UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Leveraging Large Language Models for Document Classification

Rómulo Brandão Nogueira

Mestrado em Engenharia Informática

Trabalho Projeto orientado por:
Prof. Doutor Nuno da Cruz Garcia
Prof. Hugo Mentzingen Silva

2025

Acknowledgments

As in every aspect of life, I believe that each stage is an opportunity to grow. With this spirit I embraced this project and, in the same spirit, I would like to express my gratitude to all those who made this journey possible, namely:

To my parents, for encouraging me to embrace this adventure and for providing me with the conditions necessary to study, grow, and dream.

To my girlfriend, for her love, patience, support, and constant motivation throughout this journey.

To my sisters, for the moments of laughter and joy that helped maintain my balance along the way.

To the rest of my family, who, whether near or far, have always been a safe harbor and have cheered for my success.

To my friends, who, even indirectly, contributed greatly to making this possible by cheering me on and creating moments of fun, relief, and happiness.

To all my colleagues at Deloitte, for their motivation, shared ideas, and support during different stages of this process.

To my supervisor, Professor Doctor Nuno Cruz Garcia, who has accompanied me since the beginning of my academic journey, for his guidance, support, and trust.

And finally, to my co-supervisor Hugo Mentzingen, who not only shared this academic journey with me but has also been a fundamental guide in my professional growth.

Without each of you, this path would not have been the same. To all of you, my deepest thanks!

Acknowledgments.

Resumo

O crescimento exponencial do volume de documentos gerados diariamente trouxe consigo desafios expressivos em diversas áreas como segurança, organização, análise e gestão da informação. A diversidade intrínseca destes documentos, que podem variar em formato, extensão, estrutura ou origem, desde ficheiros digitais até manuscritos digitalizados de baixa qualidade, acentua ainda mais a complexidade. Esta heterogeneidade é frequentemente acompanhada de sobreposições semânticas entre diferentes classes, onde conteúdos semelhantes podem assumir finalidades distintas, aumentando o risco de erro na interpretação e dificultando processos de automação.

No setor bancário, estas dificuldades assumem particular relevância, uma vez que, uma parte significativa das operações diárias depende do tratamento de documentos não estruturados, como contratos, extratos, faturas ou formulários. Muitos destes documentos apresentam características adversas, tornando a automatização complexa e obriga a um elevado grau de intervenção humana. A dependência de validações manuais, compromete a escalabilidade dos sistemas atuais e limita a possibilidade de alocar recursos a tarefas de maior valor acrescentado para as empresas. Neste contexto, a capacidade de identificar de forma automática a tipologia, natureza e finalidade dos documentos torna-se crítica para suportar a automação de operações subsequentes, como extração de dados, encaminhamento em fluxos de validação ou armazenamento estruturado.

A classificação de documentos tem sido objeto de interesse científico desde a década de 1960, estando em constante evolução e acompanhando diretamente os desenvolvimentos da Inteligência Artificial (IA). As raízes da IA remontam aos anos de 1950 e 1960, enquanto esta ainda se afirmava como disciplina, através de trabalhos que marcam a sua história como o artigo de Turing [1] e o projeto de Dartmouth [2]. Uma das primeiras propostas para classificação automática de documentos, ainda que rudimentar, representou um marco essencial, uma vez que determinou a viabilidade da mesma iniciando um campo de investigação promissor [3].

Ao longo das décadas seguintes, a evolução da IA passou por diferentes fases. Períodos de otimismo foram marcados pelo aparecimento de algoritmos como Term Frequency-Inverse Document Frequency (TF.IDF) [4, 5], sistemas especialistas [6, 7, 8], modelos baseados em aprendizagem automática (K-Vizinhos Mais Próximos [9, 10], Naïve Bayes [11, 12] e Árvores de Decisão [13, 14]) e também abordagens mais avançadas como as redes neuronais [15, 16, 17, 18]. Em contrapartida, também existiram dois períodos onde os desenvolvimentos estagnaram, conhecidos como os invernos de IA. Estes foram originados por motivos relacionados com "conhecimento", nomeadamente a ausência deste [19, 20] e, após o aparecimento dos sistemas especialistas, a dificuldade em adquiri-lo e formalizá-lo [6, 7, 8, 21].

A introdução da arquitetura Transformer em 2017 [22] constituiu um momento decisivo no processamento de linguagem natural (PLN). O mecanismo de atenção permitiu modelar relações de longo alcance e capturar contexto de forma muito mais eficaz do que abordagens anteriores. Assim, esta inovação abriu caminho para o surgimento dos modelos de linguagem de larga escala (LLMs), cuja capacidade de generalização e de representação semântica tornou-os particularmente adequados para tarefas de PLN, como a classificação de documentos [23, 24]. Desta forma, os LLMs distinguem-se dos métodos clássicos pela capacidade de compreensão de relações semânticas e estruturais, traduzindo-se em ganhos expressivos de precisão na interpretação de texto [22, 25].

O sucesso dos LLMs não está apenas relacionado com o interesse da comunidade científica [25], mas também com a rápida adesão na indústria e no quotidiano de tanto utilizadores técnicos como não técnicos. Sendo exemplo o lançamento do ChatGPT em 2022, que alcançou o primeiro milhão de utilizadores em cinco dias, tornando-se a segunda aplicação mais rápida de sempre a atingir este feito [26].

No que concerne a classificação automática de documentos, a utilização de LLMs tem se mostrado promissora com diversas soluções que apesar de assinalarem progressos na área quando comparados com os métodos tradicionais, o seu estado atual ainda não corresponde plenamente ao que seria desejável.

Deste modo, e com objetivo de colmatar as lacunas ainda presentes, a *framework* proposta explora um paradigma de aprendizagem "*zero-shot*"de modo a reduzir a dependência de conjuntos de dados pré-selecionados. Para além disto, integra fluxos de trabalho configuráveis e adota estratégias de recuperação seletiva de contexto, visando a otimização e eficiência de custos, sem penalizar o desempenho. Ao combinar todas estas vantagens num só sistema, a *framework* estabelece uma base robusta, escalável e prática para a classificação automática de documentos num mundo moderno.

