ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in

Artificial Intelligence

HYPOTHESIS-DRIVEN FORENSIC EMAIL ANALYSIS WITH RETRIEVAL-AUGMENTED GENERATION

CANDIDATE
Ali Loloee Jahromi

SUPERVISOR
Prof. Paolo Torroni

CO-SUPERVISOR
Prof. Michael Spranger

Academic year 2025–2026
Session 1st

To myself, and to those who hold a place in my heart.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In forensic investigations, the value of communication data depends not only on its availability but also on the ability to interpret it in relation to an investigative question. This thesis begins from the perspective that evidence discovery is not merely a search problem, but a process of connecting assumptions and supporting traces. In this chapter, the thesis *Hypothesis-Driven Forensic Email Analysis with Retrieval-Augmented Generation (RAG)* is introduced by first presenting the motivation and problem statement, followed by the definition of the research objectives and the proposal of a RAG-based solution, and concluding with an outline of the thesis structure.

1.1 Motivation and Problem Statement

The exponential growth of digital communication has fundamentally transformed how individuals and organizations exchange information [1]. Among the various communication channels, email remains one of the most widely used communication media in both corporate and personal environments, with global daily traffic exceeding 360 billion messages [2]. As a result, large volumes of email data are continuously generated, creating a "digital memory" that captures critical decisions, interactions, and organizational activities.

This abundance of data presents a significant opportunity for digital forensics and investigative analysis [3]. For example, in the Amerithrax investigation, the FBI investigated the 2001 anthrax letter attacks, which killed five people and sickened seventeen others. As part of the case, investigators reviewed thousands of emails to identify documentary evidence relevant to the investigation [4].

While the availability of large-scale email datasets creates valuable opportunities for forensic investigation, it also introduces significant analytical challenges. Digital forensic investigations are often structured around investigative objectives and hypotheses that guide evidence identification and analysis [5]. In this context, the task extends beyond retrieving emails based solely on explicit keywords or predefined queries. Instead, the objective is to determine whether a given hypothesis, such as the occurrence of fraudulent behavior, collusion between employees, or policy violations, is supported by the available communication data [6].

This setting can be framed as a hypothesis-driven message retrieval problem, where the objective is to identify emails that are relevant to a given hypothesis, either by directly supporting it or by providing contextual evidence. In contrast to traditional information retrieval tasks, which typically rely on short and well-defined queries [7], forensic communication analysis often begins with investigators formulating hypotheses about topics discussed in communications and seeking evidence to support them [8]. These hypotheses can be understood as complex, case-specific information needs that express abstract and detailed information beyond simple keyword queries [8, 9]. This formulation naturally emphasizes the need for approaches capable of handling richer semantic representations and contextual reasoning [6, 10].

Conventional retrieval approaches, such as keyword-based search or basic semantic similarity methods, exhibit several limitations in this context. First, they rely heavily on lexical overlap between the query and the documents, making them ineffective when relevant emails do not explicitly contain the terms used in the hypothesis. Second, they lack the ability to reason over multiple pieces of information distributed across different messages, although such connections may be important for identifying meaningful evidence in forensic investigations. Third, these methods do not provide mechanisms for contextual interpretation, making it difficult to assess whether a retrieved email truly contributes to validating the hypothesis.

Additionally, the increasing volume of data in modern digital forensic investigations creates significant analytical challenges, including longer processing times, investigative backlogs, and the risk of overlooking relevant evidence [11].

Consequently, the need for intelligent retrieval methods has become increasingly urgent. Recent advances in Large Language Models (LLMs) have

significantly enhanced the ability to understand and reason over unstructured text, enabling new paradigms in information retrieval that integrate generation and retrieval processes [6, 12]. In particular, RAG has emerged as a powerful framework that combines parametric knowledge with external evidence retrieval, improving factual grounding and contextual reasoning in knowledge-intensive tasks [10, 13, 14].

Recent surveys highlight that modern RAG systems extend beyond simple retrieval-generation pipelines by incorporating hybrid retrieval strategies, multi-step reasoning, and adaptive query mechanisms, making them particularly suitable for complex analytical tasks [15, 16]. In this context, hypothesis-driven message retrieval emerges as a natural extension of these developments. Instead of relying on keyword queries, investigators can formulate high-level hypotheses and iteratively retrieve and evaluate supporting evidence, aligning with contemporary views of information retrieval as an interactive and reasoning-driven process [6].

Although this work focuses on email communication, the underlying problem is not limited to emails. Other forms of digital communication, such as instant messages, chat logs, and collaborative messaging platforms, are equally relevant in forensic investigations and may contain critical evidence. Therefore, the proposed hypothesis-driven retrieval perspective can be considered transferable to broader communication data, provided that the data can be represented as text-based messages and analyzed in relation to investigative hypotheses.

A key challenge, therefore, lies in designing a retrieval system that can move beyond surface-level matching and instead capture the semantic and contextual relationship between a hypothesis and the content of emails. Such a system should be capable of interpreting the intent behind a hypothesis, retrieving relevant messages even in the absence of explicit cues, and supporting the investigator by highlighting potential evidence within large collections of data.

1.2 Research Objectives and Proposed Solution

Motivated by these challenges and opportunities, the primary objective of this thesis is to investigate and develop effective methods for the problem of hypothesis-driven message retrieval in large-scale email corpora within a

forensic context. To address this problem, this thesis adopts a RAG framework that combines information retrieval techniques with language model-based reasoning [13]. The proposed approach is designed not only to retrieve relevant emails but also to provide interpretable explanations and supporting evidence for each retrieved result.

The overall system is structured as a multi-stage pipeline. Given a hypothesis expressed in natural language, the system first performs a retrieval step, where relevant email contexts are identified from the corpus and later assessed at the email level. Multiple retrieval strategies are considered, including lexical or sparse retrieval methods such as BM25 [17], dense embedding-based approaches [18], and hybrid retrieval methods, enabling a comparative analysis of their effectiveness in the context of hypothesis-driven retrieval.

Following the retrieval phase, the selected emails are passed to a generation component, where a large language model is used to analyze the relationship between each email and the given hypothesis. This step produces a justification that describes why a particular email may be relevant, allowing the system to move beyond simple document matching toward contextual interpretation [19]. This integration of language model-based reasoning enhances interpretability by generating explanations that describe the relationship between retrieved emails and the given hypothesis.

In addition to generating explanations, the system also performs evidence extraction, identifying specific text spans within each email that support the hypothesis. This capability is particularly important in forensic applications, where verifiability is essential [20]. By highlighting concrete excerpts from the source documents, the system enables investigators to quickly assess the validity of the retrieved information and supports more transparent and verifiable results.

To achieve this aim, this thesis focuses on the following objectives:

- Design and implementation of retrieval pipelines: Develop multiple retrieval approaches, including lexical (e.g., BM25) [17], dense embedding-based [18], and hybrid retrieval methods, integrated within a RAG framework [13].
- Comparative evaluation of retrieval effectiveness: Assess and compare the performance of different retrieval strategies using standard information retrieval metrics such as precision, recall, and F1 scores, in the context of hypothesis-driven email retrieval.

- Integration of language model-based reasoning: Incorporate large language models to generate explanations that describe the relationship between retrieved emails and the given hypothesis, enhancing interpretability.
- Extraction of supporting evidence spans: Identify specific textual segments within emails that serve as supporting evidence for the hypothesis, enabling more transparent and verifiable results.
- Evaluation of explanation and evidence quality: Investigate the effectiveness of LLM-generated reasoning and extracted evidence in supporting the hypothesis, analyzing their accuracy, relevance, and usefulness in a forensic setting.
- Analysis of the interaction between retrieval and reasoning: Explore how the quality of retrieved documents influences the quality of generated explanations and evidence, and whether improved retrieval leads to more reliable interpretability.

The combination of retrieval, reasoning, and evidence extraction results in an integrated framework for interpretable hypothesis-driven retrieval. This approach aims to bridge the gap between traditional information retrieval methods and the needs of forensic analysis, where understanding and justifying results is as important as retrieving them.

1.3 Structure of the Thesis

The remainder of this thesis is organized as follows.

Chapter 2 presents the background and current state of research. It introduces the relevant foundations of digital forensics, e-discovery, information retrieval, and RAG. It also discusses related benchmarks and evaluation frameworks, including legal retrieval, high-recall retrieval, evidence verification, and RAG evaluation, in order to position the present work within existing research.

Chapter 3 describes the dataset construction and annotation process. It introduces the TREC Legal Track and EDRM Enron dataset used in this thesis, explains how the final hypothesis-specific subsets were selected, and presents the three investigative hypotheses derived from selected TREC topics. The chapter also defines the annotation criteria, including the distinction between relevant, partially relevant, and not relevant emails, and explains how these

labels are converted into the binary useful/not-useful setting used in the main evaluation.

Chapter 4 presents the system architecture and implementation. It describes the preprocessing and cleaning of email data, the semantic chunking and refinement process, the creation of sparse and dense representations, and the storage of chunks and emails in a vector database. The chapter then explains the query expansion process, the BM25, dense, and hybrid retrieval pipelines, the aggregation of retrieved chunks at the email level, and the language model-based inference stage for producing usefulness labels, evidence spans, and explanations.

Chapter 5 reports the experimental evaluation, results, and discussion. It first introduces a random baseline and the evaluation setup, then presents the retrieval and inference results for each of the three hypotheses. The chapter compares the performance of BM25, dense retrieval, and hybrid retrieval under default and enriched context settings. It also evaluates reasoning quality and evidence-span quality, and includes a multi-run analysis to examine the stability of the pipeline across different query-expansion runs.

Chapter 6 concludes the thesis by summarizing the main findings, limitations, and directions for future work. It discusses the role of retrieval as the main bottleneck of the pipeline, the sensitivity of query expansion, the conditional usefulness of context enrichment, and the dependence of retrieval performance on the nature of the hypothesis. It also outlines future improvements related to corpus-aware query expansion, structure-aware context selection, reranking, larger annotated datasets, finer-grained evidence annotations, and broader hypothesis types.

Chapter 2

Background and Current State of Research

This chapter presents the theoretical and research background required to situate the work conducted in this thesis. The proposed system combines information retrieval, large language model-based reasoning, and evidence extraction for the task of hypothesis-driven forensic email analysis. Therefore, the chapter first introduces the role of digital forensics and e-discovery in large-scale document review, with particular attention to email communication as an investigative source. It then discusses information retrieval techniques that are commonly used to identify relevant documents, including sparse, dense, and hybrid retrieval methods. After that, the chapter introduces RAG as a framework for combining retrieval with language model-based interpretation. Finally, related benchmarks and evaluation metrics are reviewed in order to clarify how the present thesis differs from existing retrieval, legal reasoning, and RAG evaluation settings.

2.1 Digital Forensics, E-Discovery, and Email Analysis

Digital forensics is concerned with the identification, preservation, analysis, and interpretation of digital evidence. In modern investigations, relevant information is often distributed across large volumes of heterogeneous data, including emails, chat messages, documents, logs, and other digital traces [3]. The role of the investigator is therefore not limited to accessing data, but also

involves determining which pieces of information are meaningful in relation to a specific investigative question. This makes digital forensic analysis both a retrieval problem and an interpretive reasoning problem.

Email communication is a particularly important source of evidence because it often records organizational decisions, informal discussions, negotiations, and exchanges between individuals. In corporate and legal investigations, email collections may reveal how certain decisions were made, whether policies were followed, or whether individuals discussed activities that are relevant to an allegation. However, the evidential value of emails also depends on context. A single message may not explicitly state the behavior being investigated, but may still become relevant when interpreted together with the investigative hypothesis, the surrounding communication, or domain-specific knowledge.

A closely related field is electronic discovery, or e-discovery, where the objective is to identify electronically stored information that is relevant to litigation, regulatory review, or an internal investigation. E-discovery has long been connected to information retrieval research because legal document collections can contain hundreds of thousands or even millions of documents [21]. Manual review of such collections is expensive and time-consuming, which has motivated the development of search, filtering, and technology-assisted review methods [22]. In this setting, recall is often especially important, because failing to identify relevant evidence may have serious legal or investigative consequences. At the same time, precision is also important, since low precision increases the amount of irrelevant material that reviewers must inspect.

The TREC Legal Track is one of the most influential evaluation settings for legal information retrieval and e-discovery. It provided shared tasks, document collections, and relevance judgments for studying how retrieval systems perform in legal review scenarios [21]. The track included tasks based on Enron-derived data and focused on the problem of identifying documents responsive to legal information needs. This makes it particularly relevant to the present thesis, which also uses Enron email data and studies the retrieval of messages in relation to higher-level information needs. However, the task considered in this thesis differs from traditional e-discovery retrieval because the system is not only expected to return relevant documents. It is also expected

to classify whether an email is useful for a hypothesis, extract supporting evidence spans, and generate an explanation of the relationship between the email and the hypothesis.

Technology-Assisted Review (TAR) represents another important line of work in e-discovery. TAR systems use machine learning methods to support document review, often by prioritizing or classifying documents based on examples of relevance [22]. Prior work has shown that machine-assisted review can improve the efficiency of legal review while still maintaining strong effectiveness according to retrieval-oriented measures such as precision, recall, and F1 [22]. This is relevant to the present thesis because both settings involve the use of computational methods to reduce the burden of manual document inspection. Nevertheless, the focus of this thesis is different from conventional TAR. Rather than learning from a large set of coded examples in an iterative review process, the system begins from a natural-language hypothesis and attempts to retrieve and interpret emails that may support or contextualize it.

This distinction is important because investigative hypotheses are often more complex than ordinary keyword queries. A hypothesis may describe an alleged behavior, a financial strategy, or a coordinated action without necessarily using the same words that appear in the relevant emails. Previous work in digital forensic search has also noted that simple text-string searching can be limited when investigators must inspect large result sets and identify meaningful evidence among many possible hits [23]. For example, an investigator may be interested in whether employees discussed manipulating schedules, concealing debt, or destroying documents. Relevant emails may express these ideas indirectly, using domain-specific terminology, abbreviations, or partial references. As a result, hypothesis-driven email analysis requires methods that can go beyond exact lexical matching and support semantic interpretation.

For this reason, the task studied in this thesis can be understood as a specialized form of forensic information retrieval. The input is not a short keyword query, but a richer hypothesis describing an investigative assumption. The desired output is not simply a ranked list of documents, but a set of emails that are useful for assessing the hypothesis, together with evidence spans and explanations that make the result interpretable. This places the work at the intersection of digital forensics, e-discovery, information retrieval, and retrieval-augmented language model reasoning.

2.2 Information Retrieval for Evidence Discovery

Information retrieval (IR) is concerned with finding documents or passages that satisfy an information need expressed by a user query [7]. In standard search settings, this information need is often represented by a short keyword query. In forensic and e-discovery settings, however, the information need may be more complex. Investigators may not know the exact words used in the relevant documents, and relevant evidence may be expressed indirectly, through domain-specific terminology, abbreviations, or contextual references. For this reason, retrieval in investigative settings must support both exact matching and broader semantic matching.

In the context of this thesis, retrieval is the first stage of the analysis pipeline. Given a hypothesis expressed in natural language, the retrieval component identifies candidate email chunks that may contain useful evidence. These chunks are then grouped at the email level and passed to the inference component for further analysis. The quality of retrieval is therefore critical, because the language model can only reason over evidence that is included in its input context. If a relevant email is not retrieved, the generation stage cannot extract evidence from it or explain its relevance.

2.2.1 Sparse Retrieval

Sparse retrieval methods represent queries and documents using lexical features, usually based on the words that appear in the text. These approaches are often implemented with inverted indexes, which make it efficient to retrieve documents containing query terms [7]. A central advantage of sparse retrieval is that it is transparent and effective when relevant documents share explicit vocabulary with the query. This is particularly useful in legal and forensic contexts, where names, dates, technical terms, transaction identifiers, and specific phrases may be highly informative.

One of the most widely used sparse retrieval models is BM25, which is based on the probabilistic relevance framework [17]. BM25 ranks documents according to the occurrence of query terms, while also accounting for term frequency, inverse document frequency, and document length normalization. A common form of the BM25 scoring function is shown in Equation 2.1:

$$\text{BM25}(q, d) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{f(t, d)(k_1 + 1)}{f(t, d) + k_1 \left(1 - b + b \frac{|d|}{\text{avgdl}}\right)} \tag{2.1}$$

where q is the query, d is the document, $f(t, d)$ is the frequency of term t in document d, $|d|$ is the document length, avgdl is the average document length in the collection, and k_1 and b are parameters controlling term-frequency saturation and document-length normalization [17].

The strength of BM25 lies in its robustness and its ability to retrieve documents that contain exact lexical cues. In forensic email analysis, this can be especially valuable when evidence depends on specific expressions such as financial terms, employee names, project names, or operational terminology. However, sparse retrieval also has important limitations. It relies heavily on lexical overlap between the query and the document. If a hypothesis uses abstract wording while the relevant emails use different terminology, BM25 may fail to retrieve useful evidence. This limitation is often described as a vocabulary mismatch problem in information retrieval [7].

2.2.2 Dense Retrieval

Dense retrieval methods address some of the limitations of sparse retrieval by representing queries and documents as dense vectors in a continuous embedding space. Instead of relying only on exact word overlap, dense retrieval uses neural models to encode semantic meaning, so that texts with similar meanings can be close to each other even when they do not share the same vocabulary. This makes dense retrieval attractive for tasks where the user's information need is abstract, implicit, or expressed in different language from the relevant documents.

A common architecture for dense retrieval is the dual-encoder model, where the query and the document are encoded separately and compared using a similarity function such as dot product or cosine similarity. Dense Passage Retrieval (DPR), for example, showed that learned dense representations can be effective for retrieving passages in open-domain question answering [18]. Sentence-level and passage-level embedding models have also become widely used for semantic search because they allow efficient comparison between text units in vector space [24].

In hypothesis-driven forensic email retrieval, dense retrieval is useful because the hypothesis may describe a behavior rather than a fixed set of keywords. For instance, a hypothesis about concealing debt, manipulating supply, or coordinating document retention may correspond to emails that use indirect or domain-specific wording. Dense retrieval can therefore help identify messages that are semantically related to the hypothesis even when they do not contain the same surface terms.

Nevertheless, dense retrieval also has limitations. It may retrieve semantically similar but investigatively irrelevant texts, especially when the hypothesis is broad. It may also be less reliable for exact matching of rare terms, identifiers, names, or highly specific phrases. In forensic settings, this is an important concern because small lexical details can be evidentially significant. Therefore, dense retrieval should not be understood as a complete replacement for sparse retrieval, but rather as a complementary approach.

2.2.3 Hybrid Retrieval

Hybrid retrieval combines sparse and dense retrieval methods in order to benefit from the strengths of both approaches. Sparse retrieval is effective for exact lexical matches, while dense retrieval is better suited for semantic similarity and vocabulary mismatch. Combining the two can therefore improve retrieval robustness, especially when the information need contains both precise terms and abstract concepts.

One simple and widely used approach for combining ranked lists is Reciprocal Rank Fusion (RRF) [25]. RRF assigns a score to each document based on its rank in multiple retrieval result lists. A document that appears highly ranked by several retrieval methods receives a stronger combined score. The general form of RRF is shown in Equation 2.2:

$$\mathrm{RRF}(d) = \sum_{r \in R} \frac{1}{k + \mathrm{rank}_r(d)} \tag{2.2}$$

where R is the set of retrieval systems, $\mathrm{rank}_r(d)$ is the rank of document d in the result list produced by retriever r, and k is a constant used to reduce the impact of lower-ranked documents [25].

Hybrid retrieval is particularly relevant for the task considered in this thesis. Some emails may be retrieved because they contain explicit lexical evidence, while others may be useful because they are semantically related to the

hypothesis despite using different wording. By combining sparse and dense methods, the retrieval stage can cover a wider range of possible evidence. This is important in forensic analysis, where missing relevant material may be more problematic than retrieving additional candidates for later inspection.

2.2.4 Query Expansion

Query expansion is another technique used to improve retrieval effectiveness. The basic idea is to reformulate or enrich the original query with additional related terms, phrases, or descriptions. Classical approaches include relevance feedback and pseudo-relevance feedback, where information from known or assumed relevant documents is used to modify the query [26, 27]. Query expansion can reduce vocabulary mismatch by increasing the chance that the query contains terms that appear in relevant documents.

In the setting of hypothesis-driven retrieval, query expansion is especially useful because the original hypothesis may be long, abstract, or written in language that differs from the emails. A single hypothesis can imply several possible search directions. For example, a hypothesis about concealing debt may be expanded into queries involving loans, financing, balance sheet treatment, accounting structure, or special-purpose entities. Similarly, a hypothesis about manipulating energy supply may be expanded into queries involving schedules, bids, load volumes, availability, and market prices.

Recent work has explored the use of large language models for query expansion. Query2doc, for instance, uses an LLM to generate pseudo-documents that expand the original query and improve both sparse and dense retrieval [28]. HyDE follows a related idea by generating a hypothetical document from the query and then using its embedding for dense retrieval [29]. More recent work has also investigated corpus-steered and LLM-assisted pseudo-relevance feedback methods, where corpus evidence and language models are used to refine or expand the original query [54, 55]. These approaches are relevant because they show that generative models can help bridge the gap between a short or abstract information need and the documents that should be retrieved.

For this thesis, query expansion is used to transform an investigative hypothesis into multiple retrieval queries. This makes the retrieval stage less dependent on a single formulation of the hypothesis and allows the system

to search for different expressions of the same underlying idea. In a forensic setting, this is important because the same behavior may be described using different vocabulary across different emails, authors, or organizational contexts.

2.3 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a framework that combines external information retrieval with neural text generation. Instead of relying only on the parametric knowledge stored inside a language model, a RAG system first retrieves relevant documents or passages from an external collection and then conditions the generation process on the retrieved context [13]. This design is especially useful in knowledge-intensive tasks, where the system must produce outputs that are grounded in specific source material rather than only in the model's internal knowledge.

The general RAG pipeline consists of two main components: a retriever and a generator. Given an input query x, the retriever selects a set of relevant documents or passages z from a corpus. The generator then produces an output y conditioned on both the original query and the retrieved context. In the original formulation of RAG, the output probability is modeled by marginalizing over retrieved documents as shown in Equation 2.3:

$$p(y|x) \approx \sum_{z \in \text{top-}k(x)} p_\eta(z|x) p_\theta(y|x, z) \tag{2.3}$$

where $p_\eta(z|x)$ represents the retriever distribution over documents and $p_\theta(y|x, z)$ represents the generator distribution conditioned on the query and retrieved document [13]. Although modern RAG systems are often implemented in more flexible pipeline-based forms, this formulation captures the central idea that generation should be informed by retrieved evidence.

