VisualMRC: Machine Reading Comprehension on Document Images. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021 (cit. on pp. 126, 128, 132).

[206] Yixuan Tang and Yi Yang. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. In: *arXiv preprint arXiv:2401.15391* (2024) (cit. on p. 98).

[207] Akanksha Tiwari, Christian Von Der Weth and Mohan S Kankanhalli. Multimodal multiplatform social media event summarization. In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* (2018), 1–23 (cit. on p. 77).

[208] Hugo Touvron et al. Llama 2: Open foundation and fine-tuned chat models. In: *arXiv preprint arXiv:2307.09288* (2023) (cit. on pp. 29, 30).

[209] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In: *Proceedings of the conference. Association for computational linguistics*. 2019 (cit. on p. 75).

[210] Michael Tschannen et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. In: *arXiv preprint arXiv:2502.14786* (2025) (cit. on p. 17).

[211] Ashok Urlana, Pruthwik Mishra, Tathagato Roy and Rahul Mishra. Controllable Text Summarization: Unraveling Challenges, Approaches, and Prospects - A Survey. In: *Annual Meeting of the Association for Computational Linguistics*. 2024, 1603–1623 (cit. on p. 19).

[212] Oleg V. Vasilyev, Vedant Dharnidharka and John Bohannon. Fill in the BLANC: Human-free quality estimation of document summaries. In: (2020), 11–20 (cit. on p. 62).

[213] Ashish Vaswani et al. Attention is all you need. In: *Advances in Neural Information Processing Systems* (2017) (cit. on pp. 12, 125).

[214] Dhruv Verma, Debaditya Roy and Basura Fernando. Effectively Leveraging CLIP for Generating Situational Summaries of Images and Videos. In: *International Journal of Computer Vision* (2025), 5302–5325 (cit. on p. 77).

[215] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder and Furu Wei. Multilingual E5 Text Embeddings: A Technical Report. In: *arXiv preprint arXiv:2402.05672* (2024) (cit. on p. 109).

[216] Peng Wang et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. In: *arXiv preprint arXiv:2409.12191* (2024) (cit. on pp. 128, 139).

[217] Weiyun Wang et al. InternVL3.5: Advancing Open-Source Multimodal Models in Versatility, Reasoning, and Efficiency. In: *arXiv preprint arXiv:2508.18265* (2025) (cit. on p. 139).

[218] William Yang Wang, Yashar Mehdad, Dragomir Radev and Amanda Stent. A low-rank approximation approach to learning joint embeddings of news stories and images for timeline summarization. In: *Proceedings of the conference of the north American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2016 (cit. on p. 77).

[219] Jason Wei et al. Chain-of-thought prompting elicits reasoning in large language models. In: *Advances in neural information processing systems* (2022), 24824–24837 (cit. on pp. 15, 29).

[220] Luis Wiedmann et al. Finevision: Open data is all you need. In: *arXiv preprint arXiv:2510.17269* (2025) (cit. on pp. 126, 132).

[221] Jeff Wu et al. Recursively summarizing books with human feedback. In: *arXiv preprint arXiv:2109.10862* (2021) (cit. on p. 18).

[222] Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony R. Dick and Anton van den Hengel. Visual question answering: A survey of methods and datasets. In: *Comput. Vis. Image Underst.* (2017), 21–40 (cit. on p. 125).

[223] Renqiu Xia et al. StructChart: On the schema, metric, and augmentation for visual chart understanding. In: *arXiv e-prints* (2023), arXiv–2309 (cit. on p. 44).

[224] Renqiu Xia et al. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. In: *IEEE Transactions on Image Processing* (2025) (cit. on pp. 43, 44).

[225] Bin Xiao et al. Florence-2: Advancing a unified representation for a variety of vision tasks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, 4818–4829 (cit. on p. 77).

[226] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In: *Proceedings of the international ACM SIGIR conference on research and development in information retrieval*. 2024, 641–649 (cit. on p. 108).

[227] Wen Xiao, Iz Beltagy, Giuseppe Carenini and Arman Cohan. PRIMERA: Pyramid-based Masked Sentence Pre-training for Multi-document Summarization. In: *Procedings of the Annual Meeting of the Association for Computational Linguistics*. 2022, 5245–5263 (cit. on p. 18).

[228] Bohao Yang, Chen Tang, Kun Zhao, Chenghao Xiao and Chenghua Lin. Effective Distillation of Table-based Reasoning Ability from LLMs. In: *Proceedings of the Joint International Conference on Computational Linguistics, Language Resources and Evaluation*. 2024, 5538–5550 (cit. on p. 29).

[229] Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel and Shachi Paul. TableFormer: Robust Transformer Modeling for Table-Text Encoding. In: *Procedings of the Annual Meeting of the Association for Computational Linguistics*. 2022, 528–537 (cit. on p. 16).

[230] Xianjun Yang, Yan Li, Xinlu Zhang, Haifeng Chen and Wei Cheng. Exploring the limits of ChatGPT for query or aspect-based text summarization. In: *arXiv preprint arXiv:2302.08081* (2023) (cit. on p. 19).

[231] Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. In: *Advances in neural information processing systems* (2022), 27168–27183 (cit. on p. 96).

[232] Jiabo Ye et al. mPLUG-DocOwl: Modularized Multimodal Large Language Model for Document Understanding. In: *arXiv preprint arXiv:2307.02499* (2023) (cit. on p. 128).

[233] Jiabo Ye et al. UReader: Universal OCR-free Visually-situated Language Understanding with Multimodal Large Language Model. In: *Findings of the Association for Computational Linguistics: EMNLP*. 2023 (cit. on p. 128).