A *framework* pode ser decomposta em quatro macro componentes. A primeira diz respeito à configuração, onde o utilizador define os rótulos de classificação, através da seleção de um nome único e uma breve descrição. Esta parametrização mínima e em linguagem natural, possibilita que mesmo perfis não técnicos possam usufruir da *framework*, tornando-a flexível e facilmente escalável. Segue-se a ingestão, onde todo o conteúdo do documento é extraído. No que concerne a seleção de contexto, a *framework* foi projetada para ser altamente flexível, permitindo a utilização de diferentes métodos. Atualmente, existem quatro abordagens implementadas:

- **Estratégia 1 – Primeiras Páginas**: reflete uma abordagem natural humana, onde são examinadas as primeiras N páginas, e por isso, operando sob a suposição de que as informações mais relevantes estão posicionadas no início do documento;

- **Estratégia 2 - Primeiras e Últimas Páginas**: amplia a abordagem anterior, incorporando não só o conteúdo inicial (primeiras N páginas) como também do final do documento (últimas M páginas). A suposição subjacente é que, enquanto as secções introdutórias for-

necem um contexto essencial, as secções conclusivas contêm frequentemente resumos, resultados ou observações finais que podem ser igualmente informativos para a classificação;

- **Estratégia 3 - Vetorização do texto completo**: utiliza uma versão adaptada do algoritmo TF.IDF, onde o documento de input é processado como uma coleção de páginas e o objetivo é, através do uso deste algoritmo, avaliar a importância dos termos em cada página em relação às outras páginas dentro do mesmo documento. Consequentemente, permitindo identificar as páginas mais informativas de um documento com base apenas na importância dos termos;

- **Estratégia 4 – Geração Aumentada por Recuperação (RAG)**: utiliza a técnica de recuperação semântica baseada em *embeddings*, onde tanto o documento de entrada como os rótulos acompanhados das suas descrições são projetados num espaço vetorial comum. O objetivo é medir a similaridade entre esses vetores e, assim, identificar quais as partes de texto que estão mais próximos dos rótulos. Consequentemente, este processo garante que o modelo funcione num contexto mais relevante e controlado.

Nesta fase, a *framework* reúne todos os elementos necessários para proceder à classificação. Deste modo, é construída uma *prompt* estruturada em XML, que inclui os dados contextuais, isto é, pares rótulo-descrição e páginas recuperadas. De seguida, através da *prompt* do sistema é atribuida uma *persona* ao modelo e são definidas diretrizes gerais de decisão. A saída é regulada por um modelo *Pydantic*, que exige a existência de dois campos na resposta do modelo: o "*rationale*", onde o modelo expõe o raciocínio que suporta a decisão, e "classificação", onde atribui o rótulo mais adequado. Adicionalmente, um mecanismo de segurança foi implementado de modo a controlar o tamanho da *prompt*, truncando o contexto caso ultrapasse a janela de *tokens*. Finalmente, o LLM é invocado e devolvendo a classificação e respetiva justificação.

Para avaliar a *framework*, foi criado um conjunto de dados de documentos em bruto orientados ao setor bancário a partir de documentos públicos. Com base neste conjunto de dados, definiram-se três casos de uso: validação documental para decisão de crédito, formalização de operações bancárias e gestão de seguros e garantias. Os testes foram conduzidos através de numa *pipeline* automática, desenhada para mitigar a natureza probabilística dos LLMs, onde cada documento foi processado duas vezes e, se houvesse inconsistências, o teste repetia-se até cinco vezes adicionais prevalecendo o resultado mais frequente.

A análise dos três subconjuntos de dados revelou padrões consistentes. Em termos da comparação de modelos, o *GPT-4o* destacou-se claramente como o modelo mais robusto, com um impacto mais reduzido da estratégia de recuperação, mas com custos operacionais elevados. O *GPT-4o-mini* apresentou-se como a opção com melhor equilíbrio entre desempenho e custo, alcançando resultados próximos do *GPT-4o* a uma fração do preço. Já o *GPT-3.5-turbo* mostrou-se sistematicamente inferior, embora ocasionalmente tenha produzido resultados aceitáveis, nunca se revelou competitivo face a alternativas mais fiáveis e económicas.

No que respeita às estratégias da framework, o *RAG* demonstrou maior consistência entre

modelos e subconjuntos de dados. Apesar de não ser a solução mais económica, evidenciou o melhor compromisso entre custo e desempenho, sobretudo em casos de elevada similaridade. As abordagens *Primeiras Páginas* e *Primeiras e Últimas Páginas* posicionaram-se como opções intermédias, sendo estas mais económicas, mas sem ganhos expressivos de precisão. Já o *TF.IDF* destacou-se negativamente, revelando o pior desempenho aliado a custos elevados.

Das bases de comparação testadas, só a chamada direta ao LLM conseguiu obter resultados competitivos face à *framework*, sobretudo nos modelos de 4.ª geração. Ainda assim, quando esta abordagem utilizou o modelo mais pequeno (*GPT-3.5-turbo*) ficou aquém. Em todos os casos, esta estratégia implicou custos elevados e pouco controláveis, bastante superiores aos da *framework* e sem ganhos em termos de taxa de acerto.

Os casos de estudo em crédito, conformidade legal e seguros confirmaram a capacidade de generalização da *framework*. Em todos os domínios, assegurou desempenho superior, eficiência de custos e escalabilidade, sublinhando o seu valor como solução prática e pronta para adoção em cenários reais.

Palavras-chave: Classificação de documentos; Modelos de Linguagem de Larga Escala; Setor bancário; Classificador Zero-Shot; Geração aumentada por recuperação.

Abstract

Document classification serves as foundational step in critical tasks such as information extraction, analysis and decision-making. However, existing approaches often struggle with the variability, volume, and complexity of real-world documents. These methods are further limited by a lack of configurability and explainability, requiring specialized technical expertise to accommodate diverse user needs and often producing results that are difficult to interpret. To address the complexities of modern document processing, this dissertation introduces a novel zero-shot document classification framework that leverages Large Language Models (LLMs), designed for accessibility and configurability by both technical and non-technical users. Unlike traditional methods, which require extensive labeled data, the zero-shot configuration enables the framework to perform the classification task without any prior exposure to labeled examples of the target categories, relying instead on semantic understanding derived from user-provided label descriptions and document content. To validate the proposed framework, a dataset tailored to the banking sector was constructed, bringing together documents of different types and sizes. Based on this corpus, three distinct use cases were defined, designed to assess the practical usefulness of the framework in different scenarios. The subsets were further explored through evaluations of different retrieval strategies and through comparisons with competing zero-shot approaches whether using LLMs or not, providing a broader perspective on the framework's effectiveness. Experimental results show that the framework achieves higher accuracy while requiring fewer tokens, which directly translates into lower operating costs compared to the baseline, pointing toward a gradual refinement in current document classification practices.