RAG is closely related to earlier work on retrieval-augmented language modeling and open-domain question answering. REALM, for example, introduced a retrieval-augmented pre-training approach in which a language model retrieves knowledge from a textual corpus while learning to answer knowledge-intensive questions [30]. Dense Passage Retrieval (DPR) further

showed that dense neural retrievers can be used to retrieve passages for open-domain question answering, replacing or complementing traditional sparse retrieval methods [18]. Later architectures such as Fusion-in-Decoder demonstrated that generation quality can improve when the model is able to condition on multiple retrieved passages and integrate information across them [19]. These developments established RAG as a central approach for tasks that require both access to external knowledge and flexible natural language reasoning.

2.3.1 RAG for Evidence-Grounded Analysis

The main advantage of RAG is that it provides a mechanism for grounding model outputs in retrieved evidence. This is important because large language models can generate fluent and plausible text even when their outputs are not supported by reliable sources. By retrieving relevant documents and conditioning the generation step on them, RAG systems can reduce the dependence on the model's internal knowledge and make the output more verifiable [13]. In domains such as law, medicine, finance, and digital forensics, this grounding is especially important because users need to inspect the source material and verify whether a generated conclusion is justified.

In the context of this thesis, RAG is not used only for answering natural language questions. Instead, it is adapted to the task of hypothesis-driven forensic email analysis. The input is an investigative hypothesis, and the retrieval component identifies email chunks that may contain relevant evidence. The generation component then analyzes whether each retrieved email-level context is useful for assessing the hypothesis, extracts supporting text spans, and produces an explanation of the relationship between the email and the hypothesis. Therefore, the role of generation is not merely to produce an answer, but to support interpretation and evidence assessment.

This use of RAG is connected to broader work on explainable and evidence-grounded NLP. In evidence-based tasks, it is not sufficient for a model to produce a correct label or answer; it should also identify the source evidence that supports the output. This idea is central to fact verification datasets such as FEVER, where systems must determine whether a claim is supported or refuted by textual evidence and identify the relevant supporting sentences [20]. Although FEVER is based on Wikipedia rather than forensic

emails, it illustrates the importance of linking model judgments to concrete textual evidence.

For forensic applications, evidence grounding has an additional practical importance. Investigators must be able to inspect why a document was considered relevant and whether the reasoning is supported by the original text. A system that only outputs a relevance label may reduce the amount of manual review, but it does not fully support transparent decision-making. By contrast, a system that provides both an evidence span and an explanation allows the user to evaluate the result more directly. This makes RAG suitable for investigative tasks where interpretability and verifiability are as important as retrieval performance.

2.3.2 RAG Variants and Recent Developments

Recent research has extended the basic retrieve-then-generate pipeline in several ways. Some approaches focus on improving the retrieval stage, for example through better dense retrievers, query expansion, re-ranking, or hybrid retrieval. Others focus on improving the interaction between retrieval and generation. Self-RAG introduces a self-reflective framework in which the model learns to decide when retrieval is needed and to critique the usefulness and supportiveness of retrieved passages [31]. This is relevant because not every generated response benefits equally from retrieval, and not every retrieved passage is useful for the final output.

Corrective RAG follows a related motivation by introducing mechanisms to evaluate and correct retrieved information before generation [32]. The goal is to reduce the risk that irrelevant or misleading retrieved passages negatively affect the generated answer. Such approaches are important because retrieval is not automatically beneficial: if the retrieved context is noisy, incomplete, or only weakly related to the query, the generator may produce unsupported or incorrect conclusions.

Other recent work has emphasized adaptive and iterative retrieval. In these systems, retrieval may happen more than once, and the query may be reformulated based on intermediate results. This is particularly relevant for complex information needs, where a single query may not be sufficient to retrieve all necessary evidence. In hypothesis-driven analysis, the same issue appears because a hypothesis can contain several sub-concepts, and relevant emails may

express those concepts in different ways. Query expansion and multi-query retrieval can therefore be seen as practical mechanisms for adapting RAG to complex investigative hypotheses.

2.3.3 Limitations of RAG in Investigative Settings

Despite its advantages, RAG does not guarantee correct or complete analysis. The first major limitation is retrieval dependency. If the retrieval stage fails to retrieve a relevant email, the generation stage cannot reason over it. This is especially important in forensic and e-discovery settings, where missing relevant evidence can be more problematic than retrieving additional irrelevant material. Therefore, recall-oriented evaluation remains important even when a language model is used after retrieval.

A second limitation concerns noisy or partially relevant context. Retrieval systems may return documents that share some terms or semantic similarity with the hypothesis but do not actually provide useful evidence. When such context is passed to a language model, the model may over-interpret weak evidence or produce explanations that appear plausible but are not fully supported by the source text. This makes it necessary to evaluate not only whether the system retrieves relevant documents, but also whether the generated reasoning is faithful to the retrieved evidence.

A third limitation is context length. Email threads, attachments, and long messages may contain more information than can be included in the model input. Chunking is therefore necessary, but chunking can separate related information across different text segments. If the retrieved chunk is too narrow, the model may lack the surrounding context needed to interpret the evidence correctly. If the retrieved context is too broad, the model may receive irrelevant information that makes the inference more difficult. This creates a trade-off between context completeness and context precision.

A fourth limitation is the difficulty of evaluating generated explanations. Standard retrieval metrics such as precision and recall can measure whether relevant documents were retrieved, but they do not directly measure whether the generated explanation is correct, faithful, or useful. For this reason, RAG evaluation often requires additional dimensions such as faithfulness, answer relevance, context relevance, and evidence quality [33]. In the present thesis, this issue is addressed by evaluating retrieval performance, useful/not-useful

inference labels, evidence-span quality, and reasoning quality as separate components.

These limitations show why RAG should be understood as an evidence-support system rather than a fully automated substitute for forensic judgment. In an investigative setting, the purpose of RAG is to help identify potentially useful material, highlight supporting evidence, and explain why a message may be relevant to a hypothesis. The final assessment still depends on whether the retrieved evidence and generated reasoning can be verified against the original documents.

2.4 Related Benchmarks and Similar Work

The task addressed in this thesis is related to several existing research areas, including legal information retrieval, e-discovery, high-recall retrieval, open-domain retrieval, fact verification, legal language model evaluation, and RAG evaluation. However, none of these benchmarks fully matches the complete setting considered here: retrieving emails in response to an investigative hypothesis, classifying their usefulness, extracting evidence spans, and generating explanations grounded in the retrieved text. This section reviews the most relevant benchmarks and lines of work and clarifies how they relate to the present thesis.

2.4.1 Legal and E-Discovery Benchmarks

The closest research area to this thesis is legal information retrieval and e-discovery. The TREC Legal Track is one of the most influential shared evaluation efforts in this area. It was designed to evaluate retrieval systems for legal discovery tasks, where systems must identify documents responsive to legal information needs [21]. The track is particularly relevant because it used large legal document collections, including Enron-derived material, and provided relevance assessments for evaluating retrieval effectiveness. This makes it an important precedent for studying search and review methods over corporate email collections.

The connection between the TREC Legal Track and the present thesis is direct but not complete. Both settings involve large-scale document review

and the identification of documents relevant to an information need. However, the TREC Legal Track primarily evaluates retrieval effectiveness in an e-discovery setting, whereas this thesis studies a RAG-based pipeline that also performs language model-based interpretation. In other words, the objective here is not only to retrieve responsive documents, but also to determine whether retrieved emails are useful for assessing a hypothesis, extract textual evidence, and generate explanations that make the result inspectable.

Technology-Assisted Review (TAR) is another closely related line of work. TAR refers to the use of machine learning and information retrieval methods to support legal document review, often by prioritizing or classifying documents according to their likely relevance [22]. TAR is relevant to this thesis because both settings aim to reduce the burden of manual review in large document collections. In legal review, the goal is often to identify relevant documents as efficiently as possible while maintaining high recall. Similarly, in forensic email analysis, investigators need methods that can reduce the search space while minimizing the risk of overlooking useful evidence.

Nevertheless, conventional TAR differs from the task considered in this thesis. TAR systems typically rely on reviewer feedback, training examples, or iterative relevance assessment. By contrast, the approach in this thesis starts from a natural-language investigative hypothesis and uses retrieval and generation to identify and interpret potentially useful emails. The system is therefore closer to hypothesis-driven evidence discovery than to standard supervised document review.

The TREC Total Recall Track is also relevant because it focuses on high-recall retrieval. Its objective was to evaluate systems that attempt to find as many relevant documents as possible, ideally approaching complete recall, while minimizing review effort [34]. This is conceptually important for forensic and legal settings, where missing relevant material can be costly. The present thesis shares this recall-oriented concern, especially at the retrieval stage, because an email that is not retrieved cannot later be analyzed by the language model. However, Total Recall focuses mainly on the retrieval and review process, whereas this thesis also evaluates downstream inference, evidence extraction, and explanation quality.

2.4.2 General Retrieval Benchmarks

Beyond legal retrieval, this thesis is also related to general information retrieval benchmarks. BEIR is a widely used benchmark for evaluating retrieval models across diverse datasets and domains [35]. It includes several retrieval tasks and has been used to compare sparse retrieval methods such as BM25 with dense retrieval models and other neural approaches. BEIR is relevant because it shows the importance of evaluating retrieval methods across different types of information needs rather than assuming that one retrieval approach will perform best in every domain.

The relevance of BEIR to this thesis lies mainly in its comparison of retrieval paradigms. The present thesis also compares sparse, dense, and hybrid retrieval methods. However, BEIR is not designed for forensic email analysis or hypothesis-driven evidence discovery. Its tasks are mostly general-domain or domain-specific retrieval tasks with predefined queries and relevance labels. In contrast, this thesis studies hypotheses that describe investigative assumptions and require emails to be judged according to their usefulness for supporting or contextualizing those assumptions.

MS MARCO is another important benchmark in modern retrieval research. It was introduced for large-scale machine reading comprehension and passage ranking using real user queries [36]. MS MARCO has been widely used for training and evaluating dense retrievers and neural ranking models. It is relevant because many modern retrieval models and embedding approaches are developed or evaluated in relation to passage-ranking tasks of this kind.

However, MS MARCO differs substantially from the setting of this thesis. It is primarily based on web-style questions and passage retrieval, whereas the present work deals with corporate email communication and forensic hypotheses. The information need in MS MARCO is usually a question seeking an answer, while the information need in this thesis is an investigative hypothesis seeking supporting or contextual evidence. Therefore, MS MARCO is useful as background for retrieval research, but it is not a direct benchmark for the task studied here.

2.4.3 Evidence Verification and Legal Reasoning Benchmarks

The evidence extraction component of this thesis is related to fact verification and evidence-grounded NLP. FEVER is one of the most influential benchmarks in this area. In FEVER, systems must determine whether a claim is supported, refuted, or not verifiable based on evidence from Wikipedia, and they must identify the textual evidence that supports the decision [20]. This is relevant because the present thesis also requires a model to connect a higher-level statement to textual evidence.

The similarity between FEVER and this thesis is that both require a judgment to be grounded in evidence. However, the domains and tasks are different. FEVER focuses on claim verification over Wikipedia, while this thesis focuses on forensic email analysis. In FEVER, the input is usually a factual claim, and the system determines whether it is supported or refuted. In this thesis, the input is an investigative hypothesis, and the system must identify whether an email is useful for assessing that hypothesis. The output is also different: this thesis evaluates not only the label, but also the quality of the extracted evidence span and the generated explanation.

LegalBench is another related benchmark, but from the perspective of legal language model evaluation. It provides a collection of tasks designed to evaluate legal reasoning and legal understanding in language models [37]. This is relevant because the present thesis also operates near the legal and investigative domain and uses language models for interpretation. However, LegalBench is mainly concerned with legal reasoning tasks, not retrieval over large email corpora. It does not directly evaluate whether a system can retrieve relevant evidence from a forensic collection and generate explanations grounded in that evidence.

2.4.4 RAG Evaluation Benchmarks and Frameworks

Since this thesis uses a RAG-based pipeline, it is also related to recent work on evaluating RAG. RAGAS is a framework for evaluating RAG systems along dimensions such as faithfulness, answer relevance, context precision, and context recall [33]. This is relevant because it recognizes that RAG systems cannot be evaluated only by the final generated answer. The quality of the retrieved context and the faithfulness of the generated output to that context are

also important.

This idea is closely aligned with the evaluation design of the present thesis. The system is evaluated not only in terms of retrieval precision, recall, and F1, but also in terms of inference labels, evidence-span quality, and reasoning quality. However, the evaluation in this thesis is adapted to the forensic hypothesis-driven setting. Instead of evaluating general answer quality, the focus is on whether the retrieved and generated outputs help determine whether an email is useful for an investigative hypothesis.

Recent work has also argued that retrieval evaluation in RAG systems should consider the effect of retrieved documents on downstream generation. For example, eRAG examines retrieval quality from the perspective of how retrieved documents contribute to generation, rather than relying only on traditional query-document relevance labels [38]. This is important for the present thesis because a retrieved email may be topically related to the hypothesis but still not useful for generating a correct and evidence-grounded inference. Conversely, a document may be useful because it provides a small but important piece of evidence.

This distinction motivates the multi-stage evaluation used in this thesis. Retrieval metrics measure whether useful emails are retrieved, but they do not fully determine whether the language model correctly interprets those emails. Therefore, the downstream inference and evidence extraction stages must be evaluated separately. This reflects the broader view that RAG systems should be assessed as pipelines, where retrieval quality and generation quality interact but are not identical.

2.4.5 Comparative Summary of Related Benchmarks

The benchmarks and evaluation frameworks reviewed in the preceding subsections address different components of the task considered in this thesis. Table 2.1 summarizes their primary evaluation focus and their relationship to hypothesis-driven forensic email analysis.

Table 2.1: Summary of related benchmarks and evaluation frameworks.

Resource	Primary focus	Relation to the present thesis
TREC Legal Track	Retrieval of documents responsive to legal information needs	Closely related to the corporate email and e-discovery setting, but does not evaluate usefulness classification, evidence spans, or generated explanations.
TREC Total Recall	High-recall retrieval with reduced review effort	Reflects the importance of finding as much relevant evidence as possible, but focuses mainly on retrieval and document review.
BEIR	Comparison of retrieval models across multiple domains	Relevant to the comparison of sparse and dense retrieval, but is not designed for forensic emails or investigative hypotheses.
MS MARCO	Passage ranking for natural-language web queries	Important for modern retrieval research, but its answer-seeking queries differ from hypotheses seeking supporting or contextual evidence.
FEVER	Claim verification and supporting-evidence identification	Related to evidence-grounded decisions, but operates over Wikipedia claims rather than forensic email collections.
LegalBench	Evaluation of legal understanding and reasoning in language models	Relevant to legal-domain interpretation, but does not evaluate retrieval from a document corpus or hypothesis-driven evidence discovery.
RAGAS	Evaluation of context quality, answer relevance, and faithfulness in RAG	Motivates the separation of retrieval and generation quality, but is primarily designed for general question-answering settings.
eRAG	Evaluation of retrieved documents according to their contribution to generation	Connects retrieval quality with downstream generation, but does not address hypothesis-driven forensic email analysis.

As shown in Table 2.1, the reviewed resources cover individual aspects of the present task, including legal retrieval, high-recall review, sparse and dense retrieval, evidence verification, legal reasoning, and RAG evaluation. However, they differ in their input, document domain, expected output, and evaluation focus. None of them jointly evaluates the retrieval of forensic emails from an investigative hypothesis, the classification of their usefulness, the extraction of supporting evidence spans, and the generation of grounded explanations.

2.5 Evaluation Metrics

Evaluation is an essential part of information retrieval and RAG-based systems because different stages of the pipeline may fail in different ways. A system may retrieve useful documents but generate an incorrect explanation, or it may generate a plausible explanation from context that does not actually contain sufficient evidence. For this reason, the evaluation of hypothesis-driven forensic email analysis should not be reduced to a single score. Instead, retrieval quality, classification quality, evidence grounding, and explanation quality need to be assessed separately.

In this thesis, the evaluation follows this multi-component perspective. Retrieval is evaluated according to whether useful emails are retrieved. The inference stage is evaluated according to whether the language model correctly classifies emails as useful or not useful. The evidence-span component is evaluated according to whether the extracted spans are present in the retrieved context and whether they support the hypothesis. Finally, the reasoning component is evaluated according to whether the generated explanation is relevant, correct, and clear.

2.5.1 Retrieval Metrics

Retrieval metrics measure how effectively a system identifies relevant items from a collection in response to an information need. In classical information retrieval, systems are often evaluated using ranked retrieval metrics such as Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), or Normalized Discounted Cumulative Gain (nDCG) [7]. These metrics are useful when the position of each retrieved document in a ranked list is important.

However, in the setting of this thesis, the main objective is to determine whether useful emails are retrieved at all, rather than to evaluate the exact rank order of every retrieved email. Retrieved chunks are grouped at the email level, and the downstream inference component receives email-level contexts. Therefore, set-based metrics such as precision, recall, and F1 are more appropriate for the main evaluation.

Let G denote the set of ground-truth useful emails for a hypothesis, and let R denote the set of emails retrieved by a retrieval method. Retrieval precision is defined as:

$$\text{Precision} = \frac{|R \cap G|}{|R|} \tag{2.4}$$

Precision measures the proportion of retrieved emails that are actually useful. High precision means that the retrieval method returns little irrelevant material. In forensic analysis, this is important because low precision increases the amount of material that an investigator or downstream model must inspect.

Retrieval recall is defined as:

$$\text{Recall} = \frac{|R \cap G|}{|G|} \tag{2.5}$$

Recall measures the proportion of all useful emails that were successfully retrieved. In forensic and e-discovery settings, recall is particularly important because missing useful evidence may affect the completeness of the investigation. If a useful email is not retrieved, the language model cannot later extract evidence from it or explain its relationship to the hypothesis.

The F1 score combines precision and recall using the harmonic mean:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.6}$$

F1 is useful when both precision and recall matter. It provides a single summary measure that penalizes systems that perform well on only one of the two dimensions. Nevertheless, precision and recall should still be interpreted separately, especially in investigative settings where the relative importance of finding all useful evidence and reducing review burden may vary.

2.5.2 Classification Metrics

After retrieval, the language model classifies each retrieved email-level context as useful or not useful for the given hypothesis. This can be evaluated as a binary classification problem. In this thesis, the useful class is treated as the positive class. Relevant and partially relevant emails, which will be explained in the upcoming chapters, are therefore useful as well, while not relevant emails are considered not useful.

The predictions can be summarized using a confusion matrix. The four entries of the confusion matrix are:

- True Positives (TP): useful emails correctly predicted as useful.
- False Positives (FP): not useful emails incorrectly predicted as useful.
- False Negatives (FN): useful emails incorrectly predicted as not useful.
- True Negatives (TN): not useful emails correctly predicted as not useful.

Using these quantities, classification precision is defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.7}$$

Classification precision measures how often emails predicted as useful are truly useful. This is important because a high number of false positives can reduce trust in the system and increase the amount of unnecessary review.

Classification recall is defined as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.8}$$

Classification recall measures how many of the truly useful emails are included in the final predicted-useful set. In the end-to-end evaluation used in this thesis, low recall can result either because a useful email was not retrieved or because the inference module classified a retrieved useful email as not useful.

The classification F1 score is again computed as the harmonic mean of precision and recall:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.9}$$

These metrics therefore measure the combined effect of retrieval and downstream usefulness classification on the final pipeline output.

2.5.3 Evidence-Span Quality

In evidence-grounded systems, it is not sufficient to produce a label or an explanation. The system should also identify the textual evidence on which the decision is based. This is important in forensic settings because investigators must be able to verify the connection between the system output and the original document. Similar motivations appear in fact-verification tasks such as FEVER, where systems are evaluated not only for claim classification but also for their ability to identify supporting evidence [20].

In this thesis, the language model extracts evidence spans from the retrieved email-level context. These spans are evaluated using two criteria: faithfulness and supportiveness.

Faithfulness measures whether the extracted span is actually present in the retrieved context. A faithful evidence span should be an exact or directly verifiable quotation from the source text. This criterion checks whether the model grounded its output in the available context rather than producing an unsupported or hallucinated excerpt. Faithfulness is closely related to the broader RAG evaluation concern that generated outputs should be supported by retrieved context [33].

Supportiveness measures whether the extracted span provides meaningful support for the usefulness label in relation to the hypothesis. A span may be faithful because it appears in the email, but still fail to be supportive if it is too generic, unrelated, or insufficient to justify the hypothesis-level interpretation. For example, a span containing a financial term may be present in the email, but it is only supportive if it contributes to the specific hypothesis being evaluated.

These two criteria are intentionally separated. Faithfulness asks whether the span exists in the context, while supportiveness asks whether the span actually helps justify the decision. This distinction is important because a model can quote text correctly while still selecting evidence that does not meaningfully support the hypothesis.

2.5.4 Reasoning Quality

The reasoning component explains why an email is or is not useful for assessing the hypothesis. This explanation is important because the system is intended to support investigative analysis rather than simply output a binary

decision. A useful explanation should make clear how the evidence relates to the hypothesis, avoid unsupported claims, and be understandable to a human reviewer. The reasoning quality is evaluated using three criteria: relevance, correctness, and clarity.

Relevance measures whether the explanation addresses the relationship between the email context and the hypothesis. An explanation is relevant if it focuses on the investigative issue expressed in the hypothesis rather than discussing unrelated details from the email.

Correctness measures whether the explanation accurately represents the retrieved context. An explanation may be relevant but still incorrect if it overstates the evidence, introduces claims not present in the email, or misinterprets the meaning of the retrieved text. This criterion is connected to the broader problem of faithfulness in generated text, where a model's output should remain grounded in the available evidence [33].

Clarity measures whether the explanation is understandable and logically expressed. In forensic analysis, clarity is important because explanations are meant to help reviewers inspect and assess system outputs. A technically correct explanation may still be less useful if it is vague, confusing, or poorly structured.

LLM-based evaluation has increasingly been used to assess open-ended generated text, especially when exact reference answers are unavailable or when evaluation requires judgment over qualities such as relevance, coherence, and factual consistency [39]. However, LLM-based evaluation also has limitations. Automated judges may be sensitive to prompt wording, may show bias toward fluent explanations, and may not always capture domain-specific standards. Therefore, reasoning-quality scores should be interpreted as structured diagnostic assessments rather than absolute measures of forensic validity.