[234] Xi Ye, Ruoxi Sun, Sercan Ö. Arik and Tomas Pfister. Effective Large Language Model Adaptation for Improved Grounding and Citation Generation. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2024, 6237–6251 (cit. on pp. 21, 22, 107, 126).

[235] Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang and Yongbin Li. Large Language Models are Versatile Decomposers: Decomposing Evidence and Questions for Table-based Reasoning. In: *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2023, 174–184 (cit. on p. 29).

[236] Shi Yu et al. VisRAG: Vision-based Retrieval-augmented Generation on Multi-modality Documents. In: *International Conference on Learning Representations*. 2025 (cit. on p. 128).

[237] Jingying Zeng et al. Cite Before You Speak: Enhancing Context-Response Grounding in E-commerce Conversational LLM-Agents. In: *arXiv preprint arXiv:2503.04830* (2025) (cit. on p. 105).

[238] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In: *International Conference on Computer Vision*. 2023, 11941–11952 (cit. on p. 17).

[239] Gongbo Zhang et al. Closing the gap between open source and commercial large language models for medical evidence summarization. In: *NPJ Digital Medicine* (2024) (cit. on p. 61).

[240] Haopeng Zhang, Xiao Liu and Jiawei Zhang. Extractive Summarization via ChatGPT for Faithful Summary Generation. In: *Findings of the Association for Computational Linguistics*. 2023, 3270–3278 (cit. on pp. 18, 61).

[241] Jiajie Zhang et al. LongCite: Enabling LLMs to Generate Fine-grained Citations in Long-Context QA. In: *Annual Meeting of the Association for Computational Linguistics*. 2025, 5098–5122 (cit. on pp. 21, 22, 105, 107).

[242] Jingqing Zhang, Yao Zhao, Mohammad Saleh and Peter J. Liu. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In: *Proceedings of the International Conference on Machine Learning*. 2020, 11328–11339 (cit. on pp. 18, 19, 63).

[243] Jingyi Zhang, Jiaxing Huang, Sheng Jin and Shijian Lu. Vision-Language Models for Vision Tasks: A Survey. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024), 5625–5644 (cit. on pp. 125, 128).

[244] Liang Zhang et al. TinyChart: Efficient Chart Understanding with Program-of-Thoughts Learning and Visual Token Merging. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2024, 1882–1898 (cit. on p. 44).

[245] Tianjun Zhang et al. Raft: Adapting language model to domain specific rag. In: *arXiv preprint arXiv:2403.10131* (2024) (cit. on p. 98).

[246] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT. In: *International Conference on Learning Representations*. 2020 (cit. on pp. 32, 62).

[247] Xin Zhang et al. mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2024, 1393–1412 (cit. on pp. 108, 110).

[248] Yue Zhang et al. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. In: *Computational Linguistics* (2025), 1–46 (cit. on pp. 14, 21).

[249] Yusen Zhang et al. MACSum: Controllable Summarization with Mixed Attributes. In: *Transactions of the Association for Computational Linguistics* (2023), 787–803 (cit. on p. 61).

[250] Zihan Zhang, Meng Fang and Ling Chen. RetrievalQA: Assessing Adaptive Retrieval-Augmented Generation for Short-form Open-Domain Question Answering. In: *Findings of the Association for Computational Linguistics*. Aug. 2024, 6963–6975 (cit. on p. 98).

[251] Yilun Zhao, Haowei Zhang, Shengyun Si, Linyong Nan, Xiangru Tang and Arman Cohan. Investigating table-to-text generation capabilities of large language models in real-world information seeking scenarios. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing: Industry Track*. 2023, 160–175 (cit. on p. 29).

[252] Xu Zheng et al. MLLMs are Deeply Affected by Modality Bias. In: *arXiv preprint arXiv:2505.18657* (2025) (cit. on p. 76).

[253] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu and Xuanjing Huang. Extractive Summarization as Text Matching. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 2020, 6197–6208 (cit. on pp. 63, 64).

[254] Zihan Zhou et al. LLM×MapReduce: Simplified Long-Sequence Processing using Large Language Models. In: *Procedings of the Annual Meeting of the Association for Computational Linguistics*. 2025, 27664–27678 (cit. on p. 18).

[255] Jinguo Zhu et al. InternVL3: Exploring Advanced Training and Test-Time Recipes for Open-Source Multimodal Models. In: *arXiv preprint arXiv:2504.10479* (2025) (cit. on pp. 126, 139).

[256] Junnan Zhu, Haoran Li, Tianshang Liu, Yu Zhou, Jiajun Zhang and Chengqing Zong. MSMO: Multimodal Summarization with Multimodal Output. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2018, 4154–4164 (cit. on pp. 19, 76, 77).

[257] Junnan Zhu, Long Zhou, Haoran Li, Jiajun Zhang, Yu Zhou and Chengqing Zong. Augmenting Neural Sentence Summarization Through Extractive Summarization. In: *Natural Language Processing and Chinese Computing*. 2017, 16–28 (cit. on p. 77).

[258] Junnan Zhu, Yu Zhou, Jiajun Zhang, Haoran Li, Chengqing Zong and Changliang Li. Multimodal summarization with guidance of multimodal reference. In: *Proceedings of the AAAI conference on artificial intelligence*. 2020 (cit. on pp. 76, 77).

[259] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang and Jifeng Dai. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In: *International Conference on Learning Representations*. 2021 (cit. on p. 79).

[260] Yuke Zhu, Oliver Groth, Michael Bernstein and Li Fei-Fei. Visual7W: Grounded Question Answering in Images. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016 (cit. on p. 125).