Keywords: Document Classification; Large Language Models; Banking Sector; Zero-Shot Classifier; Retrieval-Augmented Generation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The growing volume of documents generated on a daily-basis raises significant challenges across various areas, such as security, organization, analysis, and information management. These challenges are further compounded by the intrinsic diversity of the documents, which can vary in size, format, structure, origin, and nature, whether they are digital, scanned, handwritten, or printed. This complexity is further aggravated by the frequent semantic overlap between documents from different classes which, despite having similar content, they differ substantially in objective, context, or purpose. These challenges are particularly pronounced in the banking sector, where many essential operations relies on the processing of unstructured documents, such as contracts, statements, invoices, and forms, that sometimes are handwritten and of poor quality. As a result, human intervention is inevitable in many cases, making automation difficult, thus, limiting the ability to allocate human resources to higher value-added tasks. In this context, automatic classification of documents acquires particular relevance, being a technically demanding and strategically attractive challenge, due to its significant implications for operational efficiency and daily workflows.

The ability to correctly identify the typology, nature, and purpose of each document without manual intervention is essential for enabling subsequent operations like data extraction, routing to validation workflows, or structured storage. Thus, automated classification serves as a cornerstone in the development of intelligent, scalable, and efficient document management systems.

Beyond its practical usefulness, the topic of document classification has been of scientific interest since the 1960s, constantly evolving as new technologies emerge. Despite this progress, traditional Artificial Intelligence (AI) solutions typically require large volumes of labeled data to achieve effective performance [27], which limits their applicability in real-world scenarios where such data is scarce or costly to obtain. Simultaneously, as AI is increasingly reshaping task execution across industries, it becomes essential that modern systems prioritize accessibility, scalability, and configurability. However, many current systems still require specialized knowledge to be trained, configured, or adapted, making their adoption difficult in dynamic or regulated contexts, such as the banking sector.

In order to fully leverage the transformative potential of AI, organizations must adopt solu-

tions that empower not only technical experts but also non-technical user to meaningfully engage with and tailor these tools, thus empowering all their teams, without compromising and ideally surpassing current levels of quality and performance.

Motivated by the limitations identified in existing approaches and the current state of the art, the present dissertation was developed in collaboration with the company *Deloitte Technology*, as part of a large-scale project implemented at one of the leading banks in Portugal, where document classification assumes a critical role in an automatic document validation process. This project offered a practical and realistic setting for the research, enabling the application, testing, and refinement of the proposed concepts within an environment characterized by high technical and operational demands. The adopted methodology followed a classical research and development (R&D) approach, culminating in the design and implementation of a framework adapted both to the specificities of the bank's technological infrastructure and its functional needs, thus ensuring the relevance and applicability of the results obtained.

1.2 Objectives

The main objective of this dissertation is to explore the potential of LLMs in the task of automatic document classification and further design and develop a configurable, flexible zero-shot solution based on these models capable of delivering high performance.

For this purpose, the investigation begins with a review of the state of the art, in which the historical evolution of document classification is traced, allowing for the identification of its main challenges and limitations. This review culminates in a critical analysis of the traditional methodologies employed in the field, in order to highlight weaknesses and identifying opportunities for improvements. Also, a characterization of LLMs is presented as an emerging technology within the domain of Natural Language Processing (NLP), addressing its architecture, capabilities, and applicability to complex NLP tasks. The review concludes with the analysis of recent approaches that already integrate LLMs into document classification pipelines, highlighting the main advances and remaining limitations.

Based on this theoretical foundation, the design and development of a zero-shot, domain-agnostic, and user-configurable framework is proposed. The solution aims to combine the performance of LLMs with practical usability in real-world, demanding environments, such as the banking sector, while dynamically adapting to evolving requirements.

In order to validate the proposed framework, several experiments are conducted using a real-world dataset from the banking sector. These experiments assess the effectiveness of the framework in concrete scenarios, exploring different context retrieval strategies and comparing results with other zero-shot classifiers. The objective is to demonstrate, based on empirical evidence, the practical advantages of adopting the proposed framework in terms of flexibility, performance, and applicability in an environment like banking.

1.3 Contributions

This dissertation tackles the current challenges of automatic document classification through the following main contributions:

- Provide a comprehensive study and critical analysis of the document classification evolution, identifying current limitations and highlighting the motivation behind the adoption of LLMs for this task.
- Create a dataset in order to address the lack of public banking-sector corpora by assembling a small yet comprehensive collection of real-world financial documents that capture the sector's diversity and complexity.
- A novel user-centric zero-shot classification framework built on LLMs capable of processing large unstructured documents while effectively addressing the variability, structural differences, and content similarities common in banking documents, with possible extension to other domains.
- Investigating and assessing the impact of various approaches in providing relevant context to the LLM responsible for selecting the right label.
- Benchmarking the proposed framework against other zero-shot classifiers, whether LLM-based approaches or not.

The proposed zero-shot approach fundamentally enhances accessibility and configurability for end users by reducing the complexity to a single and intuitive step: defining classification labels and their descriptions (section 4.1) entirely in natural language. In a zero-shot setting, the model is able to perform classification tasks without any prior exposure to labeled examples from the target categories, relying instead on the semantic information provided. This straightforward parameterization process allows users, regardless of their background, to tailor the system around their needs without requiring annotated training data or complex setup. Moreover, the framework goes beyond a simple classification by providing explanations for its decisions, thereby contributing to greater transparency and explainability, which is an essential aspect for trustworthy document intelligence. As a result, the framework offers a solution that is highly flexible, domain-independent, rapidly deployable, easily adapted across domains, minimally demanding in setup, and truly user-friendly.