2.5.5 Macro-Averaging Across Hypotheses

Because this thesis evaluates three separate hypotheses, results can be reported both per hypothesis and as macro-averages across hypotheses. Macro-averaging gives equal weight to each hypothesis, regardless of the number of useful emails or the difficulty of the hypothesis. If M_i denotes a metric score for hypothesis i, and there are N hypotheses, the macro-average

is:

$$\text{MacroAverage} = \frac{1}{N}\sum_{i=1}^{N} M_i \tag{2.10}$$

Macro-averaging is useful because each hypothesis represents a distinct investigative information need. Reporting only aggregate counts across all emails could hide differences between hypotheses. For example, a retrieval method may perform well on a hypothesis with strong lexical cues but poorly on a hypothesis where the relevant evidence is expressed more indirectly. Per-hypothesis reporting and macro-averaging therefore provide a more balanced view of system behavior.

2.6 Research Gap and Positioning of This Thesis

As summarized in Table 2.1, existing research addresses several individual components of the problem considered in this thesis. E-discovery and TAR mainly focus on identifying responsive documents, while general retrieval benchmarks evaluate retrieval effectiveness for predefined queries. Fact-verification benchmarks connect predictions to textual evidence, and RAG evaluation frameworks assess the quality and faithfulness of retrieved and generated outputs. However, these research areas do not fully address forensic email analysis based on natural-language investigative hypotheses.

The research gap therefore lies at the intersection of legal and forensic retrieval, evidence-grounded language modeling, and hypothesis-driven email analysis. The task is not limited to retrieving documents or answering a question. It requires identifying emails that are useful for assessing an investigative hypothesis, extracting the textual evidence that supports the assessment, and explaining how the email content relates to the hypothesis.

The contribution of this thesis is not the introduction of a new retrieval paradigm or a new large language model. Instead, it lies in the design, implementation, and evaluation of a retrieval-augmented pipeline adapted to this specific forensic analysis setting. The pipeline combines sparse, dense, and hybrid retrieval, query expansion, email-level context construction, language model-based usefulness classification, evidence-span extraction, and explanation generation. This design makes it possible to investigate not only which

retrieval method identifies more useful emails, but also how the retrieved context affects downstream interpretation and evidence extraction.

A further distinguishing feature is the multi-component evaluation design. Rather than evaluating the pipeline through a single final score, the evaluation separates retrieval performance, usefulness classification, evidence-span quality, and reasoning quality. This distinction is necessary because the pipeline may fail at different stages: a useful email may not be retrieved, a retrieved email may be misclassified, a quoted span may not support the decision, or an explanation may misrepresent or overstate the available evidence. Evaluating these components separately provides a clearer account of the interaction between retrieval and language model-based reasoning.

Overall, this thesis positions hypothesis-driven forensic email analysis as an evidence-discovery task that combines information retrieval with language model-based interpretation. Its purpose is to reduce the document search space, identify potentially useful emails, and make system decisions inspectable through quoted evidence and explanations. The proposed pipeline should therefore be understood as an investigative support system rather than a replacement for human forensic judgment.

Chapter 3

Data

This chapter describes the data used in this thesis and explains how the final hypothesis-specific email dataset was constructed. It first introduces the TREC Legal Track and the EDRM Enron Email Data Set Version 2, which provide the source material for the experiments. Since the original corpus is too large for direct manual annotation in the context of this thesis, a smaller task-specific subset was created using the seed sets from the TREC 2009 Legal Track Interactive Task. The chapter then explains the selection of three legally and forensically relevant topics, the derivation of the corresponding hypotheses, and the filtering process used to obtain the final email sets. Finally, it presents the annotation criteria, label definitions, annotation workflow, and the conversion of the original three-label annotation scheme into the binary useful/not-useful setting used for evaluation in the remainder of the thesis.

3.1 Data Description

3.1.1 TREC and EDRM Enron Dataset

The experiments in this thesis are conducted using data from the TREC Legal Track collection. The Text Retrieval Conference (TREC) is a long-running evaluation initiative that provides standardized datasets, tasks, and relevance judgments for information retrieval research [40]. Within TREC, the TREC Legal Track focuses specifically on e-discovery scenarios, where the goal is to identify legally relevant documents from large collections [41].

EDRM Enron Email Data Set Version 2 was released through a collaboration between the Electronic Discovery Reference Model, ZL Technologies,

and the TREC Legal Track community. The release was intended to provide a large-scale, publicly available corporate email corpus to support research in e-discovery, information retrieval, and related analytical domains [41].

The collection consists of Enron corporate email communications that became publicly available following investigations into Enron [42]. As documented in the TREC 2010 Legal Track overview, the original messages were obtained by the Federal Energy Regulatory Commission and later processed for research use by EDRM in collaboration with ZL Technologies [21]. This provenance makes the dataset particularly valuable, as it reflects authentic organizational communication rather than synthetic data, thereby supporting realistic investigative and legal retrieval studies.

The EDRM Enron Version 2 dataset contains over 1.7 million documents, distributed in multiple formats, and including emails and their associated attachments [43]. The XML format used in this thesis contains text renderings of each email message and attachment, as well as the original native format [21].

For the TREC 2010 Legal Track, EDRM dataset was refined into a controlled corpus for legal retrieval experiments, with deduplication playing a central role [44]. This process identified 455,449 distinct canonical messages which, together with 230,143 attachments, resulted in a final collection of 685,592 documents [21]. Additional resources, including canonical mappings, candidate document sets, and de-duplicated text renderings, were provided to support systematic and reproducible evaluation [44].

3.1.2 Limitations and Motivations

The decision to construct a dedicated subset for this thesis was primarily motivated by the mismatch between the scale and structure of existing Enron-based resources and the requirements of hypothesis-driven retrieval. While the full EDRM Enron v2 dataset is extremely large, the TREC 2010 Legal Track collection derived from it still contains over 600,000 documents [21], making direct manual annotation for a new task impractical, particularly when document-level relevance judgments are required.

Existing Enron-based studies also present limitations for this work. For example, research by Robert F. McCullough on Enron's strategies during the 2002–2003 California energy crisis identifies several behavioral patterns

among employees that could inform forensic hypothesis design [45]; however, it does not provide a reusable annotated subset. Similarly, the UC Berkeley Enron Email Analysis project focuses on classification tasks over a subset of emails, but its annotation scheme is not aligned with the investigative objectives of this thesis [46].

These limitations motivated the creation of a new, task-specific annotated resource. A key opportunity emerged from the structure of the TREC 2009 Legal Track itself. In the Interactive Task, a subset of more than 8,000 documents from the Enron corpus was assessed against a set of legally motivated topics. These topics included prepay transactions, financial forecasts, document destruction, energy schedules and bids, analyst communications, as well as informal categories such as fantasy-football-related communications [41].

Furthermore, the Interactive Task provided the results of their work as a CSV file, referred to as seed sets. This file contained the document identifier, topic number, and assessment value indicating whether each document was responsive to the corresponding topic [44]. This framework provides narrower entry points into the corpus, as it enabled the use of a smaller, pre-examined, topic-linked subset of the corpus, rather than relying solely on the full unannotated collection.

3.2 Subset Selection and Annotation

3.2.1 Topic Selection and Hypothesis Design

As previously discussed, the TREC 2009 Legal Track Interactive Task introduced a range of topics, numbered 201 to 207, each representing a distinct thematic focus. From these topics, three were selected for this thesis based on their relevance to forensic communication analysis and their emphasis on substantive organizational behavior rather than informal or peripheral content [47].

The selected topics are as follows:

- Topic 201: All documents or communications that describe, discuss, refer to, report on, or relate to the Company's engagement in structured commodity transactions known as "prepay transactions."

- Topic 204: All documents or communications that describe, discuss, refer to, report on, or relate to any intentions, plans, efforts, or activities involving the alteration, destruction, retention, lack of retention, deletion, or shredding of documents or other evidence, whether in hard-copy or electronic form
- Topic 205: All documents or communications that describe, discuss, refer to, report on, or relate to energy schedules and bids, including but not limited to, estimates, forecasts, descriptions, characterizations, analyses, evaluations, projections, plans, and reports on the volume(s) or geographic location(s) of energy loads.

From each topic, one hypothesis was derived, reflecting the focus of its corresponding topic while simultaneously introducing a forensic dimension.

The hypotheses are as follows:

- H1 (from topic 201): Employees described prepay transactions as loans or financing in order to keep associated debt off the balance sheet or to improve the company's reported financial position.
- H2 (from topic 204): Employees communicated about altering, deleting, or selectively retaining documents in anticipation of regulatory scrutiny, audits, or legal investigations.
- H3 (from topic 205): Employees discussed adjusting, restricting, or reallocating energy schedules, bids, or load volumes in ways that could affect supply availability or market prices.

3.2.2 Email Selection

The seed set from the TREC 2009 Legal Track Interactive Task contains document IDs for 8,245 non-unique items, including both emails and attachments. These IDs referred to documents in the original corpus that had been inspected during the task. To extract these documents, the IDs in the seed set were matched against the document IDs in the original corpus.

Table 3.1 presents the distribution of documents in the seed set across existing topics, while Table 3.2 illustrates the distribution of documents across different levels of responsiveness.

Table 3.1: Distribution of documents across existing topics.

Topic	Number of documents
200	850
201	732
202	1463
203	1042
204	1229
205	1961
206	365
207	603

Table 3.2: Distribution of documents across responsiveness levels.

Responsiveness	Meaning	Number of documents
-2	Document not inspected	13
-1	Document not inspected	347
0	Document is not responsive	5923
1	Document is responsive	1962

Table 3.3 shows the process by which the initial 8,245 non-unique seed-set records were reduced to 2,123 email files. During this process, attachments were first removed to simplify the dataset and reduce the annotation burden. After that, the emails assessed as non-responsive, exhibiting contradictory responsiveness across topics, or associated with multiple topics were subsequently excluded.

Table 3.3: Document filtering steps and remaining emails.

Step	Description	Remaining Documents
Initial	Seed set from TREC 2009 Legal Track Interactive Task	8,245
Corpus Matching	Matching IDs in the seed set against the corpus; 1,248 unique IDs not found	6,894 (2,294 unique)
Attachment Removal	Retaining only email messages by removing attachment documents	3,260
Non-responsive Removal	Removing documents assessed as non-responsive to their respective topics	3,182
Undefined Topic Removal	Removing documents associated with undefined or unused topic identifiers (e.g., Topic 200)	2,435
Duplicate Removal	Removing duplicate documents with identical topic and responsiveness labels	2,340
Contradiction Removal	Removing documents with conflicting responsiveness assessments for the same topic	2,320
Multi-topic Removal	Removing documents associated with multiple topics	2,123

Table 3.4 presents the distribution of emails across topics, the number of documents assessed as responsive, and the subset of topics selected for hypothesis design. Topics 201, 204, and 205 were identified as suitable for further analysis, as they are sufficiently explicit and can be interpreted without requiring additional contextual expansion.

Table 3.4: Total and responsive emails per topic with derived hypotheses.

Topic	Total Emails	Responsive Emails	Derived Hypothesis
201	241	62	H1
202	282	156	–
203	243	25	–
204	386	42	H2
205	706	106	H3
206	87	4	–
207	178	63	–

In contrast, Topics 202 and 203 were excluded due to their reliance on external domain knowledge. Topic 202 refers to accounting standard definitions, which require specialized interpretation and are not self-contained. Similarly, Topic 203 addresses the company's financial forecasts and projections, introducing ambiguity and lacking clearly defined behavioral indicators [47].

Additionally, Topic 206 was excluded due to the limited number of responsive documents, rendering it unsuitable for meaningful analysis. Finally, Topic 207 was not considered, as it primarily concerns informal content, such as fantasy football discussions, which falls outside the scope of forensic communication analysis [47].

As shown in Table 3.5, the final dataset used in the remainder of this thesis was constructed to contain 300 emails, with 100 emails assigned to each hypothesis. For H1, all 62 responsive emails were included, along with 38 randomly selected non-responsive emails. For H2, all 42 responsive emails were included, together with 58 randomly selected non-responsive emails. For H3, 20 responsive emails were randomly sampled from the responsive emails, together with 80 randomly selected non-responsive emails.

Table 3.5: Composition of the final hypothesis-specific email sets.

Hypothesis	Origin topic	Responsive emails	Non responsive emails	Total emails
H1	201	62	38	100
H2	204	42	58	100
H3	205	20	80	100

It is important to note that the final dataset contains emails whose responsiveness was assessed with respect to the original topics, not the hypotheses. Meanwhile, because the hypotheses were derived from these topics, emails that were non-responsive to a topic were also considered non-responsive to the corresponding hypothesis. This reduced the annotation effort, since only the remaining topic-responsive emails needed to be annotated against the hypotheses. In total, this meant annotating approximately 124 of the 300 emails.

3.2.3 Annotation Criteria

Hypothesis Structure

The hypotheses used in this thesis consist of two analytical components: *CAUSE* and *EFFECT*. The *CAUSE* refers to the behavior, communication, or intention expressed by employees, including what they are doing, discussing, or planning to do. The *EFFECT* refers to the expected, intended, or avoided consequence of that action, including what employees aim to achieve, anticipate, or prevent. For example, in the hypothesis "*Employees described prepay transactions as loans or financing in order to keep associated debt off the balance sheet or to improve the company's reported financial position*", action and outcome are as follows:

- CAUSE: "*Employees described prepay transactions as loans or financing*"
- EFFECT: "*keeping associated debt off the balance sheet or improving the company's reported financial position*"

Label Definitions

Each email was assigned one of three labels according to its relevance to the corresponding hypothesis.

An email was labeled *RELEVANT* when the cause was clearly present and the effect was either explicitly stated or strongly implied. This label was used when the email directly addressed the hypothesis and the relationship between the email content and the hypothesis required little or no interpretation.

An email was labeled *PARTIALLY RELEVANT* when the cause was present, but the effect was unclear, weakly implied, ambiguous, or absent. This label was also used when the content related to the hypothesis indirectly or required some degree of interpretation.

An email was labeled *NOT RELEVANT* when there was no clear evidence of either the cause or the effect. Emails were also considered not relevant when they had no meaningful connection to the hypothesis or when their content was administrative, personal, or otherwise unrelated.

Decision Rules

Annotators annotated each email only in relation to its corresponding hypothesis and not in relation to other hypotheses. Therefore, if multiple actions or outcomes appeared in a single email, only those actions and outcomes relevant to the corresponding hypothesis were considered.

Additionally, each email was annotated solely on the basis of its body content. No external knowledge, contextual assumptions, or information beyond the email text was used. In cases of uncertainty between two possible labels, annotators were instructed to choose the more conservative label. Specifically, when uncertain between relevant and partially relevant, they selected partially relevant; when uncertain between partially relevant and not relevant, they selected not relevant.

Annotation Workflow

Each email was independently annotated by two annotators. Annotators worked separately and were not permitted to consult with one another before or during the labeling process. When both annotators assigned the same label, that label was accepted as the final annotation. In cases of disagreement, a third annotator reviewed the email, and the final label was determined by majority vote. If all three annotators assigned 3 different labels, a fourth annotator was involved, and their decision was used as the final label.

Annotation results

Table 3.6 shows the results of annotation:

Table 3.6: Annotation results.

Hypothesis	# Emails to be annotated	# Relevant	# Partially Relevant	# Not Relevant
H1	62	2	16	44
H2	42	7	8	27
H3	20	14	6	0

Considering the non-responsive emails that are already not relevant to their corresponding hypotheses, Table 3.7 shows the final labels for the whole dataset:

Table 3.7: Labels of all emails.

Hypothesis	# Total emails	# Relevant	# Partially Relevant	# Not Relevant
H1	100	2	16	82
H2	100	7	8	85
H3	100	14	6	80

However, this multi-label setting is used only for annotation purposes. The primary evaluation setting throughout this thesis is the binary setting, in which the *RELEVANT* and *PARTIALLY RELEVANT* labels are merged into a single *USEFUL* class, while the *NOT RELEVANT* label is renamed *NOT USEFUL* for consistency. Table 3.8 presents this binary labeling scheme.

Table 3.8: Useful vs Not-useful setting.

Hypothesis	# Total emails	# Useful	# Not Useful
H1	100	18	82
H2	100	15	85
H3	100	20	80

Chapter 4

System Architecture

This chapter describes the architecture and implementation of the proposed hypothesis-driven forensic email analysis system. The system is designed as a multi-stage pipeline that prepares raw email data, converts investigative hypotheses into retrieval queries, retrieves potentially useful email contexts, and applies a language model to generate structured inference outputs. The chapter first introduces the practical use case and privacy considerations that motivate the system design. It then explains the main processing stages, including email cleaning, anonymization, semantic chunking, chunk refinement, normalization, embedding, vector database storage, query expansion, retrieval, context aggregation, context enrichment, and inference generation.

4.1 Practical Context and System Design

4.1.1 Use Case

To illustrate the practical application of the proposed system, consider a forensic investigation scenario involving a large corporate email dataset. An investigator is tasked with evaluating a hypothesis related to potential misconduct, such as the existence of undisclosed financial agreements or collusion between employees. For example, the investigator may formulate the following hypothesis:

"Employees were involved in unauthorized financial arrangements that were not disclosed through official communication channels."

In a traditional workflow, the investigator would rely on keyword-based

searches using terms such as *"payment"*, *"contract"*, or *"agreement"*, or manually inspect a large number of emails. However, relevant evidence may be expressed indirectly, using informal language or context-dependent phrasing, making it difficult to retrieve using simple keyword queries.

Using the proposed system, the investigator can instead input the hypothesis directly in natural language. The system first retrieves a set of emails that are potentially relevant based on semantic similarity and retrieval strategies. These emails are then processed by a language model, which generates explanations describing how each message relates to the hypothesis. For instance, the system may retrieve an email containing a sentence such as:

"Let's finalize the arrangement offline and avoid putting the details in the official report."

Although this email does not explicitly mention financial misconduct, the generated explanation may indicate that the message suggests an attempt to conceal information, making it relevant to the hypothesis. Additionally, the system highlights this sentence as an evidence span, allowing the investigator to quickly verify the basis of the result.

By combining retrieval, reasoning, and evidence extraction, the system provides a structured and interpretable output that supports the investigative process. This approach enables investigators to efficiently identify relevant communications while maintaining transparency and verifiability in their analysis.

4.1.2 Privacy and Practical Considerations

An important consideration in the design of the proposed system is the handling of potentially sensitive or confidential data. In forensic applications, email corpora may contain private, legally protected, or evidentiary information, requiring careful control over data processing and storage. The use of cloud-based or API-accessed large language models introduces potential risks related to data exposure, confidentiality, and compliance, as input data may be transmitted to external systems outside the control of the investigator [48].

In contrast, locally deployed or open-source language models provide greater control over data processing and reduce the risk of unintended data leakage. However, these models also introduce practical challenges related to computational resources and performance. In particular, such models may

exhibit increased latency in resource-constrained environments, which can negatively affect their usability.

The Ollama server [49] at Hochschule Mittweida provides several large language models suitable for the tasks addressed in this thesis. Table 4.1 lists 4 of the available models.

Table 4.1: List of 4 Available LLMs at Ollama Server of Hochschule Mittweida

Rank	LLM
1	Mixtral 8x22B
2	Gemma 4 31B
3	Ministral 3 14B
4	Qwen 3.5 9.7B Q4_K_M

To address privacy concerns, this thesis primarily relies on locally deployed open-weight language models, allowing inference to be performed without transmitting data to external services. The trade-off between inference latency and model performance is managed by primarily using the locally executed Gemma-4-31B model [50], while selectively employing the APIs of Gemma-4-31B-IT model [51] when improved response quality or faster inference is required.

4.1.3 Overview of the Proposed Pipeline

Figure 4.1 provides an overview of the proposed hypothesis-driven forensic email analysis pipeline. The system takes a collection of email threads as input, which are subsequently cleaned, anonymized, chunked, refined, normalized, embedded, and stored in a vector database. The input hypothesis is then expanded into k sparse and dense queries, which are used by the BM25, dense, and hybrid retrieval methods to retrieve the top m chunks.

The retrieved chunks may be enriched with neighboring chunks before being aggregated at the email level. Each resulting email-level context is then passed to the generation and inference component, which produces a usefulness label, quoted evidence spans, and an explanation of the relationship between the email and the hypothesis. The following sections describe each stage of the pipeline in greater detail.

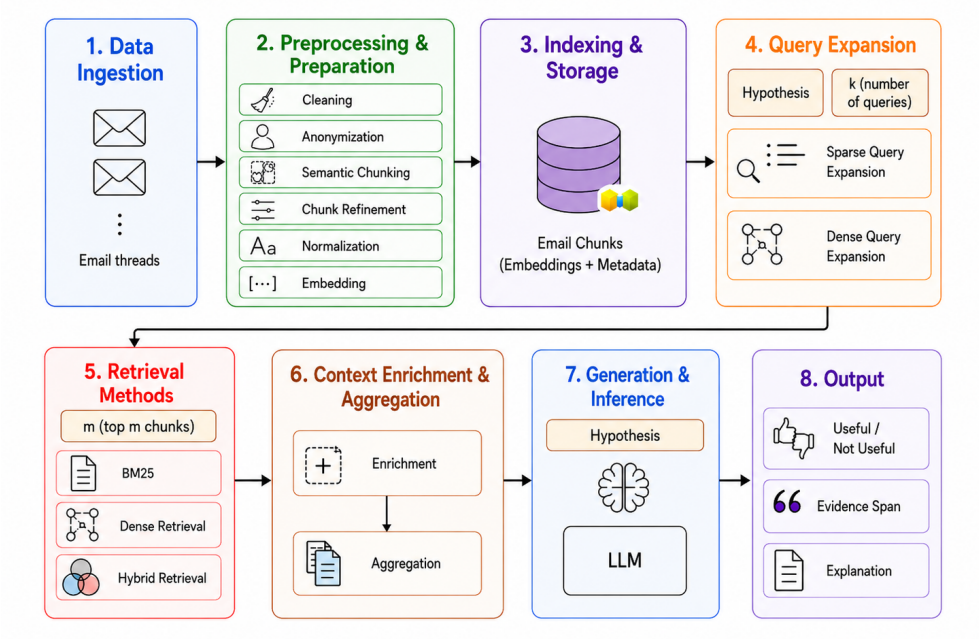

Figure 4.1: Overview of the proposed pipeline.