These contributions collectively address the limitations of both traditional and LLM-based AI approaches (identified respectively in section 3.1 and section 3.2), thus bridging the gap between state-of-the-art classification systems and real-world usability. To my best knowledge, no prior work has introduced a zero-shot, LLM-based document classification framework that jointly addresses all these qualities in a single solution, marking a significant step toward more inclusive and adaptable document intelligence.

1.4 Document Structure

This document is organized as follows:

- Chapter 1 - introduces the research problem, objectives, and motivation of this work.
- Chapter 2 - reviews the state of the art in document classification, discussing its historical evolution, main challenges, and recent approaches already integrating LLMs.
- Chapter 3 - presents a critical analysis that bridges the state of the art and the current challenges, identifying requirements that a framework should address to be successful.
- Chapter 4 - describes the implementation of the proposed framework, explaining its architecture and functioning in detail.
- Chapter 5 - evaluates the framework through experimental studies, assessing its performance in terms of accuracy, costs, and other relevant metrics.
- Chapter 6 - concludes the dissertation by summarizing the main findings, discussing limitations, and outlining directions for future work.

Chapter 2

Literature Review

The present chapter explores the evolution of document classification by drawing a parallel with the major milestones in the history of Artificial Intelligence (AI). To structure this discussion, the chapter is divided into two main parts: traditional approaches and modern approaches based on LLMs. Figure 2.1 illustrates the organization of the subsections, highlighting how the narrative progresses from the early foundations and feature engineering, through the machine learning revolution and finally reaching the current LLM-based methods.

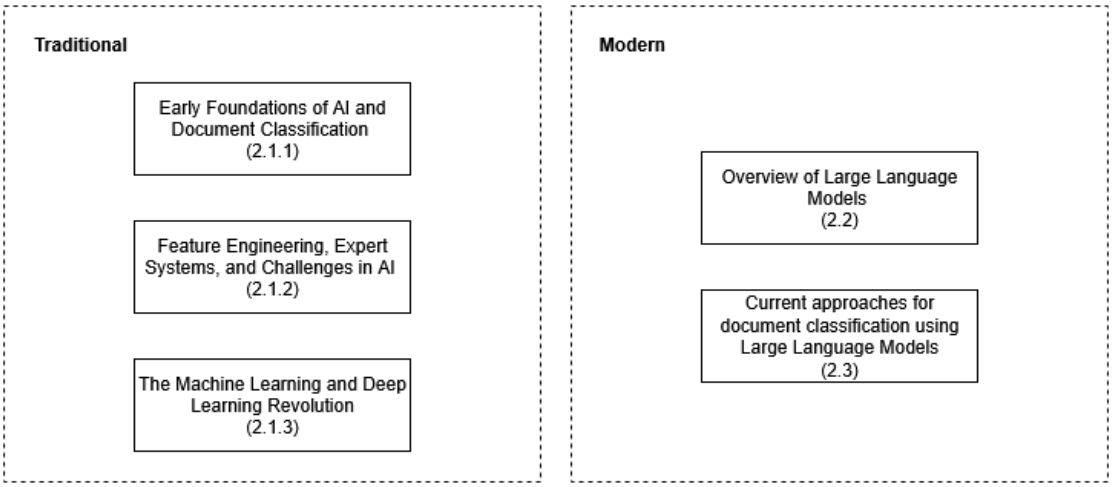

Figure 2.1: Traditional vs. Modern approaches in Document Classification

2.1 Historic Evolution

Since the beginning, automatic document classification has closely followed the evolution of AI itself, and so, the Historical Evolution section seeks to analyze this joint journey. As illustrated in Figure 2.1, the journey begins with the initial foundations of AI and document classification (subsection 2.1.1), followed by a phase with greater emphasis on feature engineering, expert systems, as well as the challenges these approaches faced (subsection 2.1.2). Finally, this section presents the revolution driven by machine learning and deep learning (subsection 2.1.3), which significantly

transformed document classification techniques and represented the last major advance before the current state of the art dominated by LLMs.

2.1.1 Early Foundations of AI and Document Classification

The origins of AI date back to the second half of the 20th century, when the advent of the first computers gave rise to intriguing questions such as, ”Can a machine think?” [1]. Alan Turing was the first to challenge the odds through his article ”Computing Machinery and Intelligence”, which is regarded by many in the scientific community as the starting point of AI. However, at that time, this article remained a singular contribution, as AI was neither a discipline nor a research field, due to the lack of a research community and, consequently, the absence of intelligent systems [20].

A few years later, the term 'Artificial Intelligence' was used for the first time through the proposal of the ”Dartmouth Summer Research Project”, where ten researchers studied how machines could simulate learning and intelligence [2].

The first works on document classification appeared in the mid-1960s with a mathematically derived and empirically based through the application of factor analysis technique [3]. This technique infers the meaning or thematic content of a document by statistically examining its words, based on the assumption that a document can be classified according to the words it contains. This experiment led the authors to very interesting conclusions for the time, more precisely that automated document classification was viable. [3].

2.1.2 Feature Engineering, Expert Systems, and Challenges in AI

In the following years, significant milestones were achieved in the field of NLP, one of which was the introduction of the Term Frequency-Inverse Document Frequency (TF.IDF) algorithm [4]. This algorithm was introduced by Karen Jones in 1972, originally in the context of information retrieval systems [4]. Since then, TF.IDF has become a widely used technique for evaluating the importance of words within a *corpus* of documents by weighting terms based on their occurrence and significance [5]. As a result, this algorithm identifies the most unique content from each document [4, 5], although its application to automated document classification did not become prominent until later decades.

Until 1974, there was a wave of optimism, growth, and apparent progress, giving rise to the golden age of AI, as everything seemed possible [20]. Advances in AI were remarkable, including in automatic document classification, where the research community was highly focused on refining and improving existing algorithms. A relevant example is the work conducted by Dattola [28], who identified a significant limitation in the algorithms developed up to that point for automatic classification. The need to compute a complete similarity matrix between documents resulted in high processing times and resource consumption. This issue was even more critical considering the computational limitations of the time. To address this problem, Dattola proposed a new algorithm that, instead of calculating all pairwise similarities between documents, compared each document with a set of predefined clusters. Through this approach, the construction

of the complete similarity matrix was avoided, thus reducing the algorithm's complexity from N^2 to $N.log(N)$, without resulting in the loss of relevant information [28]. The introduction of this new technique represented a substantial improvement in terms of performance and scalability, especially considering the quadratic nature of previous methods, which often became impractical in contexts with large volumes of data [28, 29].

Despite the optimism and enthusiasm surrounding AI, by the mid-1970s, progress had stagnated, leading many to believe that domains associated with intelligent activity were falling into disuse and approaching their end [19]. The atmosphere of skepticism within the scientific community stemmed largely from the perception that AI was focused on solving overly generic problems, neglecting an essential element for any truly intelligent system: knowledge [20].

In the mid-1980s, a new knowledge-based solution emerged, known as expert systems, which operated using rules and heuristics to capture human knowledge and subsequently use it in decision-making [20]. Although these systems generated considerable enthusiasm within the community, particularly due to their simplicity, transparency, and the ease with which the underlying "reasoning" behind the outputs could be interpreted, knowledge acquisition became a significant limitation. The construction of rules and heuristics had to be formulated by a knowledge engineer and a domain expert [6, 7]. Furthermore, the performance and flexibility of these systems were strongly related to the quality and accuracy of the rules in correctly representing the specific use case [8]. For these reasons, by the end of the 1980s, systems based on this approach experienced a steep decline and collapsed, giving rise to what became known as the second AI winter, lasting until the mid-1990s [21]. Despite practical experiments coming to a standstill, the development and release of public benchmark datasets was, nevertheless, an important milestone and proved to be very useful in the subsequent years [21].

2.1.3 The Machine Learning and Deep Learning Revolution

As a result of the emergence of Machine Learning (ML) at the end of the 20th century and the beginning of the 21st century, some of the most popular algorithms were tested for document classification, such as K-Nearest Neighbors (K-NN) [9, 10], Naïve Bayes [11, 12], Decision Trees [13, 14], and even strategies employing MapReduce [30]. At the same time, the first experiments began to explore the potential of neural networks (NNs) [15, 16, 17].

K-NN methods use the Vector Space Model (VSM) to represent documents as feature vectors and, consequently, to categorize them by measuring similarities between the centers of the nearest neighbors [9, 10]. This algorithm is known for its simplicity and strong performance in classification tasks (including document classification). Since this method does not rely on assumptions about the underlying data distribution, it is adaptable to various datasets. However, similarity calculations can be computationally expensive, and its sensitivity to imbalanced data distributions may lead to misclassifications near category boundaries [9, 10, 31].

Naïve Bayes is a probabilistic classifier that assumes independence among variables (in this case, the features of the document) and operates based on the concept of likelihood to estimate the

probability that a document belongs to a given category. To this end, it employs models such as the Multinomial and Bernoulli, to represent word occurrences, as well as techniques like TF.IDF to weight terms according to their relevance and discriminative power [11, 12, 31]. This approach makes Naïve Bayes efficient and straightforward, enabling it to handle large volumes of text with minimal computational requirements. It performs well with a small amount of training data and can yield accurate results, especially when combined with feature selection methods to refine the input. However, the main limitation lies in the independence assumption, as it disregards context and relationships between words, potentially reducing classification accuracy in cases where semantics and context play a crucial role [11, 12, 31]. Some attempts to mitigate this limitation include incorporating domain-specific knowledge through ontologies [11]. Nevertheless, such solutions demand manual intervention of a technical and specialized nature, thus, reducing adaptability and flexibility to dynamic or continuously evolving datasets.

Decision tree models are also common approaches for document classification, owing to their recursive "divide and conquer" methodology [31], which offers high interpretability and enables users to easily understand the "reasoning" underlying the classifications, a crucial feature for applications such as personalized content recommendations [14]. Moreover, decision trees can be optimized for efficiency; for example, the Naive Tree (NT) algorithm, which combines a decision tree with Naïve Bayes, significantly reducing the construction time compared to traditional methods like C4.5 [13]. However, their simplicity makes this approach susceptible to overfitting, especially when applied to high-dimensional data. In addition, decision trees often struggle to capture complex patterns and contextual relationships inherent to natural language, which can compromise performance in more sophisticated text processing tasks [13, 14, 31].

NNs have emerged as powerful tools for various use cases, including document classification. These models are composed of multiple layers of interconnected nodes, or neurons, which enable the representation of a wide range of functions and patterns, allowing for the capture of complex relationships between input and output variables [18]. Models such as convolutional neural networks (CNNs), widely used in image-based classification [32], and backpropagation neural networks (BPNNs), applied to structured documents [17] or Recurrent Neural Networks (RNNs) offer great adaptability to different types of documents. Furthermore, NNs benefit from efficiency gains through techniques such as dimensionality reduction [15], and demonstrate good performance even in scenarios with exclusively positive samples [16]. Although these approaches combine several advantages, they also present significant limitations, such as the need for high computational resources during training and their susceptibility to overfitting when small datasets are used. As a result, large volumes of labeled data are frequently required to achieve good results. In addition, another challenge rises from the lack of explainability, since the complexity of interactions between layers makes it difficult to humans understand the decisions made [18].

The rapid growth in the availability and accessibility of documents has led to the need for, and interest in, automatic text extraction [33, 34], a task also known as Optical Character Recognition (OCR). The earliest studies in this area date back to the period between the 1950s and 1970s,

when initial efforts faced significant limitations, such as slow processing speeds, low accuracy, restricted character recognition capabilities, and limited computational power [34, 35], which hindered significant progress. However, the rise of ML introduced many innovative approaches to OCR tasks [35]. The increasing sophistication of this technique had a major impact on the field of NLP and, consequently, on document classification, as experiments could henceforth rely not only on born-digital documents but also on texts extracted from non-native sources, whether scanned and handwritten materials. This achievement considerably expanded both the range of accessible documents and the volume of available textual data, thereby, contributing to the development of greater models.

In subsequent years, NNs became larger and deeper [20], and were increasingly tested and adapted for document classification tasks, even when they were not originally designed for such purposes. A notable example is a study that employed images of photographed documents to fine-tune a pre-trained AlexNet architecture, which was originally developed for visual recognition and image classification tasks [36]. In this way, using a photographed image of a document, two distinct problems were addressed: (1) determining to which book the image belongs, and (2) identifying the type of book represented by the image [32].