4.2 Preprocessing and Preparation

4.2.1 Data Cleaning

The email documents in their original form consist of full email threads that include substantial amounts of metadata and auxiliary content that are not relevant to the objectives of this thesis, which focuses on the analysis of email body content rather than metadata. Consequently, it is essential to remove such elements, including metadata fields, disclaimers, and signature blocks, as they introduce noise and can be negatively influential.

An initial approach involved manually inspecting a sample of emails to identify common patterns of noise, followed by the application of regular expression–based methods for their removal. While this approach proved effective for sparse retrieval methods, it was ultimately insufficient due to the high variability and inconsistency of the noise across documents. This variability significantly degraded the performance of dense retrieval models, necessitating a shift away from rule-based filtering toward a more robust preprocessing strategy.

LLM-based Data Cleaning

The process of data cleaning in this thesis was implemented using prompt-based interactions with large language models in two sequential phases. The first phase focused on noise removal and preparing the emails for both annotation and sparse retrieval. The second phase further refined the output of the first phase by anonymizing the content and masking personally identifiable information (PII). This allowed the embeddings to focus more directly on the core concepts in the emails, ultimately making the content more suitable for dense retrieval methods. The prompts used in both phases are provided in the Appendix A.

The primary model used for email cleaning in both phases was the locally deployed Gemma-4-31B model, served through an Ollama instance hosted at Hochschule Mittweida. However, due to its relatively high inference latency, some emails could not be processed within the predefined timeout limit and consequently resulted in timeout errors. In such cases, the API-hosted Gemma-4-31B-IT model was used as a fallback to complete the processing.

In the second phase, the outputs of the first phase were anonymized according to the schema presented in Table 4.2.

Table 4.2: Anonymization labels used for different entity types.

Entity	Anonymization
People Names	[PERSON]
Phone/Fax Numbers	[PHONE]
Email Addresses	[URL]
Passcodes/Access Codes	[PASSCODE]
Dates/Time Expressions	[DATE]
Monetary Values	[AMOUNT]
Percentage Values	[PERCENTAGE]

4.2.2 Semantic Chunking and Refinement

After the email cleaning stage, each document consisted of an email thread in which individual message blocks were separated using the marker <-- *block* -->. A synchronized semantic chunking [52] strategy was then applied to both the sparse and dense versions of the emails. The chunking process first split each document by the message-level block marker and then by paragraph

boundaries, using the ordered delimiters <-- *block* --> and \n\n.

This procedure was applied consistently across both representations. During the cleaning process, the semantic structure of the emails was preserved between the sparse and dense versions; only predefined entities were replaced in the dense version according to the anonymization schema presented in Table 4.2. As a result, both representations retained aligned semantic chunks.

The resulting chunk-level dataset contained the following fields: *sparse chunk*, *dense chunk*, *chunk index*, and *sparse chunk length*. The *chunk index* served as a shared identifier that linked each sparse chunk to its corresponding dense chunk.

Before finalizing the chunk-level dataset, additional filtering and restructuring steps were applied to manage the length of the chunks. Very short or uninformative chunks were either removed or merged with adjacent chunks when they did not provide enough context on their own. Conversely, overly long chunks were divided into smaller segments to keep the retrieval units manageable and focused.

All of the refinement operations described above were carried out in a synchronized manner across the sparse and dense chunk versions. The sparse chunk and its length were used as the reference version. Therefore, whenever a sparse chunk was removed, merged with an adjacent chunk, or split into smaller parts, the same operation was applied to its corresponding dense chunk using the shared chunk index. This preserved the one-to-one alignment between the sparse and dense datasets and ensured that both retrieval pipelines operated on equivalent units of text.

Removal of Short Blocks and Short Paragraphs

Extremely small email blocks were removed. As previously described, each document consists of an email thread composed of individual message blocks. Blocks containing only a single chunk with a length below 60 characters were discarded, as they typically contained minimal or non-informative content.

Very short paragraphs were also removed following paragraph-level splitting based on the delimiter \n\n. Paragraphs with a length below 30 characters were excluded, as they predominantly corresponded to non-substantive elements such as closing phrases, including “Best”, “Regards”, and “Thank you”.

Merge of Adjacent Short Neighboring Paragraphs

Short adjacent paragraphs were merged to improve coherence. Paragraphs with a length below 100 characters were combined with neighboring segments that were also below 100 characters, to form more meaningful textual units.

Breaking long chunks

This operation involved segmenting overly long chunks while avoiding the creation of excessively short sub-chunks. Long chunks can reduce retrieval precision because they may contain multiple ideas within a single retrieval unit. In such cases, a relevant passage may be embedded within a large amount of unrelated surrounding text, making it more difficult for the retrieval model to identify and rank the most relevant evidence. Although paragraph-based splitting was already applied to separate distinct ideas and reduce the likelihood of this problem, paragraphs of an email can still be unusually long and may contain several discussion points. Therefore, an additional long-chunk breaking operation was introduced to further improve the focus of the retrieval units. For this purpose, an upper limit of 600 characters was defined, and chunks exceeding this limit were selected for segmentation into smaller sub-chunks.

At the same time, this operation involved a trade-off, since splitting long chunks may separate related sentences or weaken contextual links within an email. To reduce this risk, a minimum length of 200 characters was used to avoid creating sub-chunks that were too short to provide sufficient context. Additionally, segmentation was performed at sentence boundary delimiters such as “.”, “!”, and “?”, and an overlap of one sentence was introduced between consecutive sub-chunks. This overlap was defined not in terms of characters, but as the number of shared sentences between consecutive sub-chunks.

Each chunk exceeding the upper limit was first decomposed into a sequence of sentences using boundary delimiters. These sentences served as the fundamental units for subsequent sub-chunk construction and were iteratively appended until the desired length range was reached. Overlap was also introduced during the construction of each subsequent sub-chunk to preserve contextual continuity between consecutive sub-chunks.

One consideration of this process is that all resulting sub-chunks should ideally exceed the minimum length threshold. While this condition is typically satisfied for earlier sub-chunks, the final segment may fall below the minimum

length. To address this, an exception is applied: if the last sub-chunk is shorter than the minimum length, additional preceding sentences are included beyond the standard overlap in order to increase its length and bring it within the acceptable range.

Another consideration arose from the use of overlap, which could in some cases produce sub-chunks that exceeded the defined upper limit. In this study, the upper limit was used primarily to identify chunk candidates for segmentation, rather than as a strict maximum boundary for the length of the resulting sub-chunks. This was also acceptable because the context window of the LLM used in this study for generation module was larger than the specified maximum chunk length. As a result, such sub-chunks were retained without further segmentation. Observations, presented in later figures, indicated that these cases occurred infrequently.

Table 4.3 shows a summary of parameters used in this section:

Table 4.3: Parameters used during chunk refinement operations.

Parameter	Value	Description
Block threshold	60 chars	Email blocks below this range were dropped
Paragraph threshold	30 chars	Paragraphs below this range were dropped
Merge threshold	100 chars	Adjacent neighboring paragraphs below this range were merged
Upper limit	600 chars	Chunks above this range were broken into sub-chunks
Minimum length	200 chars	Sub-chunks should not be below this range
Overlap	1 sentence	Number of sentences used as overlap between sub-chunks

The following figures and tables illustrate how these refinement operations affected the chunk-length distribution and related statistics for hypotheses H1, H2, and H3.

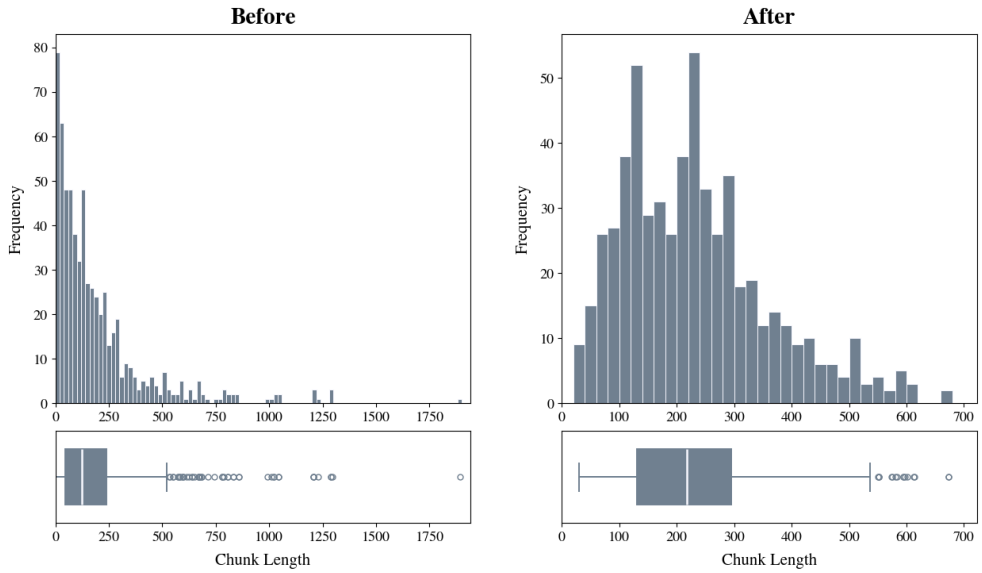

Figure 4.2: Chunk-length distribution before and after refinement for H1.

Table 4.4: Chunk statistics before and after refinement for H1.

	Before Refinement	After Refinement
Number of chunks	636.00	578.00
Minimum chunk length	3.00	30.00
Maximum chunk length	1892.00	673.00
Mean chunk length	191.75	230.79
Median chunk length	123.00	217.00
Standard deviation	230.01	128.28
Chunks below 30 characters	109.00	0.00
Blocks below 60 characters	9.00	0.00
Chunks above 600 characters	38.00	5.00

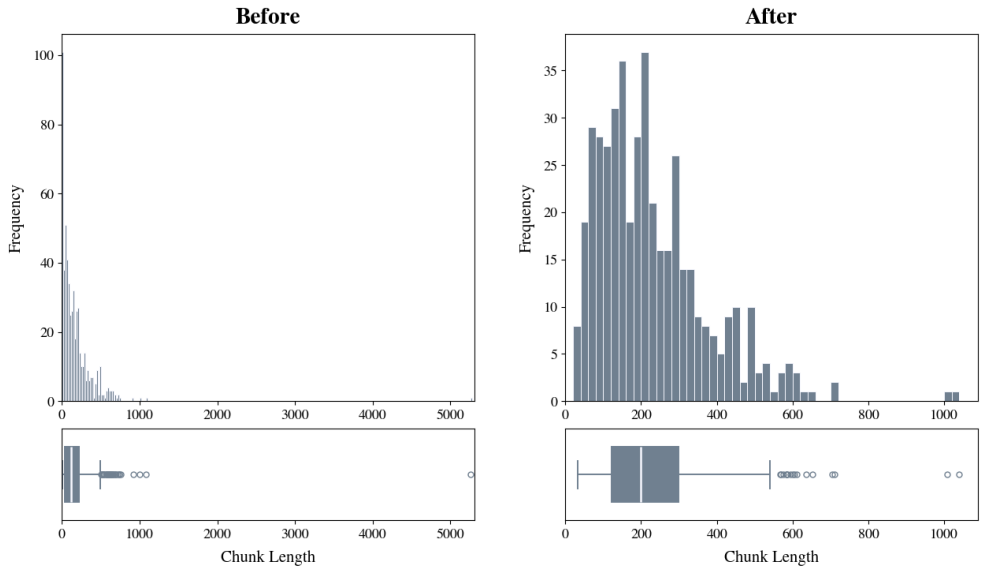

Figure 4.3: Chunk-length distribution before and after refinement for H2.

Table 4.5: Chunk statistics before and after refinement for H2.

	Before Refinement	After Refinement
Number of chunks	561.00	453.00
Minimum chunk length	4.00	32.00
Maximum chunk length	5263.00	1039.00
Mean chunk length	174.23	229.44
Median chunk length	113.00	200.00
Standard deviation	275.84	148.18
Chunks below 30 characters	123.00	0.00
Blocks below 60 characters	17.00	0.00
Chunks above 600 characters	19.00	8.00

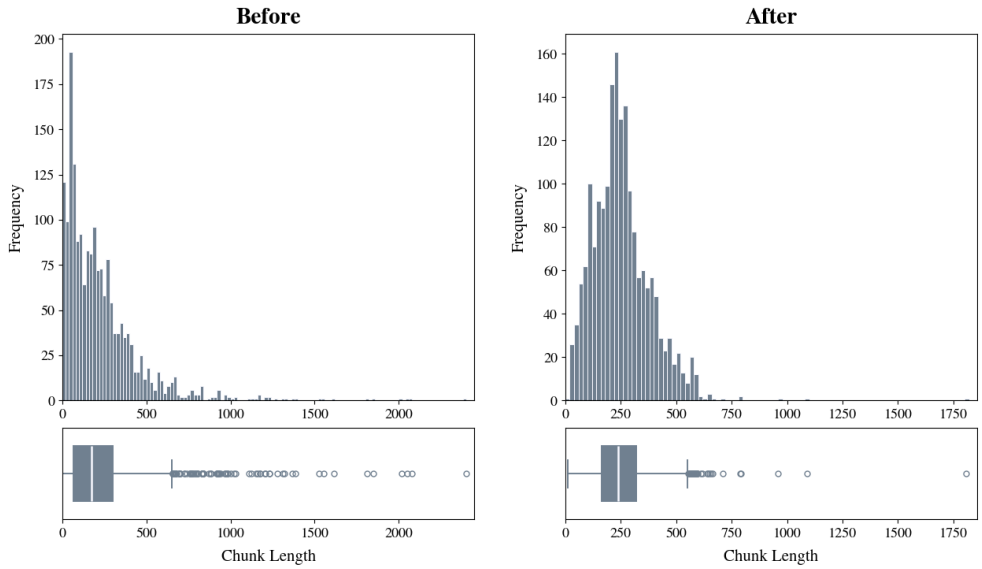

Figure 4.4: Chunk-length distribution before and after refinement for H3.

Table 4.6: Chunk statistics before and after refinement for H3.

	Before Refinement	After Refinement
Number of chunks	1841.00	1837.00
Minimum chunk length	3.00	13.00
Maximum chunk length	2400.00	1808.00
Mean chunk length	227.74	253.49
Median chunk length	172.00	238.00
Standard deviation	237.03	132.15
Chunks below 30 characters	161.00	1.00
Blocks below 60 characters	10.00	0.00
Chunks above 600 characters	108.00	13.00

Overall, the refinement process changed the chunk-length distribution in a clear and measurable way. The total number of chunks decreased, while both the mean and median chunk lengths increased, showing that the remaining chunks are generally longer than before refinement. At the same time, the standard deviation decreased, indicating that chunk lengths became less dispersed. The minimum chunk length increased, and chunks or blocks below the short-length threshold were eliminated almost entirely, with one short

chunk remaining for H3. The maximum chunk length also decreased substantially, and the number of chunks above the upper-length threshold was reduced, showing that extreme long chunks were limited. Together, these results indicate that the refinement process produced a more balanced chunk-length distribution.

4.2.3 Normalization and Embedding

After obtaining the sparse and dense chunks through semantic chunking, the next step was to prepare both chunk types for retrieval. Because sparse and dense retrieval methods process text differently, each chunk type was handled using a separate preprocessing pipeline.

For the sparse chunks, a normalization process was applied. Each sparse chunk was first converted to lowercase. Then, all URLs were replaced with a single space. Next, all characters except lowercase letters, digits, the @ symbol, periods, plus signs, and hyphens were removed and replaced with spaces. Finally, consecutive whitespace characters were collapsed into a single space. This process produced a normalized textual representation of each sparse chunk.

Dense retrieval uses semantic representations of text. Therefore, the semantic content of the dense chunks was preserved, and the dense chunks were not subjected to the same normalization steps used for sparse chunks. However, the dense chunks were passed to an embedding model to obtain vector representations. The embedding model used in this stage was *BAAI/bge-small-en*, with embedding normalization enabled.

4.2.4 Vector Database

After generating the final sparse chunks, final dense chunks, and dense chunk embeddings, the next step was to store them in a vector database so that they could be efficiently retrieved during the search process. For this purpose, *Weaviate* was used as the vector database.

Two separate collections were created in the vector database. The first collection was designed to store information at the email level. This collection contained the email ID, the cleaned version of the email content, and the raw email content. This collection preserved the full email-level context and

allowed the system to return or reference the original email information after retrieval.

The second collection was designed to store information at the chunk level. The fields stored in this collection included the email ID, chunk ID, chunk index, sparse chunk, dense chunk, and the embedding of the dense chunk. The chunk ID served as a counter for each chunk within each email. The chunk index served as a unique identifier for the chunks and was constructed using both email ID and chunk ID.

The dense chunk embedding was stored together with the chunk object so that semantic similarity search could be performed directly in Weaviate. At the same time, the sparse chunk was also stored to support keyword-based retrieval. This structure allowed the database to maintain both semantic and lexical representations of each chunk.

By separating the data into these two collections, the system could retrieve the most relevant chunks while still preserving the connection to the complete email from which each chunk was generated.

4.2.5 Query Expansion

Before querying the database, it is necessary to transform the hypotheses into suitable search queries. As mentioned before, the hypotheses are expressed in natural language and can be complex or contain multiple parts and different aspects. Therefore, they cannot be used directly as search queries. Additionally, queries used for sparse retrieval and dense retrieval are expressed in different forms. While sparse retrieval benefits from keyword-style queries, dense retrieval performs much better with descriptive narrative queries.

Figure 4.5 shows the pipeline used to generate a list of k queries that were suitable for the target retrieval method.

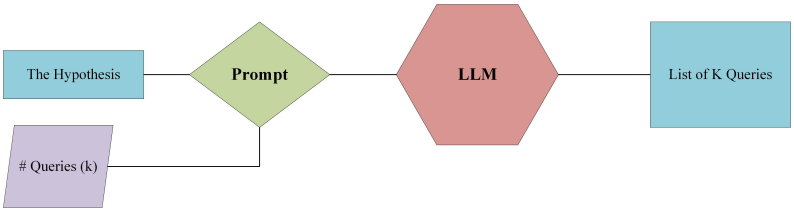

Figure 4.5: Pipeline for expanding a hypothesis into k queries.

While the pipeline architecture is the same for both retrieval methods,

the corresponding prompts are different. For sparse retrieval, the prompt instructed the model to generate short keyword-oriented queries. These queries were designed to match terms that could appear directly in the texts. Therefore, the sparse queries focused on important words, phrases, and variations that were likely to occur in the email text. For dense retrieval, the prompt instructed the model to generate more semantic and natural-language queries. These queries were designed to capture the meaning of the hypothesis rather than only exact keywords. As a result, the dense queries were usually more descriptive and sentence-like, allowing the search for conceptually related chunks. Both prompts can be found in Appendix A.

The output of this stage was two query lists for each hypothesis: one list of k sparse queries and one list of k dense queries. These expanded queries were then used in the retrieval stage to search through the chunks stored in the vector database.

4.3 Retrieval

After the documents had been chunked, embedded, and stored in the vector database, three retrieval pipelines were implemented: BM25 retrieval, dense retrieval, and hybrid retrieval. All three pipelines operate on the same preprocessed emails, but they differ in the way relevant chunks are identified, scored, and ranked for a given query.

A common parameter across all three retrieval methods is the number of desired unique chunks to be retrieved, denoted by m. This parameter defines the final number of distinct chunks returned by each pipeline. In the following subsections, each retrieval method is described in terms of how, given a hypothesis and k generated queries, it retrieves m distinct chunks.

4.3.1 Retrieval Pipelines

BM25

The BM25 retrieval pipeline receives k sparse expanded queries as input and aims to retrieve m distinct chunks from the vector database. Since multiple queries may retrieve overlapping chunks, the pipeline retrieves more candidates than the final required number of chunks. This increases the likelihood that, after duplicate chunks are removed, at least m unique chunks remain.

For each query, the number of initially retrieved chunks is defined in Equation 4.1:

$$n = 2 \times \frac{m}{k} \tag{4.1}$$

where *m* denotes the desired number of final distinct chunks, and *k* denotes the number of expanded queries. For example, if *m*=50 and *k*=10, then *n*=10. In this case, each of the 10 queries retrieves 10 candidate chunks, resulting in a maximum of 100 retrieved candidates before deduplication.

Each query is submitted independently to the BM25 retriever, which returns a ranked list of *n* chunks with their corresponding BM25 scores. However, BM25 scores produced for different queries are not directly comparable, since they depend on the lexical overlap between a specific query and the retrieved chunks. Therefore, the scores are normalized separately within the result list of each query.

For a given query q_i, the score of each retrieved chunk c_j is normalized as shown in Equation 4.2, by dividing it by the maximum BM25 score obtained for that query:

$$s'_{i,j} = \frac{s_{i,j}}{\max(s_i)} \tag{4.2}$$

where $s_(i,j)$ is the original BM25 score of chunk c_j for query q_i, and max(s_i) is the highest BM25 score among the *n* chunks retrieved for that query. This normalization maps the highest ranked chunk for each query to a score of 1, while the remaining chunks receive relative scores between 0 and 1.

After normalization, the retrieved chunks from all *k* queries are merged into a single candidate set. Since the same chunk may be retrieved by multiple queries, duplicate chunks are removed. If a chunk appears in more than one query result list, only one instance of the chunk is retained. Its final score is defined in Equation 4.3, as the maximum normalized score assigned to it across all queries:

$$S(c_j) = \max_i s'_{i,j} \tag{4.3}$$

This strategy preserves the strongest evidence of relevance for each chunk while avoiding repeated occurrences in the final retrieval output. The merged

chunks are then ranked according to their final scores $S(c_j)$, and the top m distinct chunks are selected as the output of the BM25 retrieval pipeline.

Dense

The dense retrieval pipeline follows the same overall structure as the BM25 retrieval pipeline, but it uses semantic similarity in the embedding space instead of lexical matching. The input to this pipeline consists of k expanded queries that are specifically generated for dense retrieval.

As in the BM25 pipeline, the goal is to retrieve m final distinct chunks. To reduce the effect of duplicate chunks appearing across the results of multiple queries, the pipeline retrieves more candidates than the final required number. The formula of the number of initially retrieved chunks per query is the same as BM25 shown in Equation 4.1.