Between 2012 and 2017, substantial work was carried out in the field of document classification, utilizing a wide range of methods. According to a 2018 state-of-the-art survey in this research area, considering a sample of approximately 242 articles, 66% were published between 2012 and 2017 [37]. This finding highlights that, nearly 55 years after the earliest publications, document classification remained an active area of interest for the AI research community.

Until 2013, language modeling was done by considering words as atomic units, without any notion of similarity or relationship between words [38]. A popular example is the N-gram model used in statistical language modeling, which uses a limited history of word dependencies to predict the probability of a sequence of words [38, 39]. However, the introduction of the groundbreaking *Word2Vec* algorithm revolutionized the way text is processed, due to its ability to represent words as vectors (or embeddings) in a continuous vector space. This approach captures both the semantic and syntactic relationships between words, enabling a better understanding of language [40, 38]. These word embeddings, and their extensions such as *Doc2Vec*, had a significant impact on document classification tasks. By representing documents as aggregations of word or paragraph embeddings, these models enabled classifiers to exploit semantic similarities and relationships between words and documents, leading to substantial improvements in classification accuracy compared to traditional bag-of-words or N-gram approaches [40, 41].

The history of document classification reflects a continuously evolving and dynamic field of research. From early statistical approaches to the introduction of distributed representations such as *Word2Vec* and *Doc2Vec*, each generation of methods has addressed the increasing complexity and diversity of real-world text data. Although the early 2010s saw substantial progress with the introduction and widespread adoption of neural word and document embeddings, the field continued to advance rapidly in subsequent years. Notably, the emergence of transformer-based

models in 2017 marked a significant shift, establishing new paradigms for language modeling and document understanding. The next sections will explore these recent developments (section 2.2) and examine their implications for document classification (section 2.3).

2.2 Overview of Large Language Models

The emergence of the Transformer architecture [22] was fundamental to the development of modern LLMs, which predominantly rely on this design. By stacking multi-head attention layers within a very deep neural network [23], LLMs can effectively capture complex contextual relationships and long-range dependencies in textual data [22, 25]. As a result, LLMs are highly suitable for document classification compared to traditional approaches, which do not take sequential structure or contextual information into account and, consequently, fail to capture semantic meaning and relationships between words [24]. Models such as Llama, developed by Meta AI [42]; Claude, introduced by Anthropic [43]; and GPT, created by OpenAI [44], are prominent examples of LLMs that perform exceptionally well across different NLP tasks, demonstrating remarkable proficiency in language understanding [45].

The success of LLMs is closely tied not only to the strong interest of the research community, with more than 5000 publications in recent years [25], but also, to their rapid adoption in industry and everyday life by both technical and non-technical users. For instance, ChatGPT, launched in 2022, reached 1 million users within 5 days, making it the second fastest application to achieve this milestone[26]. As of mid-2025, it now boasts between 800 million and 1 billion weekly active users [46]. Beyond this, ChatGPT has become widely integrated into daily routines, supporting a broad spectrum of tasks ranging from technical activities, such as programming and data analysis, to non-technical uses, including education, communication, and creative writing [47].

There are several ways to interact with and utilize LLMs. One of the simplest is *prompt engineering*, which involves designing inputs, also known as *prompts*, in order to guide the model's behavior with precision. Instead of modifying the model's parameters (like in traditional NNs), prompt engineering steers its outputs through carefully crafted language [48]. This form of interaction is especially familiar to non-technical users, who can typically provide instructions in a conversational style to achieve their intended outcomes. Given its accessibility and potential, emerging methods such as Requirement-Oriented Prompt Engineering (ROPE) focus on teaching users how to articulate clear and comprehensive requirements within prompts. Studies have shown that ROPE significantly improves prompt quality compared to standard prompt engineering approaches [49].

In addition to prompt engineering, there are learning paradigms that can improve performance by leveraging examples. These examples can be designed and provided in natural language without the need for retraining or modifying model hyperparameters. The main paradigms include:

- *Zero-shot learning*: The model performs a task without any examples, relying solely on pre-training knowledge and task description [50, 51].

- *One-shot learning*: The prompt includes exactly one example helping clarify task requirements and expected responses [50].

- *Few-shot learning*: Several examples (typically 2–10) are provided in the prompt as contextual demonstrations, enabling the model to infer patterns or rules without parameter updates. This allows for quick adaptation to new tasks, although it requires more examples and can sometimes limit generalization.[50, 51].

A completely different approach from the previous methods for utilizing LLMs is *fine-tuning*, which refers to adapting a pre-trained model to better suit specific tasks or align with human preferences [51]. Modern LLM applications can also involve fine-tuned models, where retraining is performed to adjust hyperparameters, in a manner similar to traditional NNs. This can be done, whether via supervised learning on specialized datasets or through Reinforcement Learning from Human Feedback (RLHF), which is a particularly potent technique, as it can shape model responses to better reflect user norms, filter out toxic language, improve factual accuracy, and reduce hallucinations [51]. Recent studies [52] also describe additional fine-tuning methods, such as *Low-Rank Adaptation (LoRA)*, *Half Fine-Tuning*, and optimization strategies like *Direct Preference Optimization*, which further improve efficiency and alignment with human preferences.

At this stage, LLMs have already been widely applied across various domains, both academic and industrial, due to their ability to generalize and adapt to multiple contexts. According to recent surveys [23] and studies [53], several key areas of impact can be identified:

- Conversational agents and chatbots: One of the most common applications of LLMs is in the development of virtual assistants and customer support systems. These models enable more natural and contextually appropriate interactions and are used in various industries such as banking, education, healthcare, among others [23, 53].

- Summarization and information extraction: LLMs are effective in condensing large volumes of text into concise summaries and extracting relevant information [23, 53].

- Translation: LLMs perform translation both for natural languages and programming languages. For language translation, they leverage linguistic and contextual knowledge to achieve competitive performance, often approaching task-specific supervised systems [53]. For code translation, LLMs can convert code snippets between programming languages while preserving functionality and structure [23].

- Content generation: These models are also used for producing reports, technical documents, marketing material [53], and even for creative writing, such as stories or poetry [23].

- Programming and software engineering: LLMs can act as powerful programming assistants, generating code snippets, explaining code blocks, and identifying errors, thereby supporting developers throughout the software development process [23].