Before retrieval, each dense query is embedded using the same embedding model that was used to embed the document chunks. Since the stored chunk embeddings were normalized during the embedding stage, the query embeddings are also normalized to ensure consistency in the vector space. Each normalized query embedding is then used to search the vector database and retrieve the top n most similar chunks.

Unlike the BM25 pipeline, no additional score normalization is applied to the retrieved results. This is because all dense retrieval scores are computed within the same embedding space using the same embedding model and normalized representations. As a result, the similarity scores produced for different queries are comparable.

After retrieving the top n chunks for each of the k queries, the results are merged into a single candidate set. Since the same chunk may be retrieved by more than one query, duplicate chunks are removed. If a chunk appears in multiple query result lists, only one instance is retained. Its final score is defined in Equation 4.4, as the maximum similarity score assigned to it across all queries:

$$S(c_j) = \max_i s_{i,j} \tag{4.4}$$

where $s_(i, j)$ denotes the dense retrieval similarity score of chunk c_j for query q_i. This aggregation strategy keeps the strongest semantic relevance signal for each chunk while ensuring that each retrieved chunk appears only

once in the final output.

Finally, all unique chunks are ranked according to their final scores $S(c_j)$, and the top *m* distinct chunks are selected as the output of the dense retrieval pipeline.

Hybrid

The hybrid retrieval pipeline combines the outputs of the BM25 and dense retrieval pipelines in order to benefit from both lexical and semantic retrieval signals. In this approach, the BM25 pipeline is first applied using the sparse expanded queries, while the dense retrieval pipeline is applied using the dense expanded queries. As described in the previous sections, both pipelines aim to return *m* distinct chunks.

After both retrieval pipelines have been executed, their outputs are combined into a single candidate set. At this stage, a chunk may appear in only the BM25 results, only the dense retrieval results, or in both. To merge and re-rank these results, Reciprocal Rank Fusion (RRF) is applied. RRF is used because it combines ranked lists based on the position of each item rather than relying directly on the original retrieval scores. This is important in the hybrid setting, since BM25 and dense retrieval scores are produced by different scoring mechanisms and are not directly comparable.

For each chunk c_j, the RRF score is computed based on its rank in the BM25 and dense retrieval result lists, as shown in Equation 4.5:

$$\text{RRF}(c_j) = \sum_{r \in R} \frac{1}{\lambda + \text{rank}_r(c_j)} \tag{4.5}$$

where *R* represents the set of retrieval result lists, $rank_r(c_j)$ is the rank of chunk c_j in result list *r*, and λ is a constant used to reduce the impact of very high ranks. If a chunk does not appear in one of the retrieval lists, it receives no contribution from that list. Therefore, chunks that appear in both BM25 and dense retrieval results receive contributions from both rankings, while chunks appearing in only one result list are scored based only on that list.

After calculating the RRF scores, all candidate chunks are ranked according to their fused scores. Duplicate chunks are removed during this process, so each chunk appears only once in the final output. The top *m* chunks according to the RRF ranking are then selected as the final result of the hybrid retrieval pipeline.

4.3.2 Chunk Aggregation

After the top *m* distinct chunks are retrieved, they are grouped according to their *Email ID* and then sorted by their *Chunk ID*. For example, assume that chunks with IDs 5 and 2 are retrieved among the top *m* results for email ID 50. These chunks are first grouped under the same email and then ordered according to their original chunk sequence. As a result, email ID 50 is represented by a combined context containing chunks 2 and 5, respectively. This reconstructed email-level context is then passed to the generation module.

Context Enrichment

In addition to aggregating retrieved chunks at the email level, an enrichment strategy can also be applied to expand the available context. Continuing the previous example, suppose that chunks 2 and 5 are retrieved among the top *m* results for email ID 50. The database can then be queried again for each retrieved chunk using a predefined *window* parameter. For example, with a window size of 1, the immediate neighboring chunks are also retrieved: chunks 1 and 3 for chunk 2, and chunks 4 and 6 for chunk 5.

After this enrichment step, all retrieved and neighboring chunks are grouped by their *Email ID*, sorted according to their *Chunk ID*, and duplicate chunks are removed. The resulting ordered set of chunks is then used as the final email-level context passed to the generation module.

Figure 4.6 illustrates the email-level context construction for this example email under two different enrichment settings: absence of enrichment (a window size of 0) and a window size of 1.

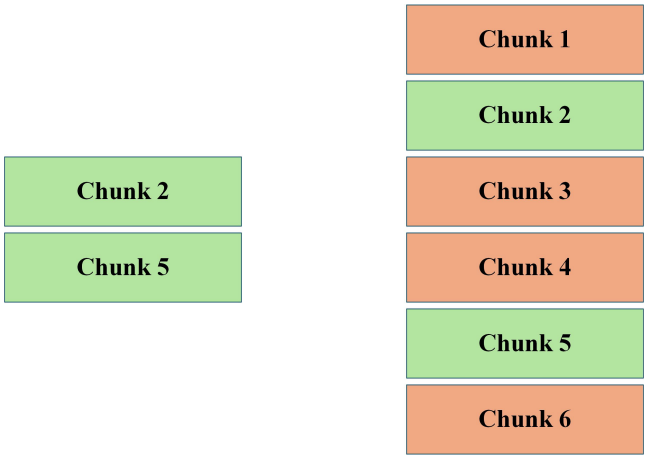

Figure 4.6: Context before and after applying enrichment with window size of 1

4.4 Generation and Inference

After the retrieval stage, the selected email-level contexts, together with the original hypothesis, are passed individually to the generation stage. This stage is applied to contexts produced by one of the retrieval pipelines: BM25, dense retrieval, or hybrid retrieval. The objective is to assess the relevance of each context to the given hypothesis and to provide both evidence and an explanation supporting the assigned relevance degree.

As shown in Figure 4.7, for each email-level context, the LLM receives two main inputs: the original hypothesis and the context.

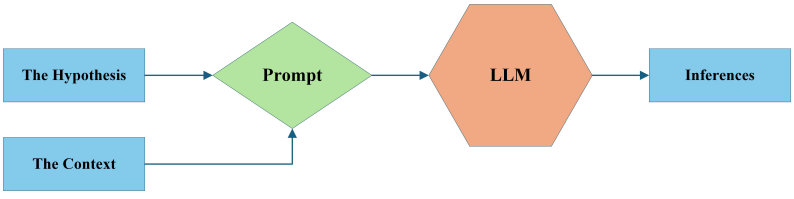

Figure 4.7: Pipeline for producing the inferences.

The prompt instructs the LLM to evaluate the relationship between the email-level context and the hypothesis and to produce a structured inference output. This output consists of a *Relevance Label*, an *Evidence Span* when applicable, and a *Reasoning* field. The inference output generated for each context is then treated as the system's prediction for the corresponding email.

4.4.1 Relevance Label

For each email-level context, the LLM first assigns one of three relevance labels: *Relevant*, *Partially Relevant*, or *Not Relevant*. A context is labeled as *Relevant* if it directly and sufficiently supports the hypothesis. It is labeled as *Partially Relevant* if it contains information related to the hypothesis, but does not fully support it or only addresses one aspect of it. Finally, it is labeled as *Not Relevant* if it does not provide a meaningful connection to the hypothesis.

In a subsequent step, the *Relevant* and *Partially Relevant* labels are merged into a single *Useful* class, while *Not Relevant* is renamed *Not Useful* for consistency. This binary formulation represents the main evaluation setting of this thesis, as it is more aligned with the forensic objective of identifying emails that may provide useful evidence for a given hypothesis.

4.4.2 Evidence Span

If an email-level context is labeled as *Relevant* or *Partially Relevant*, and therefore as *Useful* in the binary setting, the LLM is additionally required to extract a list of evidence candidates. These evidence candidates consist of exact quoted spans from the retrieved context. The purpose of this step is to ensure that the model's prediction is grounded in explicit textual evidence, rather than being based only on a general interpretation of the context.

4.4.3 Reasoning

In addition to the relevance label and evidence candidates, the LLM also produces a reasoning field. For contexts labeled as *Useful*, this field explains how the extracted evidence spans connect the email-level context to the hypothesis. For contexts labeled as *Not Useful*, it explains why the context does not provide sufficient support for, or a meaningful connection to, the hypothesis.

Chapter 5

Evaluation, Results, and Discussion

This chapter presents the evaluation of the proposed hypothesis-driven retrieval and inference pipeline. It compares BM25, dense retrieval, and hybrid retrieval across the three hypotheses introduced in Chapter 3. The evaluation considers both stages of the system: the retrieval stage, which identifies potentially useful emails, and the inference stage, which classifies retrieved emails as useful or not useful. In addition to standard precision, recall, and F1 scores, the chapter evaluates the quality of the generated reasoning and extracted evidence spans. It also examines the effect of context enrichment and analyzes the stability of the pipeline across multiple runs. The chapter concludes by discussing the main findings, the differences between hypotheses, and the implications of the results for forensic email analysis.

5.1 Random Baseline

5.1.1 Random Retrieval Setup

Before evaluating the proposed retrieval pipelines, a random baseline was constructed for each hypothesis. The purpose of this baseline is to provide a reference point for understanding how much improvement is obtained by using hypothesis-aware retrieval methods instead of randomly selecting content from the database.

For each hypothesis, 100 chunks were randomly sampled from the database. These chunks were then grouped by their corresponding email

identifiers in the same way as in the main pipeline. After grouping, an email-level context was constructed for each retrieved email. No contextual window enrichment was applied in this baseline setting. The resulting contexts were then passed to the inference module, which produced a binary useful/not-useful prediction for each email. The random baseline was repeated ten times for each hypothesis. The values reported in this section correspond to the average performance across these runs.

The choice of sampling 100 chunks was made to match the retrieval budget used in the main experiments. Therefore, the random baseline should be interpreted as a budget-matched baseline: it uses the same number of initially retrieved chunks as BM25, dense retrieval, and hybrid retrieval, but without using the hypothesis content.

5.1.2 Baseline Results

This baseline evaluates two related aspects of the pipeline. First, the retrieval metrics measure the probability of retrieving useful emails when the retrieval stage has no access to the hypothesis content. Second, the inference-label metrics measure the end-to-end usefulness predictions produced after the randomly retrieved contexts are processed by the inference module. Since useful emails that are not retrieved cannot be predicted as useful, the inference-label results reflect the combined effect of random retrieval and downstream classification rather than the isolated performance of the inference module. Table 5.1 reports the random baseline results for both retrieval and inference.

Table 5.1: Random baseline results for retrieval and inference across the three hypotheses.

Hypothesis	Stage	Precision	Recall	F1
H1	Retrieval	0.3173 ± 0.0439	0.6000 ± 0.1105	0.4141 ± 0.0610
H1	Inference	0.7983 ± 0.1909	0.1944 ± 0.0600	0.3106 ± 0.0889
H2	Retrieval	0.1610 ± 0.0258	0.4467 ± 0.0632	0.2365 ± 0.0362
H2	Inference	0.6419 ± 0.2337	0.2400 ± 0.1052	0.3428 ± 0.1339
H3	Retrieval	0.2298 ± 0.0428	0.4700 ± 0.0856	0.3080 ± 0.0548
H3	Inference	0.8725 ± 0.1902	0.2100 ± 0.0775	0.3346 ± 0.1063

Table 5.2 summarizes the baseline behavior across the three hypotheses. The macro-average is computed by averaging the corresponding mean metric values across H1, H2, and H3.

Table 5.2: Macro-average random baseline results across the three hypotheses.

Stage	Macro Precision	Macro Recall	Macro F1
Retrieval	0.2360	0.5056	0.3195
Inference	0.7709	0.2148	0.3293

5.1.3 Baseline Interpretation

The relatively high recall of the random retrieval baseline should be interpreted in light of the large retrieval budget. Since 100 chunks are sampled before email-level grouping, the baseline has many opportunities to retrieve at least one chunk from a useful email. However, this comes at the cost of low precision. In other words, although the retrieved context may originate from a useful email, it often contains chunks that do not provide sufficient value for identifying the email as useful.

The inference results show a different pattern. Compared with the retrieval baseline, the inference stage generally achieves higher precision but lower recall. This suggests that the inference module acts as a filtering step: it rejects many randomly retrieved emails and assigns the useful label to a smaller subset. When an email is predicted as useful, the prediction is often correct, particularly for H3, where inference precision reaches 0.8725 on average. However, the low recall values indicate that many useful emails are absent from the final predicted-useful set. Some are not retrieved by the random selection process, while others are retrieved with contexts that lack sufficient evidence for the model to classify them as useful.

Overall, the random baseline provides an uninformed reference point for the remaining experiments. It shows what level of performance can be achieved when the retrieval stage does not use the content of the hypothesis. Therefore, the following experiments compare BM25, dense retrieval, and hybrid retrieval against this baseline to assess whether hypothesis-aware retrieval improves both the retrieval stage and the downstream inference results.

5.2 Evaluation Setup

5.2.1 Evaluation Units and Metrics

The evaluation is conducted over the three hypothesis-specific subsets introduced in Chapter 3. Each hypothesis contains 100 emails, resulting in a total evaluation set of 300 emails. The evaluated retrieval pipelines are BM25, dense retrieval, and hybrid retrieval. The outputs are evaluated at the email level, since the final objective is to identify emails that are useful for assessing a hypothesis.

Although the annotation scheme distinguishes between relevant, partially relevant, and not relevant emails, the main evaluation uses the binary useful/not-useful setting. Relevant and partially relevant emails are merged into the useful class, while not relevant emails are treated as not Useful.

Table 5.3 summarizes the evaluation framework used for the different stages of the pipeline. Retrieval and label prediction are evaluated using standard classification metrics, including precision, recall, and F1 score, with confusion matrices additionally used for label analysis. The quoted evidence spans are evaluated for supportiveness and faithfulness, whereas the explanation text is assessed in terms of relevance, correctness, and clarity.

Table 5.3: Evaluation components, evaluated outputs, and corresponding metrics.

Component	Evaluated output	Metrics
Retrieval	Retrieved emails	Precision, Recall, F1
Label	Useful / Not Useful	Precision, Recall, F1, Confusion matrix
Evidence span	Quoted spans	Supportiveness, Faithfulness
Reasoning	Explanation text	Relevance, Correctness, Clarity

For all precision, recall, and F1 calculations, the useful class is treated as the positive class. Therefore, precision measures the proportion of predicted useful emails that are truly useful, while recall measures the proportion of all useful emails that are successfully identified.

The useful/not-useful label evaluation is performed from an end-to-end pipeline perspective over all 100 emails associated with each hypothesis. The

inference module produces labels only for emails returned by the retrieval stage; however, emails that are not retrieved are implicitly treated as not useful in the final pipeline output. Consequently, a useful email is counted as a false negative either when it is not retrieved or when it is retrieved but classified as not useful. The reported inference-label metrics therefore reflect the combined effect of retrieval and downstream inference rather than the inference module in isolation.

5.2.2 Default and Enriched Context Settings

The results for each hypothesis are reported under two settings:

- Default context: without context enrichment.
- Enriched context: with context enrichment using a window size of 1.

In the default context setting, the evaluation is reported separately for the retrieval methods, including BM25, dense retrieval, and hybrid retrieval, as well as for the inference outputs, including predicted labels, evidence spans, and reasoning.

In the enriched context setting, the retrieval metrics remain unchanged, since the same retrieved emails are used before applying context enrichment. Therefore, these metrics are not repeated. Moreover, to avoid an excessive number of tables and reduce potential confusion, the evidence span and reasoning evaluations are not reported for this setting. Instead, the enriched context setting focuses only on the analysis of the predicted labels.

5.2.3 Experimental Configuration

The performance of the pipeline for each hypothesis is reported under the experimental setting shown in Table 5.4:

Table 5.4: Experimental configuration used for running the pipeline.

Parameter	Value
Query expansion parameter (k)	10
Query expansion model	Gemma-4-31B-IT
Embedding model	BAAI/bge-small-en
Top chunks to retrieve (m)	100
Inference model	Gemma-4-31B-IT

The generated reasoning and evidence spans were evaluated using ChatGPT with GPT-5.5 and the High reasoning setting. The evaluation was performed in batches rather than independently for each email. For each experimental output, the complete inference-results file and the corresponding ground-truth emails were provided to ChatGPT. The model was instructed to assess the generated reasoning according to the definitions of relevance, correctness, and clarity, and to assess the extracted evidence spans according to the definitions of supportiveness and faithfulness presented in *Section 2.5*. The resulting assessments were then aggregated for each retrieval method.

5.3 Results for H1

5.3.1 Default Context

Retrieval Performance

Table 5.5 reports email-level retrieval performance for H1, treating the Useful class as the positive class.

Table 5.5: Retrieval performance for H1.

Retrieval Method	Retrieved Emails	Precision	Recall	F1
BM25	39	0.4359	0.9444	0.5965
Dense	36	0.4444	0.8889	0.5926
Hybrid	43	0.3953	0.9444	0.5574

Compared with the random retrieval baseline for H1, all hypothesis-aware retrieval methods improve performance. BM25 increases F1 from 0.4141 to 0.5965, while dense retrieval reaches a similar F1 of 0.5926. This shows that the expanded hypothesis queries provide a substantially stronger retrieval signal than random chunk selection. BM25 achieves the highest recall together with hybrid retrieval. Dense retrieval has slightly higher precision, but retrieves fewer useful emails. The hybrid method does not improve over BM25 in this case, suggesting that H1 contains strong lexical cues which are already effectively captured by BM25.

Inference Label Performance

Table 5.6 reports the useful/not-useful label performance of the inference module for each retrieval method.

Table 5.6: Inference label performance for H1 with default context.

Retrieval Method	Precision	Recall	F1
BM25	0.8462	0.6111	0.7097
Dense	0.6667	0.6667	0.6667
Hybrid	0.6667	0.5556	0.6061

Table 5.7 reports the end-to-end useful/not-useful confusion matrices over all 100 emails in the H1 evaluation subset. The false negatives include both useful emails that were not retrieved and retrieved useful emails that were incorrectly labeled as not-useful.

Table 5.7: End-to-end useful/not-useful confusion matrices for H1.

Retrieval Method	Actual Class	Predicted Useful	Predicted Not Useful
BM25	Useful	11	7
	Not Useful	2	80
Dense	Useful	12	6
	Not Useful	6	76
Hybrid	Useful	10	8
	Not Useful	5	77

The inference stage increases precision compared with the retrieval stage for all three retrieval methods. This indicates that the language model is effective as a filtering component, assigning the Useful label to a smaller and more accurate subset of the retrieved emails. However, this improvement in precision comes with lower end-to-end recall. Useful emails may be absent from the final predicted-useful set either because they were not retrieved or because they were retrieved but classified as not useful by the inference module. This trade-off is particularly important in forensic settings, where high recall is often desirable in order to avoid overlooking potentially relevant evidence.

Reasoning and Evidence-Span Quality

Table 5.8 shows the three reasoning quality criteria: relevance, correctness, and clarity. Each criterion was assigned one of three scores: good = 1, partial = 0, and poor = -1.

Table 5.8: Reasoning quality assessment for H1 across retrieval methods.

Retrieval Method	Relevance	Correctness	Clarity
BM25	1.0000	0.9744	1.0000
Dense	1.0000	0.9444	1.0000
Hybrid	1.0000	0.9762	1.0000

Most generated explanations were judged to be relevant, correct, and clear. The relevance and clarity scores are 1.0000 across all retrieval methods, indicating that the explanations generally addressed the relationship between the email context and H1 in an understandable way.

The small reductions in correctness are caused by a limited number of partially correct explanations. These cases mainly occur when the explanation understates the financial meaning of the retrieved context. In particular, some explanations state that the context does not contain financing-related content, although the retrieved context includes expressions such as “financing SPV”, “accounting purposes”, or “swap structure”. These expressions are relevant to H1 because they connect prepay transactions to financing structures, accounting treatment, or balance-sheet implications. Therefore, such explanations were not marked as fully correct. However, they were not treated as completely incorrect, because the retrieved context did not always explicitly state the full hypothesis, such as keeping debt off the balance sheet or improving the reported financial position.

Overall, the reasoning-quality results suggest that the inference module provides reliable explanations for H1. The main limitation is not that the model produces irrelevant or unclear reasoning, but that it sometimes applies a stricter interpretation of financing-related evidence than the annotation criteria used for the hypothesis.

Table 5.9 shows the evidence-span quality according to two criteria: supportiveness and faithfulness. Each criterion was scored using a binary scale, where yes = 1 and no = 0.

Table 5.9: Evidence-span quality assessment for H1 across retrieval methods.

Retrieval Method	Total Spans	Supportiveness	Faithfulness
BM25	20	0.8000	1.0000
Dense	25	0.8400	1.0000
Hybrid	22	0.7273	1.0000

The evidence-span evaluation shows perfect faithfulness across all retrieval methods, meaning that the extracted spans were present in the retrieved contexts and were not fabricated. This is important for the forensic setting, because evidence spans are intended to provide verifiable support for the model's decision.

Supportiveness is lower than faithfulness for all three methods. BM25 has 16 supportive spans out of 20, dense retrieval has 21 supportive spans out of 25, and hybrid retrieval has 16 supportive spans out of 22. This indicates that the model usually extracts grounded spans, but not every grounded span provides sufficient independent evidence for H1.

The non-supportive spans are mainly cases where the quoted text refers to prepay transactions only in a general way. Such spans may be topically related to H1, but they do not independently support the hypothesis unless they also connect prepay transactions to financing, loans, swap structures, accounting treatment, off-balance-sheet effects, or financial-position concerns. In other words, a span that only mentions "prepay" is faithful to the retrieved email, but it is not necessarily supportive of the specific hypothesis.

This distinction is important for interpreting the evidence-span results. The model is reliable in quoting text from the retrieved context, but span selection is more challenging than faithfulness alone suggests. For H1, useful evidence requires not only identifying prepay-related language, but also selecting text that explains the financial or accounting role of those transactions.

5.3.2 Enriched Context

Inference Label Performance

Table 5.10 reports the useful/not-useful label performance of the inference module for each retrieval method in the enriched-context setting.

Table 5.10: Inference label performance for H1 with enriched context.

Retrieval Method	Precision	Recall	F1
BM25	0.8571	0.6667	0.7500
Dense	0.6316	0.6667	0.6486
Hybrid	0.7222	0.7222	0.7222

Context enrichment improves the inference performance for BM25 and hybrid retrieval, while slightly reducing the F1 score for dense retrieval. For BM25, F1 increases from 0.7097 to 0.7500, mainly due to improved recall. The hybrid method benefits more substantially, increasing from 0.6061 to 0.7222. This suggests that adding neighboring context can help the inference model identify useful emails when the retrieved chunk alone does not contain enough information. However, the slight decrease observed for dense retrieval indicates that context enrichment may also introduce additional surrounding text that does not always support the hypothesis, potentially making the inference decision less precise.