These are just a few examples, and many more applications exist. According to a recent study [53], LLMs are transforming various sectors, namely healthcare (diagnosis support, personalized treatment plans, and patient data analysis), automotive (car virtual assistants and predictive maintenance), and consumer services (recommendation systems and behavioral analysis) [53].

These applications highlight the versatility of LLMs, which function as cross-cutting technologies impacting everything from traditional linguistic tasks to specialized domains in science and industry. Consequently, LLMs represent a disruptive technology that is fundamentally reshaping task execution across diverse fields. Their capacity to understand and generate human-like text is transforming how individuals and organizations approach complex NLP tasks, enabling innovative solutions and more efficient workflows that were previously unattainable.

2.3 Current approaches for document classification using Large Language Models

Some automatic document classification solutions have already explored the use of LLMs, particularly in the legal domain, where the need for precise interpretation and handling of complex textual structures is essential. One example is the Trautmann's study [54], which proposes a method for classifying lengthy legal documents using a chain-of-thought process in which the main classification task is decomposed into a series of smaller, prompt-driven steps. The first prompt aims to generate a summary of the input document, extracting its essential information. Next, the *sentence-transformers* library [55] is employed, using the *Custom Legal-BERT* model (specialized in legal domain documents) [56] to encode the summary produced in the previous step. Then, a nearest-neighbor search is performed within a corpus of legal documents to retrieve the summaries most similar in semantic terms. The final step consists of a label generation prompt that queries the LLM through a few-shot prompt configuration, combining the previous examples with an instruction specifically tailored to the classification task and a predefined list of expected labels for the model to choose from [54].

The study conducted by Prasad, Boughanem, and Dkaki [57] introduce the Multi-stage Encoder-based Supervised with Clustering (MESC) framework [57], a hierarchical approach for classifying lengthy and unstructured legal documents. Firstly, this method splits the document in smaller chunks, each one associated with the document label. These segments are then used to fine-tune a language model (like, BERT or GPT), enhancing their ability to capture contextual representations specific to legal texts. Subsequently, embeddings are extracted using the last four layers of the fine-tuned model, thereby capturing multiple levels of semantic information. To address structural complexity, the authors employ a supervised clustering process enriched with attention layers that allow interactions among the chunks. Finally, the resulting representations are combined with structural information of the document and processed through additional layers to produce the final classification [57].

Another study, although not focused on document classification, offers an interesting approach to image classification whose central concept can be effectively adapted to document classification.

Menon and Vondrick [58] created a zero-shot framework using Visual Language Models (VLMs) in a two-stage classification approach. First, a detailed description for each label is generated by an LLM (such as GPT-3). Then, when an image is presented, the framework extracts its features and compares them with the descriptions generated in the first stage. This approach is simple and does not require a large pre-existing database; however, it is highly dependent on the quality of the descriptions generated in the initial stage [58].

To the best of my knowledge, one of the few studies relating LLMs to document classification in the banking sector is the Loukas et al. article [59], where the authors explore cost-effective strategies for user intent classification tasks in scenarios where labeled data is scarce. The authors demonstrated that few-shot learning with models such as GPT-4 can be highly effective for text classification in the banking sector, even with only a few examples per class. However, the use of this model can be expensive, as access is restricted by paywalls [59]. To address this, a Retrieval-Augmented Generation (RAG)-based solution was implemented, in which only the most relevant examples are retrieved for LLM processing, thereby reducing operational costs without sacrificing accuracy [59].

There are also some works on text classification that address traditional problems, such as sentiment analysis [60] and customer evaluation [61]. However, the task of document classification introduces additional layers of complexity, as highlighted in section 1.1, due to the significant variability in document format, content, length, and structure. Unlike text classification, which operates on already structured and readily accessible textual data, document classification requires more complex processing. This often involves multiple stages, including OCR for text extraction and further processing to structure and organize the raw text, enabling a better interpretation and contextualization. These steps are crucial for optimizing the classification process and achieving reliable results.

Chapter 3

Analysis

This chapter critically examines the existing approaches to document classification, highlighting their limitations and identifying gaps that motivate the development of the proposed framework. While the previous chapter reviewed the evolution from early AI systems to LLMs, this section focuses on analyzing practical constraints and sector-specific challenges, particularly in the context of the banking domain.

3.1 Traditional Approaches

Throughout history, document classification has been a consistently evolving topic, reflecting the research community's enduring interest due to its practical relevance in real-world contexts. Despite considerable advancements, traditional approaches have continued to exhibit several notable limitations.

Early statistical and mathematical methods, which heavily relied on the characteristics of the specific use case, lacked a crucial ingredient for an AI system, the knowledge [20].

This limitation led to the emergence of expert systems, which, despite their initial promise, were hindered by the complexity and expert-dependency inherent in knowledge acquisition [6, 7, 8], ultimately contributing to their decline in practical use.

The advent of ML introduced a totally new and invigorating perspective, bringing with it a variety of methodological approaches. While these innovations enabled significant progress, they also introduced new constraints, such as the high computational costs and sensitivity to imbalanced data distributions observed in methods like k-nearest neighbors [9, 10, 31]. Probabilistic classifiers, such as Naïve Bayes, were restricted by strong independence assumptions, thereby neglecting crucial contextual relationships within the data [11, 12, 31]. Similarly, decision tree models, although interpretable, were prone to overfitting and often struggled to capture the intricate and context-dependent patterns typical of natural language [13, 14, 31].

The introduction of Word2Vec and its extension Doc2Vec marked a significant improvement in the language modeling. By embedding words and documents into continuous vector spaces they were capable for the first time in history to capture semantic and syntactic relationships [40, 38, 41]. Altough these embeddings enhanced the interpretation of text, directly benefiting applications

such as document classification and surpassing traditional bag-of-words and N-gram techniques, they also exhibited certain limitations. Specifically, they required large, high-quality corpora, offered limited interpretability, and primarily captured co-occurrence patterns rather than deeper, context-dependent relationships. Consequently, their effectiveness was constrained in complex, domain-specific, or resource-poor scenarios.

More advanced approaches, namely neural networks (including CNNs, BPNNs, RNNs, *Doc2Vec*, among others), despite their adaptability and representational power, demanded substantial computational resources, large labeled datasets, a pronounced propensity to overfit, particularly in scenarios with scarce training data and suffered from limited explainability [15, 16, 17, 18].