5.3.3 Summary of H1 Findings

The H1 results indicate that BM25 is the strongest retrieval method for this hypothesis. With a recall of 0.9444, BM25 retrieves almost all useful H1 emails, while dense retrieval achieves slightly lower recall and hybrid retrieval does not improve over BM25. This suggests that H1 is strongly lexically grounded. Terms such as “prepay”, “loan”, “financing”, “debt”, “balance sheet”, “accounting”, and “swap” appear to provide effective lexical cues, allowing BM25 to retrieve relevant evidence with high recall.

The inference stage substantially improves precision compared with retrieval alone. For BM25, retrieval precision is 0.4359, while inference precision increases to 0.8462 in the default-context setting and 0.8571 in the enriched-context setting. This supports the two-stage design of the pipeline: retrieval first identifies a broad set of potentially useful emails, and the inference module then filters these results by assessing their relationship to the hypothesis.

The enriched-context setting improves inference performance for BM25 and hybrid retrieval, but not for dense retrieval. BM25 improves from an F1 score of 0.7097 to 0.7500, while hybrid retrieval improves from 0.6061 to

0.7222. In contrast, dense retrieval decreases slightly from 0.6667 to 0.6486. This indicates that context enrichment can help when the initially retrieved chunks lack sufficient surrounding information, but it may also introduce additional noise in some cases.

5.4 Results for H2

5.4.1 Default Context

Retrieval Performance

Table 5.11 reports email-level retrieval performance for H2, treating the Useful class as the positive class.

Table 5.11: Retrieval performance for H2.

Retrieval Method	Retrieved Emails	Precision	Recall	F1
BM25	53	0.2642	0.9333	0.4118
Dense	39	0.3333	0.8667	0.4815
Hybrid	55	0.2727	1.0000	0.4286

Compared with the random retrieval baseline for H2, all hypothesis-aware retrieval methods improve retrieval performance. BM25 increases F1 from 0.2365 to 0.4118, while dense retrieval reaches the highest retrieval F1 score of 0.4815. This shows that the expanded hypothesis queries provide a substantially stronger retrieval signal than random chunk selection.

The three retrieval methods show different trade-offs. Hybrid retrieval achieves the highest recall, retrieving all useful H2 emails, but it also retrieves the largest number of emails overall, resulting in lower precision. Dense retrieval achieves the highest precision and F1 score, suggesting that semantic similarity is useful for H2. This is likely because the hypothesis involves document-handling concepts that may be expressed through related but not identical terms, such as deletion, retention, archiving, preservation, compliance, audits, investigations, and legal scrutiny. BM25 also performs strongly in recall, but its lower precision indicates that lexical document-management terms can retrieve many emails that are administrative rather than evidential.

Inference Label Performance

Table 5.12 reports the useful/not-useful label performance of the inference module for each retrieval method.

Table 5.12: Inference label performance for H2 with default context.

Retrieval Method	Precision	Recall	F1
BM25	0.6250	0.6667	0.6452
Dense	0.6111	0.7333	0.6667
Hybrid	0.6842	0.8667	0.7647

Table 5.13 reports the end-to-end useful/not-useful confusion matrices over all 100 emails in the H2 evaluation subset. The false negatives include both useful emails that were not retrieved and retrieved useful emails that were incorrectly labeled as not-useful.

Table 5.13: End-to-end useful/not-useful confusion matrices for H2.

Retrieval Method	Actual Class	Predicted Useful	Predicted Not Useful
BM25	Useful	10	5
	Not Useful	6	79
Dense	Useful	11	4
	Not Useful	7	78
Hybrid	Useful	13	2
	Not Useful	6	79

The inference stage improves precision substantially compared with retrieval alone for all three retrieval methods. This improvement is particularly important for H2, where broad document-management language can appear in both useful and non-useful emails. Terms related to archiving, retention, document collection, review, and preservation may refer either to ordinary administrative procedures or to actions relevant to regulatory, audit, or legal scrutiny.

Among the default-context inference results, hybrid retrieval achieves the highest F1 score. Although hybrid retrieval has relatively low retrieval precision, it provides the inference model with the broadest useful candidate set, including all useful H2 emails at the retrieval stage. The language model is then able to filter this candidate set and assign the Useful label to a smaller

and more accurate subset. However, the recall remains lower than retrieval-stage recall, showing that some useful emails retrieved in the first stage are not preserved by the inference module.

Reasoning and Evidence-Span Quality

Table 5.14 shows the three reasoning quality criteria: relevance, correctness, and clarity. Each criterion was assigned one of three scores: good = 1, partial = 0, and poor = -1.

Table 5.14: Reasoning quality assessment for H2 across retrieval methods.

Retrieval Method	Relevance	Correctness	Clarity
BM25	1.0000	0.9811	1.0000
Dense	1.0000	0.9744	1.0000
Hybrid	1.0000	0.9636	1.0000

Most generated explanations were judged to be relevant, correct, and clear. The relevance and clarity scores are 1.0000 across all retrieval methods, indicating that the explanations generally addressed the relationship between the email context and the hypothesis in an understandable way.

The small reductions in correctness are caused by a limited number of partially correct explanations. For BM25, email 914 was marked as partially correct because the retrieved context includes the phrase “please archive these tapes in a secure location,” while the explanation states that there is no mention of selective retention. Since archiving tapes can be interpreted as a form of retention, the explanation was not fully correct. However, the case was not marked as completely incorrect because the retrieved context does not explicitly connect the retention action to legal, regulatory, or audit-related anticipation.

For dense retrieval, email 93 was marked as partially correct because the explanation inferred a connection to scrutiny from an electricity-pricing context, although the retrieved context did not explicitly mention litigation, subpoenas, audits, or regulatory investigation. For hybrid retrieval, emails 334 and 914 were marked as partially correct. In email 334, the explanation described the context as highly relevant to preparing for scrutiny or audit, but the retrieved context mainly showed document collection and compliance pressure without an explicit legal or regulatory motivation. Email 914 was treated

similarly to the BM25 case because the context refers to archiving tapes while the explanation denies selective retention.

Table 5.15 shows the evidence-span quality criteria: supportiveness and faithfulness. Each criterion was scored as yes = 1 and no = 0.

Table 5.15: Evidence-span quality assessment for H2 across retrieval methods.

Retrieval Method	Total Spans	Supportiveness	Faithfulness
BM25	28	0.9643	1.0000
Dense	33	0.9697	1.0000
Hybrid	36	0.9444	1.0000

The evidence-span evaluation shows high faithfulness across all retrieval methods, indicating that the extracted spans were present in the retrieved contexts. Supportiveness is also high, but a small number of spans were judged not to provide sufficient independent support for the H2 hypothesis.

For BM25, one span was not counted as supportive because it referred to retention timing being driven by database space availability. This provides an administrative explanation rather than evidence of document handling in anticipation of scrutiny. For dense retrieval, the non-supportive span from email 599 concerned avoiding discussion with other companies or the press, but did not directly support document alteration, deletion, retention, preservation, or legal/regulatory anticipation. For hybrid retrieval, two spans were judged too vague on their own: “remind your troops of the importance of this effort” and “consider revising and resending.” These spans require surrounding context to become meaningful and were therefore not counted as independently supportive evidence.

5.4.2 Enriched Context

Inference Label Performance

Table 5.16 reports the useful/not-useful label performance of the inference module for H2 in the enriched-context setting.

Table 5.16: Inference label performance for H2 with enriched context.

Retrieval Method	Precision	Recall	F1
BM25	0.6715	0.8000	0.7301
Dense	0.6506	0.7333	0.6895
Hybrid	0.6717	0.9333	0.7812

Context enrichment improves inference performance for all three retrieval methods in H2. BM25 improves from an F1 score of 0.6452 in the default-context setting to 0.7301 in the enriched-context setting. Dense retrieval improves from 0.6667 to 0.6895, while hybrid retrieval improves from 0.7647 to 0.7812. Therefore, the enriched-context setting is beneficial for H2 across all retrieval methods, although the size of the improvement differs.

The largest improvement is observed for BM25. This suggests that neighboring context helps the inference model interpret document-handling references that may be ambiguous when viewed in isolation. For H2, terms such as archiving, retention, deletion, document collection, review, and preservation can appear in ordinary administrative contexts, but they become more relevant to the hypothesis when surrounding text connects them to legal, regulatory, audit-related, or investigative concerns.

Hybrid retrieval remains the strongest method in the enriched-context setting. This is consistent with the default-context results, where hybrid retrieval already achieved the highest inference F1 score. The result suggests that H2 benefits from a broad candidate set at the retrieval stage, while context enrichment further helps the inference model distinguish between general document-management language and evidence that is more directly related to the hypothesis.

5.4.3 Summary of H2 Findings

The H2 results show a different pattern from H1. While H1 was strongly supported by explicit financial terminology, H2 involves a broader and more ambiguous set of document-handling concepts, including deletion, retention, archiving, preservation, shredding, compliance, and legal or regulatory scrutiny. As a result, dense and hybrid retrieval are more competitive for H2 than they were for H1.

In the retrieval stage, hybrid retrieval achieves the highest recall by retrieving all useful H2 emails, but dense retrieval achieves the highest precision and F1 score. This suggests that semantic similarity is useful for identifying document-handling evidence when relevant emails do not use exactly the same terms as the hypothesis. At the same time, the low precision of all retrieval methods shows that H2 is a difficult retrieval task because many non-useful emails contain general document-management vocabulary.

In the inference stage, hybrid retrieval achieves the strongest performance in both the default-context and enriched-context settings. This indicates that the inference module can benefit from the broader candidate set produced by hybrid retrieval and can filter many of the non-useful emails retrieved in the first stage. Context enrichment further improves all three methods, suggesting that neighboring chunks help clarify whether document-handling references are merely administrative or connected to scrutiny, audits, investigations, or legal concerns.

The reasoning and evidence-span evaluations also indicate generally reliable generated outputs. Reasoning relevance and clarity are perfect across all retrieval methods, while correctness is slightly lower due to a small number of cases where the model either understated retention-related evidence or inferred legal/regulatory motivation more strongly than the retrieved context supported. Evidence-span faithfulness is 1.0000 across all methods, showing that the quoted spans are grounded in the retrieved text. Supportiveness is also high, although a few spans were too administrative or too vague to independently support the hypothesis.

5.5 Results for H3

5.5.1 Default Context

Retrieval Performance

Table 5.17 reports email-level retrieval performance for H3, treating the Useful class as the positive class.

Table 5.17: Retrieval performance for H3.

Retrieval Method	Retrieved Emails	Precision	Recall	F1
BM25	42	0.3333	0.7000	0.4516
Dense	48	0.3542	0.8500	0.5000
Hybrid	44	0.2955	0.6500	0.4062

Compared with the random retrieval baseline for H3, all hypothesis-aware retrieval methods improve retrieval performance. The random retrieval baseline reaches an average F1 score of 0.3080 for H3, while BM25, dense retrieval, and hybrid retrieval reach F1 scores of 0.4516, 0.5000, and 0.4062, respectively. This shows that the expanded hypothesis queries provide a stronger retrieval signal than random chunk selection.

Among the three retrieval methods, dense retrieval achieves the highest recall, precision, and F1 score. This suggests that H3 benefits from semantic retrieval more than from purely lexical matching. The hypothesis involves energy-market concepts such as schedules, bids, load volumes, supply availability, and market prices. These ideas may be expressed indirectly through operational or market-related language rather than through exact repetitions of the hypothesis terms. Dense retrieval appears better able to capture these semantically related expressions. BM25 also improves over the random baseline, but its lower recall indicates that exact lexical matching does not retrieve as many useful H3 emails. Hybrid retrieval performs below dense retrieval, suggesting that combining sparse and dense signals does not provide an additional benefit for this hypothesis.

Inference Label Performance

Table 5.18 reports the useful/not-useful label performance of the inference module for each retrieval method.

Table 5.18: Inference label performance for H3 with default context.

Retrieval Method	Precision	Recall	F1
BM25	0.8571	0.6000	0.7059
Dense	0.7647	0.6500	0.7027
Hybrid	0.6667	0.6000	0.6316

Table 5.19 reports the end-to-end useful/not-useful confusion matrices

over all 100 emails in the H3 evaluation subset. The false negatives include both useful emails that were not retrieved and retrieved useful emails that were incorrectly labeled as not-useful.

Table 5.19: End-to-end useful/not-useful confusion matrices for H3.

Retrieval Method	Actual Class	Predicted Useful	Predicted Not Useful
BM25	Useful	12	8
	Not Useful	2	78
Dense	Useful	13	7
	Not Useful	4	76
Hybrid	Useful	12	8
	Not Useful	6	74

The inference stage substantially improves precision compared with retrieval alone for all three retrieval methods. For BM25, precision increases from 0.3333 at the retrieval stage to 0.8571 after inference. Dense retrieval increases from 0.3542 to 0.7647, while hybrid retrieval increases from 0.2955 to 0.6667. This shows that the inference module is effective as a filtering component, assigning the Useful label to a smaller and more accurate subset of the retrieved emails.

However, this increase in precision comes with lower end-to-end recall. The final pipeline can lose useful emails at either stage: the retrieval component may fail to retrieve them, or the inference component may classify retrieved useful emails as not useful. Dense retrieval remains the strongest method in the default-context inference setting, achieving the highest recall and an F1 score close to BM25. BM25 achieves the highest precision, but identifies fewer useful emails than dense retrieval. Hybrid retrieval performs lower than the other two methods, indicating that the additional candidates introduced by the hybrid retrieval strategy do not lead to better inference performance for H3.

For H3, this trade-off is important because the hypothesis concerns operational energy-market behavior, where useful evidence may be indirect. Some emails may discuss schedules, bids, load volumes, or market conditions without explicitly stating an intention to affect supply availability or prices. The inference module improves precision by filtering weak candidates, but it may

also reject some useful emails when the connection to the hypothesis is implicit.

Reasoning and Evidence-Span Quality

Table 5.20 shows the three reasoning quality criteria: relevance, correctness, and clarity. Each criterion was assigned one of three scores: good = 1, partial = 0, and poor = -1.

Table 5.20: Reasoning quality assessment for H3 across retrieval methods.

Retrieval Method	Relevance	Correctness	Clarity
BM25	1.0000	0.9524	1.0000
Dense	1.0000	0.8958	1.0000
Hybrid	1.0000	0.8636	1.0000

The reasoning-quality results show that relevance and clarity are perfect across all three retrieval methods, indicating that the generated explanations generally address the H3 hypothesis and are understandable. The lower correctness scores show that the main issue is not readability or topic alignment, but the degree to which the explanation accurately reflects the strength of the retrieved evidence.

For BM25, one explanation was judged as incorrect. This case corresponds to email 827, where the retrieved context concerns moving “Enovate” out of a VaR or risk-management position and its effect on central desk VaR. The generated explanation interpreted this as relevant reallocation for H3. However, the context does not provide evidence about energy schedules, bids, load volumes, supply availability, or market prices. Therefore, the explanation was treated as incorrect rather than partially correct.

For dense retrieval, five explanations were marked as partially correct. These cases were relevant and understandable, but they slightly overinterpreted weak or indirect evidence. For example, email 336 mentions gas supply and mark-to-market position, but does not clearly show adjustment, restriction, or reallocation of energy schedules, bids, or load volumes. Similarly, email 425 mentions bids in ancillary and imbalance energy markets, but does not clearly show that those bids were adjusted or restricted in a way that could affect supply availability or market prices.

For hybrid retrieval, six explanations were marked as partially correct. These cases follow the same pattern: the retrieved context is generally related to energy-market activity, but the generated explanation sometimes presents the evidence as stronger than it is. Overall, the reasoning results suggest that the model can produce clear and relevant explanations for H3, but it may overstate the connection between general energy-market discussion and the more specific hypothesis.

Table 5.21 shows the evidence-span quality criteria: supportiveness and faithfulness. Each criterion was scored as yes = 1 and no = 0.

Table 5.21: Evidence-span quality assessment for H3 across retrieval methods.

Retrieval Method	Total Spans	Supportiveness	Faithfulness
BM25	35	0.9429	1.0000
Dense	28	0.7143	1.0000
Hybrid	33	0.7576	0.9697

The evidence-span evaluation shows that faithfulness is high across all retrieval methods. BM25 and dense retrieval achieve a faithfulness score of 1.0000, meaning that all extracted spans were present in the retrieved contexts. Hybrid retrieval achieves a slightly lower faithfulness score of 0.9697. The only faithfulness issue occurs in email 902, where the extracted span differs slightly from the retrieved text: the extracted span says “began last year”, while the retrieved context says “began early last year”. Under the stricter exact-span criterion, this was counted as a minor faithfulness error.

Supportiveness varies more strongly across retrieval methods. BM25 achieves the highest supportiveness score, with 33 supportive spans out of 35. The two non-supportive spans are associated with email 827. Although these spans are faithful to the retrieved context, they do not support H3 because they concern VaR or risk-management position movement rather than energy schedules, bids, load volumes, supply availability, or market prices.

Dense retrieval has lower supportiveness, with 20 supportive spans out of 28. This indicates that dense retrieval often retrieves semantically related energy-market contexts, but the extracted spans do not always provide direct evidence for the specific hypothesis. Hybrid retrieval shows a similar pattern, with 25 supportive spans out of 33. Overall, the evidence-span results suggest that the model is generally reliable in quoting from the retrieved context, but

span supportiveness is more difficult for H3 because the hypothesis requires evidence of a specific operational effect, not merely general energy-market discussion.

5.5.2 Enriched Context

Inference Label Performance

Table 5.22 reports the useful/not-useful label performance of the inference module for each retrieval method in the enriched-context setting.

Table 5.22: Inference label performance for H3 with enriched context.

Retrieval Method	Precision	Recall	F1
BM25	0.8000	0.6000	0.6857
Dense	0.7778	0.7000	0.7368
Hybrid	0.7059	0.6000	0.6486

Context enrichment has mixed effects for H3. BM25 decreases from an F1 score of 0.7059 in the default-context setting to 0.6857 in the enriched-context setting. In contrast, dense retrieval improves from 0.7027 to 0.7368, while hybrid retrieval improves slightly from 0.6316 to 0.6486. Therefore, the enriched-context setting benefits dense and hybrid retrieval, but not BM25.

Dense retrieval achieves the strongest enriched-context performance, with the highest recall and F1 score. This suggests that additional neighboring context is particularly useful when the retrieved evidence is semantically related to the hypothesis but may not be fully interpretable from the retrieved chunk alone. Since H3 concerns operational energy-market behavior, relevant evidence may be distributed across nearby sentences or chunks involving bids, schedules, load volumes, supply, or prices.

At the same time, the decrease observed for BM25 indicates that context enrichment is not universally beneficial. BM25 already retrieves lexically strong chunks, and adding neighboring text may introduce surrounding information that weakens or distracts from the original evidence. Overall, the enriched-context results suggest that context expansion is most helpful for H3 when the retrieval method captures semantically relevant but context-dependent evidence.

5.5.3 Summary of H3 Findings

The H3 results show that dense retrieval is the strongest retrieval method for this hypothesis. Unlike H1, which is supported by relatively explicit financial terminology, H3 involves operational energy-market concepts that may be expressed through varied and indirect language. References to schedules, bids, load volumes, supply, prices, and market positions do not always appear in the same lexical form as the hypothesis. This helps explain why dense retrieval achieves the highest retrieval recall and F1 score.

The inference stage improves precision substantially for all retrieval methods, showing that the language model is effective at filtering retrieved candidates. However, recall decreases compared with the retrieval stage, indicating that some useful emails are not preserved after inference. This reflects the difficulty of H3: useful evidence may be indirect, and the model may require a clear connection between energy-market discussion and a possible effect on supply availability or market prices before assigning the useful label.

The reasoning-quality evaluation shows that the generated explanations are consistently relevant and clear, but correctness is lower than in H1 and H2. The main issue is over-interpretation. In several cases, the retrieved context is related to energy-market activity, but the explanation presents it as stronger evidence for schedule, bid, or load manipulation than the text directly supports.

The evidence-span evaluation shows a similar pattern. Faithfulness is high, meaning that the model usually quotes spans that are actually present in the retrieved context. Supportiveness is lower, especially for dense and hybrid retrieval, because some quoted spans are only generally related to energy-market activity and do not independently support the specific H3 hypothesis.

Context enrichment has different effects across retrieval methods. It slightly reduces BM25 performance, but improves dense and hybrid retrieval. Dense retrieval benefits the most and becomes the strongest method in the enriched-context setting. Overall, H3 demonstrates that semantic retrieval and context enrichment are valuable for operational energy-market hypotheses, but it also highlights the difficulty of distinguishing general market discussion from evidence that directly supports the investigative claim.

5.6 Multi-Run Analysis

In the previous sections, the results were reported in detail for each hypothesis based on a single pipeline run. However, the query expansion stage may generate different expanded queries across runs, which can affect the retrieved contexts and, consequently, the downstream inference labels. Therefore, this section evaluates the stability of the full pipeline across four runs for each hypothesis. The same configuration parameters are used as in the previous experiments, with the difference that all four runs are executed using context enrichment with a window size of 1.

To keep the analysis focused, this section considers only the useful/not-useful inference label metrics. Retrieval metrics, reasoning quality, and evidence-span quality are not repeated, in order to avoid an excessive number of additional tables. The goal is to examine the robustness of the pipeline when the expanded queries change, and to analyze whether dense retrieval contributes complementary evidence to BM25 in the hybrid setting.

The results are reported using precision, recall, and F1 score, together with the mean and standard deviation across the four runs. Since only four runs are used, the standard deviation should be interpreted as an indicator of run-to-run variation rather than as a formal statistical estimate.

5.6.1 Multi-run results of H1

Figure 5.1 shows the precision, recall, and F1 scores obtained across four runs for H1 using the three retrieval methods.

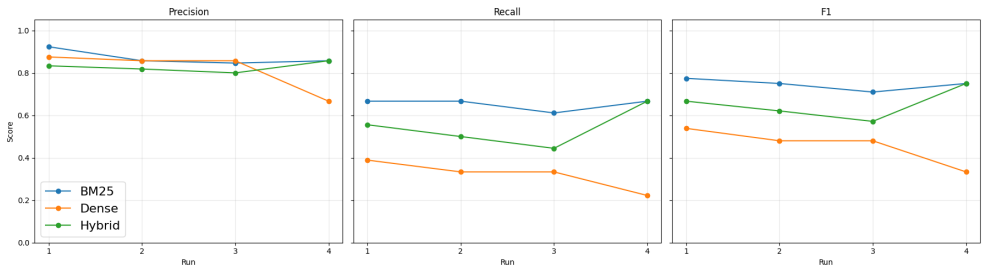

Figure 5.1: Inference label metrics for H1 across four runs.

Table 5.23 shows the average performance of each retrieval method across the runs.

Table 5.23: Mean and standard deviation of H1 inference label metrics across four runs.

Retrieval Method	Precision	Recall	F1
BM25	0.871 ± 0.035	0.653 ± 0.028	0.746 ± 0.027
Dense	0.814 ± 0.099	0.319 ± 0.070	0.458 ± 0.088
Hybrid	0.827 ± 0.024	0.542 ± 0.095	0.652 ± 0.076

For H1, BM25 is the strongest and most stable method across the four runs. It achieves the highest mean F1 score of 0.746, with a low standard deviation of 0.027. It also has the highest recall among the three methods and shows limited variation across runs. This confirms the pattern observed in the single-run analysis: H1 is strongly supported by lexical cues related to prepay transactions, loans, financing, accounting treatment, and balance-sheet effects.

Dense retrieval achieves relatively high precision, but its recall is substantially lower, with a mean recall of 0.319. This indicates that dense retrieval identifies a smaller set of useful emails and misses many useful cases for this hypothesis. Hybrid retrieval improves over dense retrieval, but it does not outperform BM25. This suggests that, for H1, dense retrieval does not provide enough complementary useful evidence to improve the BM25 results. Instead, the additional dense candidates may introduce noise into the hybrid ranking.

5.6.2 Multi-run results of H2

Figure 5.2 shows the precision, recall, and F1 scores obtained across four runs for H2 using the three retrieval methods.

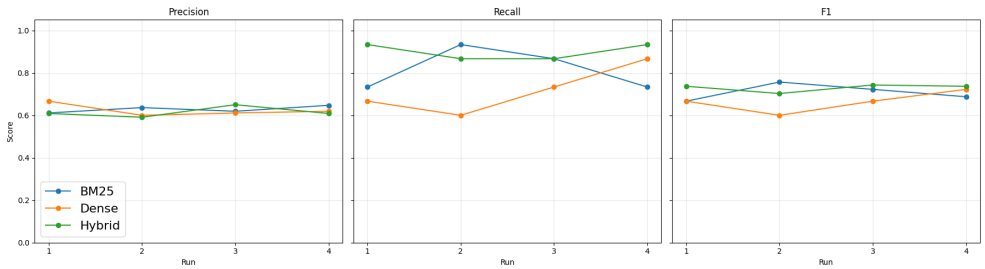

Figure 5.2: Inference label metrics for H2 across four runs.

Table 5.24 shows the average performance of each retrieval method across the runs.

Table 5.24: Mean and standard deviation of H2 inference label metrics across four runs.

Retrieval Method	Precision	Recall	F1
BM25	0.628 ± 0.016	0.817 ± 0.100	0.708 ± 0.040
Dense	0.624 ± 0.029	0.717 ± 0.114	0.664 ± 0.050
Hybrid	0.615 ± 0.025	0.900 ± 0.038	0.730 ± 0.018

For H2, hybrid retrieval achieves the strongest overall performance across the four runs. It obtains the highest mean recall of 0.900 and the highest mean F1 score of 0.730. It also has the lowest standard deviation for F1, indicating that its overall performance is more stable than BM25 and dense retrieval.

BM25 and dense retrieval show similar precision, but both have larger variation in recall. This suggests that their ability to retrieve and preserve useful H2 emails is more sensitive to the particular expanded queries generated in each run. Hybrid retrieval appears to reduce this sensitivity by combining lexical and semantic signals.

These results support the interpretation that H2 benefits from complementarity between retrieval methods. Dense retrieval alone performs below BM25 in terms of F1, but when combined with BM25 in the hybrid setting, it appears to contribute additional useful candidates. This helps hybrid retrieval outperform both individual methods for H2.

5.6.3 Multi-run results of H3

Figure 5.3 shows the precision, recall, and F1 scores obtained across four runs for H3 using the three retrieval methods.

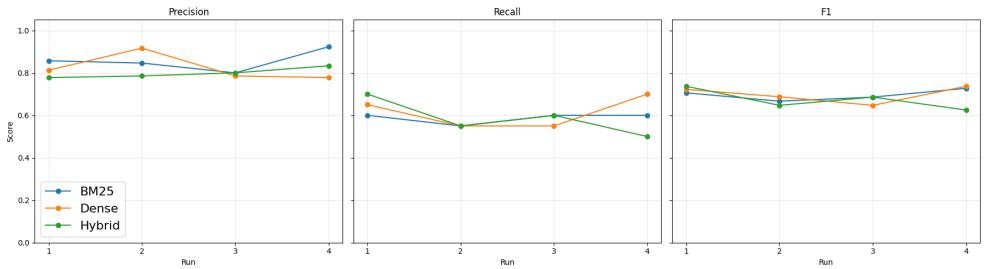

Figure 5.3: Inference label metrics for H3 across four runs.

Table 5.25 shows the average performance of each retrieval method across the runs.

Table 5.25: Mean and standard deviation of H3 inference label metrics across four runs.

Retrieval Method	Precision	Recall	F1
BM25	0.857 ± 0.051	0.588 ± 0.025	0.696 ± 0.026
Dense	0.823 ± 0.064	0.613 ± 0.075	0.698 ± 0.040
Hybrid	0.799 ± 0.025	0.588 ± 0.085	0.674 ± 0.049

For H3, the three methods show relatively close performance. Dense retrieval achieves the highest mean recall and the highest mean F1 score, but the difference compared with BM25 is small. BM25 has the highest mean precision and the lowest standard deviation for recall and F1, indicating more stable performance across runs.

This suggests that H3 does not have a single clearly dominant method. If recall is prioritized, dense retrieval is slightly preferable because it identifies more useful emails on average. If precision and stability are prioritized, BM25 is competitive and may be preferable. Hybrid retrieval performs slightly below both BM25 and dense retrieval, suggesting that combining sparse and dense retrieval does not provide a clear benefit for H3 in the multi-run setting.

Overall, the H3 multi-run results are consistent with the earlier single-run analysis: semantic retrieval is useful for this hypothesis, but the performance gap between BM25 and dense retrieval is small. This reflects the difficulty of H3, where useful evidence may be expressed through varied energy-market language but still requires a specific connection to schedules, bids, load volumes, supply availability, or prices.

5.6.4 Summary of Findings

Overall, the multi-run analysis confirms that the best retrieval strategy depends on the hypothesis. BM25 is strongest for H1, where the evidence is closely tied to explicit lexical cues. Hybrid retrieval is strongest for H2, where dense retrieval appears to complement BM25 by adding useful semantically related candidates. For H3, BM25 and dense retrieval are closely matched, with dense retrieval offering slightly higher recall and BM25 offering stronger precision and stability.

These results show that query variation affects the pipeline differently

across hypotheses. The standard deviations are generally moderate, suggesting that the system is reasonably stable across runs. However, the variation in recall, especially for dense and hybrid retrieval in some hypotheses, indicates that the expanded query set can influence which useful emails are retrieved and preserved by the inference stage.

5.7 Discussion

The previous sections evaluated the proposed pipeline from several perspectives, including retrieval performance, useful/not-useful inference, reasoning quality, evidence-span quality, context enrichment, and multi-run stability. This section summarizes the main implications of these results and the broader patterns observed across the experiments.

5.7.1 Retrieval and Inference as Complementary Stages

The results show that retrieval and inference play different but complementary roles in the pipeline. Retrieval mainly functions as a candidate-generation stage: it aims to retrieve a broad set of potentially useful emails so that relevant evidence is available to the language model. This is particularly important in a forensic setting, because an email that is not retrieved cannot later be interpreted, explained, or used as evidence by the inference stage.

Across the experiments, retrieval generally achieves high recall but lower precision. This indicates that the retrieval methods are effective at collecting candidate emails, but they also return many emails that are only topically related to the hypothesis. The inference stage partly addresses this limitation by filtering the retrieved emails according to their actual usefulness for the hypothesis. In all three hypotheses, inference improves precision compared with retrieval alone, showing that the language model can reject many weakly related candidates.

However, this precision gain comes with lower end-to-end recall. A useful email may be absent from the final output because it was not retrieved or because it was retrieved but rejected by the inference module. When inference recall is lower than retrieval recall, the difference specifically represents

useful retrieved emails that were not preserved by the inference stage. Therefore, the pipeline should be understood as a trade-off: retrieval provides coverage, while inference improves selectivity. This supports the two-stage design, since a retrieval-only system may return too much irrelevant material, while inference can only operate on evidence that was successfully retrieved.

5.7.2 Differences Between BM25, Dense, and Hybrid Retrieval

The experiments show that no single retrieval method is best for all hypotheses. The effectiveness of BM25, dense retrieval, and hybrid retrieval depends on how the useful evidence is expressed in the emails.

BM25 performs strongest when useful emails contain explicit lexical cues that overlap with the expanded queries. This is most visible in H1, where the evidence is closely connected to financial terminology such as prepay transactions, loans, financing, debt, accounting treatment, and balance-sheet effects. In this type of setting, sparse retrieval is highly effective because the relevant evidence is tied to specific terms.

Dense retrieval becomes more useful when the hypothesis is expressed through broader, more indirect, or more varied language. This is particularly relevant for H3, where useful emails may discuss schedules, bids, load volumes, supply, prices, or market conditions without using the exact wording of the hypothesis. In such cases, semantic similarity helps retrieve emails that are conceptually related to the hypothesis even when lexical overlap is weaker.

Hybrid retrieval does not automatically improve performance. Its usefulness depends on whether BM25 and dense retrieval contribute complementary evidence. For H2, hybrid retrieval is beneficial because the hypothesis involves both explicit document-handling terms and broader contextual signals related to audits, legal scrutiny, regulatory concerns, or preservation. In this case, the broader candidate set produced by hybrid retrieval can be effectively filtered by the inference module. In contrast, for H1 and H3, hybrid retrieval does not consistently outperform the best individual method, suggesting that combining retrievers can also introduce additional noise when their results are not sufficiently complementary.

Overall, the retrieval strategy should be selected according to the nature of the investigative hypothesis. Lexically explicit hypotheses benefit strongly

from BM25, semantically broader hypotheses benefit from dense retrieval, and hybrid retrieval is most useful when lexical and semantic signals capture different useful emails.

5.7.3 Effect of Context Enrichment

The enriched-context experiments show that adding neighboring chunks can improve inference performance, but the effect is not uniform across all hypotheses and methods. Context enrichment is useful when the retrieved chunk contains only part of the relevant information and surrounding text is needed to interpret the evidence correctly. This is especially important for email threads, where the meaning of a sentence or paragraph may depend on earlier or later parts of the conversation.

The benefit of enrichment is clearest when the retrieved evidence is ambiguous in isolation. For example, document-handling language in H2 may refer to ordinary administrative activity, but surrounding context can clarify whether it is connected to legal, regulatory, audit-related, or investigative concerns. Similarly, in H3, neighboring context can help interpret operational energy-market language that may otherwise appear too general.

At the same time, enrichment can introduce irrelevant surrounding text. This may weaken the evidence signal and make the inference decision less precise. For this reason, context enrichment should not be treated as a guaranteed improvement. It is most useful when the retrieved chunk is incomplete, but it may be less helpful when the original chunk already contains focused and sufficient evidence.

This finding reflects an important trade-off in RAG-based forensic analysis. Narrow chunks improve focus and reduce noise, while broader contexts preserve surrounding meaning. The best setting depends on whether the task requires precise evidence matching or broader contextual interpretation.

5.7.4 Stability Across Multiple Runs

The multi-run analysis shows that the pipeline is affected by variation in the query expansion stage. Since expanded queries can differ across runs, the retrieved contexts and downstream inference labels can also change. This is important because it shows that a single run may not fully represent the behavior of a pipeline that relies on LLM-generated query expansion.

At the same time, the multi-run results support the main patterns observed in the single-run experiments. BM25 remains strongest for H1, where the useful evidence is closely tied to explicit financial terminology. Hybrid retrieval remains strongest for H2, where combining lexical and semantic retrieval provides a useful candidate set for inference. For H3, dense retrieval and BM25 remain close, with dense retrieval showing advantages in recall and F1 while BM25 remains competitive in precision and stability.

These results suggest that the pipeline is reasonably consistent at the level of broader methodological conclusions, even though individual runs vary. The best retrieval strategy still depends on the hypothesis, and the same general interpretation holds across both the single-run and multi-run analyses. Nevertheless, the observed variation indicates that future evaluations should include multiple runs by default when LLM-generated query expansion is used.

5.7.5 Evidence and Reasoning Quality

The evaluation of reasoning and evidence spans shows that the generated outputs are useful for interpretation, but they should not be treated as final forensic judgments. Reasoning relevance and clarity are consistently strong, indicating that the model usually produces explanations that address the hypothesis and are understandable. This supports the interpretability goal of the pipeline, since the system not only returns emails but also explains why an email may or may not be useful.

Correctness is more challenging. The main issue is not that explanations are unrelated, but that the model can sometimes overstate or understate the strength of the evidence. This is especially relevant when an email is topically related to the hypothesis but does not directly support the full investigative claim. In such cases, the explanation may make the evidence appear stronger than the retrieved text justifies.

The evidence-span evaluation shows a similar distinction. Faithfulness is generally high, meaning that the extracted spans are usually present in the retrieved context and are not fabricated. However, supportiveness is lower than faithfulness. A quoted span can be faithful because it appears in the email, but still not sufficiently supportive if it only refers to the general topic without establishing the specific relationship required by the hypothesis.

This distinction is central to evidence-grounded forensic analysis. The system is reliable in grounding its outputs in retrieved text, but selecting the most independently supportive span remains difficult. Therefore, evidence spans should be treated as review aids rather than definitive proof. They help direct the investigator toward relevant parts of the email, but the final interpretation still requires human verification.

5.7.6 Implications for Hypothesis-Driven Forensic Email Analysis

Taken together, the results support the usefulness of a RAG-based pipeline for hypothesis-driven forensic email analysis. The pipeline improves over random retrieval, identifies useful emails across different types of hypotheses, and produces interpretable outputs in the form of labels, explanations, and evidence spans. This is important because the task is not only to search for emails, but to assess whether emails are useful in relation to an investigative hypothesis.

The results also show that hypothesis-driven retrieval is sensitive to the form of the hypothesis. Some hypotheses are lexically grounded and can be handled effectively by BM25. Others require semantic retrieval because the evidence is expressed through indirect or domain-specific language. Hybrid retrieval is useful when sparse and dense methods contribute complementary candidates that the inference module can filter effectively.

The inference module adds value by improving precision and providing explanations, but it also introduces the risk of losing useful emails or over-interpreting weak evidence. For this reason, the proposed system should be viewed as an investigative support tool rather than an automated decision-maker. Its main value lies in reducing the search space, highlighting potentially useful emails, and making predictions inspectable through evidence spans and reasoning.

Overall, the evaluation demonstrates that combining retrieval, context enrichment, and language-model inference can support hypothesis-driven email analysis. However, the success of the approach depends on the interaction between the hypothesis, the retrieval method, the available context, and the strictness of the inference decision. This reinforces the need to evaluate RAG-based forensic systems as complete pipelines rather than judging retrieval or

generation in isolation.

Chapter 6

Conclusion

6.1 Key Findings

The results of this thesis show that the proposed RAG-based pipeline can support hypothesis-driven forensic email analysis by combining retrieval, inference, evidence-span extraction, and explanation generation. The system is able to identify useful emails for different investigative hypotheses and provide interpretable outputs that help explain why an email may be relevant.

A central finding is that the inference module is comparatively reliable when it receives sufficient and appropriate context. In many cases, the language model is able to filter retrieved emails effectively, improve precision, and generate understandable explanations. However, when the retrieved context is incomplete, too narrow, or weakly related to the hypothesis, the inference module cannot fully compensate for this limitation. This indicates that retrieval is the main bottleneck of the pipeline.

The experiments also show that query expansion is an important but sensitive part of the retrieval process. In the current pipeline, the expanded queries are generated only from the hypothesis, without considering the vocabulary and language actually used in the corpus. As a result, some expanded queries may be semantically reasonable but not well aligned with the way relevant evidence is expressed in the emails. This can affect both retrieval performance and run-to-run stability.

Another finding is that context enrichment is useful only when the added context is relevant. Adding neighboring chunks can help when the retrieved chunk is incomplete or ambiguous, especially in email threads where meaning may depend on surrounding messages. However, blind enrichment can also

introduce unrelated text and reduce performance. Therefore, context enrichment should be treated as a conditional improvement rather than a generally positive step.

The results further show that the best retrieval method depends on the nature of the hypothesis. BM25 performs well when useful evidence is tied to explicit terms and domain-specific nouns, while dense retrieval is more useful when evidence is expressed indirectly or through broader semantic relations. Hybrid retrieval is effective when sparse and dense retrieval provide complementary candidates, but it does not automatically outperform the individual methods.

Finally, the experiments suggest that ranking and top-m selection can cause useful chunks to be lost before they reach the inference stage. Useful chunks are not always ranked highly enough by the default BM25, dense, or hybrid scoring procedures. Therefore, even when relevant information exists in the corpus, the pipeline may fail if that information is not selected among the final retrieved chunks. This reinforces the importance of improving retrieval and ranking before focusing on more complex inference strategies.

6.2 Limitations

The main limitation of this thesis is related to the dataset. A public dataset directly suited to hypothesis-driven forensic email analysis, with hypothesis-level relevance labels and evidence annotations, was not available. Therefore, a task-specific subset had to be constructed and manually annotated. Due to the cost and time required for annotation, the final evaluation was limited to three hypotheses with 100 emails each. Although this was sufficient for evaluating the proposed pipeline in a controlled setting, it limits the extent to which the results can be generalized to larger forensic collections.

A second limitation concerns the granularity of the annotation. In this work, emails were annotated at the document level as relevant, partially relevant, or not relevant, and these labels were later converted into the useful/not-useful setting used in the main evaluation. However, the dataset does not include manually annotated gold evidence spans. As a result, the evidence-span evaluation could assess whether extracted spans were faithful to the retrieved context and supportive of the hypothesis, but it could not measure span-level

precision or recall against gold annotations. This makes the evaluation of evidence extraction less complete than it would be with sentence-level, clause-level, or span-level ground truth.

Another limitation is the limited diversity of the evaluated hypotheses. The three hypotheses were selected to represent different investigative themes, including financial transactions, document handling, and energy-market communication. However, they still cover only a small part of the possible hypothesis types that may occur in forensic or legal investigations. More hypotheses would be needed to draw stronger conclusions about when BM25, dense retrieval, or hybrid retrieval is most appropriate.

The evaluation is also limited to email communication. Although the proposed approach is intended to be relevant to broader communication data, such as chat logs, reports, or other text-based investigative documents, these sources were not tested experimentally. Email threads have specific structural properties, such as replies, quoted messages, and message blocks, so the results may not transfer directly to other document types without additional adaptation.

Finally, the experiments were conducted with a fixed set of system choices, including the embedding model, the number of expanded queries, the query expansion strategy, the inference model, and the retrieval budget. Keeping these parameters fixed allowed the pipeline to be evaluated consistently, but it also means that the thesis does not provide a full optimization study. Alternative choices for any of these parameters may lead to different results.

6.3 Future Work

6.3.1 Retrieval and Pipeline

The findings of this thesis suggest several directions for future work. The first direction is to improve query expansion by making it more corpus-aware. In the current pipeline, expanded queries are generated from the hypothesis alone, without feedback from the target corpus. Future work could address this by incorporating pseudo-relevance feedback methods such as Rocchio relevance feedback, RM3, or dense-retrieval variants such as ANCE-PRF. More recent LLM-based approaches, such as Corpus-Steered Query Expansion (CSQE) and LLM-assisted pseudo-relevance feedback, are especially relevant because

they use initially retrieved documents or passages to guide query reformulation [26, 27, 53, 54, 55, 56, 57]. Such approaches could help align expanded queries with the vocabulary and phrasing actually used in the email corpus, while reducing the risk of corpus-misaligned or overly generic LLM-generated queries.

A second direction is to replace blind context enrichment with structure-aware context selection. In this thesis, context enrichment was performed by adding neighboring chunks around the retrieved chunk. However, neighboring chunks do not always belong to the same paragraph or the same part of the email thread, and this can introduce unrelated information. Future work could store additional structural metadata in the database, such as message-block identifiers, paragraph identifiers, and chunk positions. During enrichment, the system could then add only neighboring chunks that belong to the same paragraph or the same message block. This would preserve useful surrounding context while reducing the risk of adding noise from unrelated parts of the email.

Another important direction is to improve ranking before the final top-m selection. The experiments suggest that useful chunks may be lost when they are not ranked highly enough by the initial BM25, dense, or hybrid retrieval scores. Future work could introduce a second-stage reranking step after first-stage retrieval. For example, BM25, dense, or hybrid retrieval could first return a larger candidate set, and a cross-encoder reranker could then score each hypothesis–chunk pair more precisely before selecting the final top-m chunks for inference. This would increase the chance that useful evidence is placed near the top of the retrieved results and passed to the language model.

Finally, future work could investigate adaptive retrieval strategies. The results showed that the best retrieval method depends on the nature of the hypothesis. Some hypotheses are strongly tied to explicit lexical terms, while others depend more on semantic relations, actions, or broader contextual descriptions. A future system could therefore first analyze the hypothesis and then choose between BM25, dense retrieval, hybrid retrieval, or different weights in the hybrid strategy. This would make the pipeline less dependent on a fixed retrieval configuration and more responsive to the linguistic structure of the investigative hypothesis.

6.3.2 Dataset, Annotation, and Generalization

In addition to improving the retrieval pipeline, future work should also address the limitations of the dataset and annotation process. A larger annotated dataset would make it possible to evaluate the system more reliably and test whether the observed patterns hold across a wider range of forensic scenarios. This would require expanding the number of emails and the number of useful examples available for each hypothesis.

Future datasets should also include more detailed annotation. In this thesis, emails were annotated only at document level, which allowed useful and not-useful emails to be evaluated, but did not provide gold evidence spans. A more detailed annotation scheme could mark the exact sentences, clauses, or spans that support each hypothesis. For example, the CLAUDETTE corpus provides clause-level annotation for Terms of Service documents, showing how finer-grained annotation can support more detailed evaluation than document-level labels alone [58]. Applying a similar idea to forensic email analysis would make it possible to evaluate evidence extraction more directly using span-level precision, recall, and F1.

Another direction is to design and evaluate a broader range of hypotheses. The hypotheses used in this thesis follow a cause–effect structure, where a behavior is linked to an expected consequence. However, investigative hypotheses may also appear as simpler statements, questions, event descriptions, relational claims, or combinations of several conditions. Future work should therefore test hypotheses with different linguistic structures and different themes. This would make it possible to better understand how hypothesis structure influences query expansion, retrieval method selection, and inference quality.

Future work should also evaluate the pipeline on other types of investigative documents. Although this thesis focuses on email communication, forensic investigations may involve chat logs, reports, attachments, transcripts, memos, or mixed document collections. These sources may require different preprocessing, chunking, metadata handling, and context selection strategies. Testing the system beyond emails would show whether the proposed hypothesis-driven RAG approach can generalize to broader forensic document analysis.

Finally, future work should study the effect of system parameters that were kept fixed in this thesis, such as the embedding model, the number of expanded queries, the query expansion strategy, the inference model, and the retrieval

budget. A systematic optimization study could clarify which parameters have the strongest effect on retrieval and inference performance, and whether different hypotheses require different parameter settings.

Appendix A

Prompts

A.1 Sparse Email Cleaning Prompt

The following prompt instructs the LLM to clean a raw email thread so that the resulting text is suitable for sparse retrieval.

```
You are an expert in forensic email extraction. Clean the following
↪ email thread for sparse retrieval (BM25, keyword search).

GOALS:
- Extract the body of each individual email in the thread.
- Preserve all original wording and factual content.
- Remove all metadata and noise.

REQUIREMENTS:
1. Identify each email in the thread.
2. For each email, output ONLY the cleaned body text.
3. Remove:
   - All metadata (Date, From, To, CC, Subject, Message-ID, routing
   ↪ headers)
   - Signatures (e.g., "Regards", "Cheers", name blocks)
   - Legal disclaimers
   - Duplicate quoted text from earlier emails
   - Formatting artifacts, broken lines, stray characters
4. Keep:
   - All original sentences and wording in the body
   - All financial, legal, contractual, and operational details
   - All entities, numbers, obligations, and instructions
5. Restore natural paragraph formatting:
```

- merge wrapped lines within the same paragraph
- preserve paragraph breaks
- preserve bullet lists
- keep each bullet on one line
- do not merge bullets together

6. Do NOT:
- Paraphrase
- Summarize
- Rewrite sentences
- Compress content

7. Separate each cleaned email body with the delimiter:

<--- block --->

OUTPUT FORMAT (for a thread with 2 emails):
<body of email 1>
<--- block --->
<body of email 2>

A.2 Dense Email Cleaning Prompt

The following prompt instructs the LLM to clean a sparse-cleaned email thread so that the resulting text is suitable for dense retrieval.

You are transforming sparse-cleaned email into a dense-retrieval version.

Task:
Normalize the content for dense retrieval while preserving the structure.

Input:
The input contains body-only email already cleaned and separated.

Entity anonymization:
- anonymize person names as [PERSON]
- replace phone/fax numbers with [PHONE]
- replace email addresses and URLs with [URL]
- replace passcodes/access codes with [PASSCODE]
- replace dates/time/week-day/day-time expressions with [DATE]
- replace monetary values with [AMOUNT]

- replace percentage values with [PERCENTAGE]
- replace standalone sign-offs (e.g., thanks, best, regards) with
↪ [REGARD]

Output rules:
- no explanation
- no summary
- no paraphrasing
- no rewriting beyond anonymization rules
- no touching of the separators "<--- block --->" which are between
↪ blocks
- no changing of structure of sentences and bullet points
- return only the anonymized version

A.3 Sparse Query Expansion Prompt

The following prompt instructs the LLM to expand a hypothesis into a set of suitable sparse retrieval queries.

You are assisting with forensic hypothesis-based retrieval over a mixed
↪ corporate email/document corpus.

Given a forensic hypothesis, generate sparse retrieval queries for BM25
↪ or keyword search.

Goal:
Create short lexical queries that preserve the key terms, actions,
↪ entities, and domain-specific concepts in the hypothesis.

Rules:

* Generate exactly {num_queries} queries.
* Each query must be a short keyword-style phrase, not a full sentence.
* Maximum 6 words per query.
* Do not write first-person statements.
* Do not simulate what employees might say.
* Preserve important terminology from the hypothesis.
* Include close lexical variants, abbreviations, and domain synonyms
↪ where appropriate.
* Focus on concrete nouns, verbs, entities, objects, processes, and
↪ outcomes.

```
* Avoid vague phrases such as ”handle this carefully” or ”align on
↪ messaging” unless those terms are central to the hypothesis.
* Do not introduce mechanisms, motives, actors, or events that are not
↪ present in the hypothesis.
* Avoid duplicates and near-duplicates.
* Return valid Python list only.

Now generate queries for this hypothesis:
{hypothesis}
```

A.4 Dense Query Expansion Prompt

The following prompt instructs the LLM to expand a hypothesis into a set of suitable dense retrieval queries.

```
You are assisting with forensic hypothesis-based retrieval over a mixed
↪ corporate email/document corpus.

Given a forensic hypothesis, generate dense retrieval queries for
↪ semantic search.

Goal:
Create neutral, third-person, evidence-oriented queries that preserve
↪ the meaning of the hypothesis without simulating confessions or
↪ internal dialogue.

Rules:

* Generate exactly {num_queries} queries.
* Each query should be one concise natural-language sentence.
* Do not write first-person statements.
* Do not use phrases such as ”we should”, ”can we”, ”let's”, ”I think”,
↪ or ”please confirm”.
* Do not simulate what employees might write.
* Do not generate admissions of wrongdoing.
* Preserve the core meaning, actions, objects, and domain terminology
↪ of the hypothesis.
* Use neutral evidence-oriented wording such as ”communications
↪ about”, ”documents discussing”, ”internal discussion concerning”,
↪ ”records related to”, or ”analysis of”.
```

```
* Include semantic variations, but do not add mechanisms, motives,
↪ actors, or events not present in the hypothesis.
* Do not make the hypothesis more accusatory than the original
↪ wording.
* Include queries broad enough to match emails, forwarded articles,
↪ legal materials, regulatory discussions, meeting notes, summaries,
↪ or operational reports.
* Avoid vague business language that weakens the investigative
↪ meaning.
* Avoid over-specific wording that depends on knowing the target
↪ documents.
* Avoid duplicates and near-duplicates.
* Return valid Python list only.

Now generate queries for this hypothesis:
{hypothesis}
```

A.5 Context Analysis Prompt

The following prompt instructs the LLM to analyze a retrieved context and produce a structured inference output, including a relevance label, referred to as *strength* in the prompt, a list of evidence spans, and a reasoning explanation based on the context's relevance to the given hypothesis.

```
You are assisting with forensic evidence analysis.

Definitions:
- A forensic hypothesis: An assertion about a potential wrongdoing or
↪ misconduct that is being investigated.
- An email candidate: An email chosen for evidence analysis and
↪ evaluating the hypothesis.

Email candidate structure:
- The email candidate is represented as chunks of text extracted from
↪ that email.
- The chunks are separated by the delimiter <--chunk-->.
```

```
############### Analysis Procedure
↪ ##################

- Hypothesis analysis — A forensic hypothesis typically describes two
↪ parts:
  1. Cause: one or more actions, behaviors, decisions, representations,
  ↪ transactions, communications, or omissions.
  2. Effect: one or more intended, actual, or potential outcomes,
  ↪ effects, motivations, or consequences.

- Email analysis for evidence — The email may include direct or
↪ indirect evidence regarding:
  * The Cause.
  * The Effect.
  * The mechanisms, structures, processes, approvals, accounting
  ↪ treatments, financial effects, legal arrangements, controls, or
  ↪ operational activities that could connect the Cause and Effect
  ↪ described in the hypothesis.
  * Evidence that supports, contradicts, explains, contextualizes, or
  ↪ otherwise helps evaluate the hypothesis.

- Extract evidence spans exactly as they appear in the email. Do not
↪ paraphrase evidence spans.

- Reason: Explain how the extracted evidence helps evaluate the
↪ hypothesis, or how it relates only to the cause, only to the effect,
↪ or to the connection between cause and effect. If there is no
↪ relevant evidence, briefly explain why the email is not useful for
↪ evaluating the hypothesis.

- Strength: Assign the strength of the email's relevance to the
↪ hypothesis as one of:
  * low
  * medium
  * high

Use "low" when the email is weak, vague, or only provides very minor
↪ context.
Use "medium" when the email mentions, directly or indirectly, only a
↪ part of the hypothesis, like only the Cause, or only the Effect.
Use "high" when the email mentions, directly or indirectly, both the
↪ Cause and the Effect.
```

*** Crucial point: The email does not need to use the exact words
↪ appearing in the hypothesis. Consider the semantic meaning of
↪ the email and whether it provides evidence relevant to assessing
↪ the cause, the effect, or both.

################ Hypothesis analysis examples
↪ ##################

Example 1:
Hypothesis: "Employees discussed adjusting, restricting, or reallocating
↪ energy schedules, bids, or load volumes in ways that could affect
↪ supply availability or market prices."

Cause: "Adjusting, restricting, or reallocating energy schedules, bids,
↪ or load volumes."
Effect: "Affecting supply availability or market prices."

Example 2:
Hypothesis: "Employees described prepay transactions as loans or
↪ financing in order to keep associated debt off the balance sheet or
↪ to improve the company's reported financial position."

Cause: "Describing, treating, or considering prepays as income, loans,
↪ or financing."
Effect: "Keeping debt related to prepays off the balance sheet or
↪ improving the company's reported financial position."

For both examples, phrases, sentences, or statements with similar
↪ semantic meaning to the cause or effect can be considered
↪ evidence.

################ Output construction
↪ ##################

Return valid JSON only.

Output format:
{{
"evidence_spans": ["exact quoted span extracted from the email"],

”reason”: ”brief explanation of how the evidence relates to the hypothesis”,
”strength”: ”low | medium | high”
}}

If the email contains no relevant evidence, return:
{{
”evidence_spans”: [],
”reason”: ”brief explanation why the email is not useful for evaluating the hypothesis”,
”strength”: ”low”
}}

Now perform the task for the following input:

Hypothesis:
{hypothesis}

Email:
{email}

Bibliography

[1] M. Hilbert and P. López, "The World's Technological Capacity to Store, Communicate, and Compute Information," *Science*, vol. 332, no. 6025, pp. 60–65, 2011. Available: https://doi.org/10.1126/science.1200970.

[2] The Radicati Group, *Email Statistics Report, 2024–2028*. The Radicati Group, Inc., 2024. Available: https://www.radicati.com.

[3] E. Casey, *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet*, 3rd ed. Academic Press, 2011.

[4] U.S. Department of Justice, *Amerithrax Investigative Summary*, 2010. Available: https://www.justice.gov/archive/amerithrax/docs/amx-investigative-summary.pdf.

[5] N. L. Beebe and J. G. Clark, "A Hierarchical, Objectives-Based Framework for the Digital Investigations Process," *Digital Investigation*, vol. 2, no. 2, pp. 147–167, 2005. Available: https://doi.org/10.1016/j.diin.2005.04.002.

[6] Y. Zhu, H. Yuan, S. Wang, J. Liu, W. Liu, C. Deng, H. Chen, Z. Liu, Z. Dou, and J.-R. Wen, "Large Language Models for Information Retrieval: A Survey," *arXiv preprint arXiv:2308.07107*, 2023. Available: https://arxiv.org/abs/2308.07107.

[7] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008. Available: https://nlp.stanford.edu/IR-book/.

[8] J. Felser, D. Labudde, and M. Spranger, "Towards Hypothesis-driven Forensic Text Exploration System," in *Proceedings of the 2023*

IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications, Valencia, Spain, pp. 42–47, 2023. Available: https://www.thinkmind.org/articles/iaria_congress_2023_1_100_50098.pdf

[9] K. Lin, K. Lo, J. E. Gonzalez, and D. Klein, "Decomposing Complex Queries for Tip-of-the-Tongue Retrieval," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, Singapore, pp. 5521–5533, 2023. doi: 10.18653/v1/2023.findings-emnlp.367.

[10] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-Augmented Generation for Large Language Models: A Survey," *arXiv preprint arXiv:2312.10997*, 2023. Available: https://arxiv.org/abs/2312.10997.

[11] D. Quick and K.-K. R. Choo, "Impacts of Increasing Volume of Digital Forensic Data: A Survey and Future Research Challenges," *Digital Investigation*, vol. 11, no. 4, pp. 273–294, 2014. Available: https://doi.org/10.1016/j.diin.2014.09.002.

[12] T. Breuer, S. Frihat, N. Fuhr, D. Lewandowski, P. Schaer, and R. Schenkel, "Large Language Models for Information Retrieval: Challenges and Chances," *Datenbank-Spektrum*, vol. 25, pp. 71–81, 2025. Available: https://doi.org/10.1007/s13222-025-00503-x.

[13] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. Available: https://arxiv.org/abs/2005.11401.

[14] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, J. Jiang, and B. Cui, "Retrieval-Augmented Generation for AI-Generated Content: A Survey," *arXiv preprint arXiv:2402.19473*, 2024. Available: https://arxiv.org/abs/2402.19473.

[15] S. Gupta, R. Ranjan, and S. N. Singh, "A Comprehensive Survey of Retrieval-Augmented Generation Systems," *arXiv preprint arXiv:2410.12837*, 2024. Available: https://arxiv.org/abs/2410.12837.

[16] Y. Huang and J. Huang, "A Survey on Retrieval-Augmented Text Generation for Large Language Models," *arXiv preprint arXiv:2404.10981*, 2024. Available: https://arxiv.org/abs/2404.10981.

[17] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009. Available: https://doi.org/10.1561/1500000019.

[18] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense Passage Retrieval for Open-Domain Question Answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020. Available: https://aclanthology.org/2020.emnlp-main.550/.

[19] G. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 874–880, 2021. Available: https://aclanthology.org/2021.eacl-main.74/.

[20] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: A Large-Scale Dataset for Fact Extraction and Verification," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 809–819, 2018. Available: https://aclanthology.org/N18-1074/.

[21] G. V. Cormack, M. R. Grossman, B. Hedin, and D. W. Oard, "Overview of the TREC 2010 Legal Track," in *Proceedings of the Nineteenth Text REtrieval Conference (TREC 2010)*, 2010. Available: https://trec.nist.gov/pubs/trec19/papers/LEGAL10.OVERVIEW.pdf.

[22] M. R. Grossman and G. V. Cormack, "Technology-Assisted Review in E-Discovery Can Be More Effective and More Efficient Than Exhaustive Manual Review," *Richmond Journal of Law and Technology*, vol. 17, no. 3, article 11, 2011. Available: https://scholarship.richmond.edu/jolt/vol17/iss3/5.

[23] N. L. Beebe and J. G. Clark, "Digital Forensic Text String Searching: Improving Information Retrieval Effectiveness by Thematically Clustering Search Results," *Digital Investigation*, vol. 4, suppl., pp. S49–S54, 2007. Available: https://doi.org/10.1016/j.diin.2007.06.005.

[24] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019. Available: https://doi.org/10.18653/v1/D19-1410.

[25] G. V. Cormack, C. L. A. Clarke, and S. Buettcher, "Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods," in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*, pp. 758–759, 2009. Available: https://doi.org/10.1145/1571941.1572114.

[26] J. J. Rocchio Jr., "Relevance Feedback in Information Retrieval," in *The SMART Retrieval System: Experiments in Automatic Document Processing*, G. Salton, Ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1971, pp. 313–323. Available: https://sigir.org/files/museum/pub-08/XXIII-1.pdf

[27] V. Lavrenko and W. B. Croft, "Relevance-Based Language Models," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New Orleans, LA, USA, 2001, pp. 120–127. doi: 10.1145/383952.383972.

[28] L. Wang, N. Yang, and F. Wei, "Query2doc: Query Expansion with Large Language Models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9414–9423, 2023. Available: https://doi.org/10.18653/v1/2023.emnlp-main.585.

[29] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise Zero-Shot Dense Retrieval without Relevance Labels," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1762–1777, 2023. Available: https://doi.org/10.18653/v1/2023.acl-long.99.

[30] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "Retrieval Augmented Language Model Pre-Training," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, *Proceedings of Machine Learning Research*, vol. 119, pp. 3929–3938, 2020. Available: https://proceedings.mlr.press/v119/guu20a.html.

[31] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," in *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024. Available: https://arxiv.org/abs/2310.11511.

[32] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling, "Corrective Retrieval Augmented Generation," *arXiv preprint arXiv:2401.15884*, 2024. Available: https://doi.org/10.48550/arXiv.2401.15884.

[33] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAs: Automated Evaluation of Retrieval Augmented Generation," in *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 150–158, 2024. Available: https://aclanthology.org/2024.eacl-demo.16/.

[34] M. R. Grossman, G. V. Cormack, and A. Roegiest, "TREC 2016 Total Recall Track Overview," in *Proceedings of the Twenty-Fifth Text REtrieval Conference (TREC 2016)*, NIST Special Publication 500-321, 2016. Available: https://trec.nist.gov/pubs/trec25/papers/Overview-TR.pdf.

[35] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. Available: https://openreview.net/forum?id=wCu6T5xFjeJ.

[36] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, "MS MARCO: A Human Generated MAchine Reading COmprehension Dataset," in *Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches (CoCo@NIPS)*, 2016. Available: https://arxiv.org/abs/1611.09268.

[37] N. Guha et al., "LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models," in *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*, pp. 44123–44279, 2023. Available: https://arxiv.org/abs/2308.11462.

[38] A. Salemi and H. Zamani, "Evaluating Retrieval Quality in Retrieval-Augmented Generation," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2395–2400, 2024. Available: https://doi.org/10.1145/3626772.3657957.

[39] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore, pp. 2511–2522, 2023. doi: 10.18653/v1/2023.emnlp-main.153.

[40] Text Retrieval Conference, "Text Retrieval Conference (TREC)." National Institute of Standards and Technology (NIST). Available: https://trec.nist.gov/.

[41] TREC Legal Track, "TREC Legal Track Official Site." Available: https://trec-legal.umiacs.umd.edu/.

[42] Carnegie Mellon University, "Enron Email Dataset." Available: https://www.cs.cmu.edu/~enron/.

[43] EDRM, "EDRM and ZL Technologies Launch New Enron Email Data Set," 2010. Available: https://edrm.net/2010/11/edrm-and-zl-technologies-launch-new-enron-email-data-set/.

[44] TREC Legal Track, "TREC 2010 Legal Track – Document Collection (EDRM Enron Dataset v2 Helpers)." Available: https://trec-legal.umiacs.umd.edu/corpora/trec/legal10/.

[45] R. F. McCullough, "Reports on Enron Trading Strategies During the California Energy Crisis," McCullough Research, 2002–2003. Available: https://www.mresearch.com/pdfs/254.pdf.

[46] University of California, Berkeley, "Enron Email Analysis Project." Available: https://bailando.berkeley.edu/enron_email.html.

[47] B. Hedin, S. Tomlinson, J. R. Baron, and D. W. Oard, "Overview of the TREC 2009 Legal Track," in *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*, 2009. Available: https://trec.nist.gov/pubs/trec18/papers/LEGAL09.OVERVIEW.pdf.

[48] L. Weidinger et al., "Taxonomy of Risks Posed by Language Models," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pp. 214–229, 2022. Available: https://doi.org/10.1145/3531146.3533088.

[49] Ollama, "Ollama: Run Large Language Models Locally." Available: https://ollama.com/.

[50] Google, "Gemma 4 model overview." *Google AI for Developers*, 2026. Available: https://ai.google.dev/gemma/docs/core. Accessed: July 5, 2026.

[51] Google, "google/gemma-4-31B-it." *Hugging Face Model Card*, 2026. Available: https://huggingface.co/google/gemma-4-31B-it. Accessed: July 5, 2026.

[52] C. Cheerla, "Advancing Retrieval-Augmented Generation for Structured Enterprise and Internal Data," *arXiv preprint arXiv:2507.12425*, 2025. Available: https://arxiv.org/abs/2507.12425.

[53] H. Yu, C. Xiong, and J. Callan, "Improving Query Representations for Dense Retrieval with Pseudo Relevance Feedback," in *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, Virtual Event, Queensland, Australia, 2021, pp. 3592–3596. doi: 10.1145/3459637.3482124.

[54] Y. Lei, Y. Cao, T. Zhou, T. Shen, and A. Yates, "Corpus-Steered Query Expansion with Large Language Models," in *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, St. Julian's, Malta, 2024, pp. 393–401. doi: 10.18653/v1/2024.eacl-short.34.

[55] D. Otero and J. Parapar, "LLM-Assisted Pseudo-Relevance Feedback," in *Advances in Information Retrieval*, 2026, pp. 452–459. doi: 10.1007/978-3-032-21300-6_36.

[56] R. Nogueira and K. Cho, "Passage Re-ranking with BERT," *arXiv preprint arXiv:1901.04085*, 2019. doi: 10.48550/arXiv.1901.04085.

[57] O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Virtual Event, China, 2020, pp. 39–48. doi: 10.1145/3397271.3401075.

[58] CLAUDETTE Project, "CLAUDETTE Corpora." Available: http://claudette.eui.eu/corpora/index.html. Accessed: July 4, 2026.

Acknowledgements

I would like to express my gratitude to my supervisor, Prof. Paolo Torroni, for his guidance and valuable feedback throughout the development of this thesis.

I am equally grateful to my co-supervisor, Prof. Michael Spranger, for giving me the opportunity to conduct this research at Hochschule Mittweida and for his insightful discussions and support during the different stages of the project.

I would also like to thank Jenny Felser for her kind assistance and responsiveness regarding the practical and organizational aspects of my work at Hochschule Mittweida.

Finally, I would like to thank my parents, Susan and Keramat, and my sister, Saba, for their unconditional support, their confidence in me, and for always standing by the choices I have made throughout my life. Their encouragement has been a constant source of strength and motivation, helping me move forward at every stage of my journey.