The persistence of such limitations in conventional AI systems historically contributed to the occurrence of two major AI winters, mostly due to unmet expectations and overhyped promises [20]. The inability to develop systems that were simultaneously intelligent, easy to use, and adaptable to various tasks hampered both research momentum and investment, as development and progress in the field virtually froze during these periods.

These historical and technical shortcomings underscore the importance of designing modern frameworks that avoid repeating past failures. In particular, contemporary systems must aim to be intelligent, scalable, user-friendly, domain-independent, and adaptable, reducing expert-dependency while enhancing robustness in real-world, often low-resource, document processing contexts. This rationale sets the stage for the adoption of modern LLMs, which promise to overcome many of these enduring barriers.

3.2 Modern Approaches

Recent advances in document classification using LLMs have demonstrated impressive performance and improved interpretability compared to traditional approaches, although some limitations persist.

A significant portion of current research is focused only in the legal sector [54, 57, 62], while other areas of high interest, such as banking and finance remain underexplored [59]. Moreover, many studies focus on text classification rather than full document processing, ignoring the additional complexities of raw, multi-modal, or structurally rich documents, including tasks such as OCR, text extraction, and contextual organization [60, 61].

Existing approaches also differ in operational requirements and limitations. For instance, some methods demonstrate strong interpretability and facilitate debugging in complex reasoning tasks but rely on pre-existing document corpora, restricting their applicability [54, 57]. Other strategies avoid large databases but depend heavily on initial descriptive prompts, making them sensitive to prompt design [58]. Few-shot learning with models such as GPT-4 has proven effective even with minimal examples [59], yet high operational costs and access restrictions can hamper widespread adoption. To mitigate these constraints, hybrid approaches leveraging RAG have been proposed, retrieving only the most relevant examples to balance cost and accuracy [59].

Despite these innovations, persistent challenges limit real-world deployment. Current systems

often lack configurability and user-centric design [54, 57], hindering adaptability in enterprise environments where end-user usability is crucial. Many frameworks depend on curated datasets or few-shot paradigms [54, 59, 27], which can be impractical in low-resource or dynamic settings where labeled data is scarce or constantly evolving. Additionally, the high semantic similarity between documents from distinct classes requires models capable of nuanced contextual reasoning, rather than relying solely on surface-level textual features.

Taken together, these observations reveal a gap in existing LLM-based document classification frameworks: there is no unified solution that combines zero-shot capability, domain independence, user configurability, and adaptability to handle complex documents. This unmet need directly motivated the development of the framework proposed in this dissertation.

3.3 Requirements for a new Framework

The preceding analysis highlights several enduring limitations in both traditional and LLM-based document classification systems, which inform the essential requirements for a new framework. Firstly, domain independence is critical: the framework must operate effectively across diverse sectors, avoiding over-specialization that restricts applicability. Secondly, adaptability and robustness are paramount. Real-world document processing often involves heterogeneous, unstructured, or low-resource datasets, necessitating systems capable of handling varied formats, missing data, and noisy inputs.

User-centric design constitutes a third requirement. The framework should provide intuitive configuration mechanisms, allowing users to define classification options and adjust other parameters in a simple manner. These adjustments should contribute to a clear and transparent decision-making process, enhancing the accessibility of the framework to all users, including those with limited technical expertise.

Beyond user-oriented design, the need for efficiency and scalability is crucial, as high computational costs, heavy reliance on large volumes of data, and limited access to state-of-the-art LLMs remain significant barriers. That said, an effective framework must therefore optimize the use of available resources without compromising classification accuracy.

In summary, a new document classification framework must combine domain independence, robustness, user-centric design, operational efficiency, and advanced semantic reasoning to address the shortcomings of existing approaches and ensure practical viability.

3.4 Bridging Analysis to Proposed Solution

Building on these requirements, the proposed framework aims to directly address the limitations identified in prior sections.

The integration of zero-shot capability enables the framework to avoid reliance on large, pre-curated corpora, while maintaining flexibility to operate across multiple domains. The use of LLMs ensures that contextual reasoning is incorporated, thereby capturing subtle semantic re-

lationships within documents, going beyond superficial text analysis. The framework was designed with configurable workflows and a very simple and intuitive configuration method (section 4.1), ensuring accessibility for non-specialist operators and adaptability to various business contexts. Moreover, efficiency considerations are embedded through selective retrieval and processing strategies, minimizing computational overhead and costs without compromising classification accuracy.

By bridging historical insights, advances provided by incorporation techniques, and the latest developments in LLMs, this framework proposes a holistic solution aligned with both the theoretical and practical requirements of modern document classification. This establishes a solid foundation for robust, scalable, and user-friendly systems capable of overcoming the challenges identified above.

Chapter 4

Implementation

The proposed framework, illustrated in Figure 4.1, can be divided into four different macro-components, and each one will be described in detail in the following subsections.

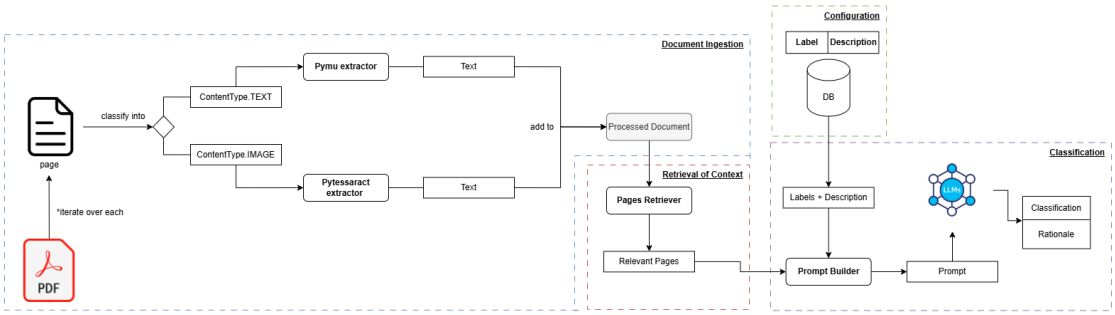

Figure 4.1: Architecture diagram subdivided into the four macro-components. The "Retrieval of Context" component is abstracted and generic since various implementations are possible.

4.1 Configuration

The only prerequisite for using the proposed framework is parameterize the possible classification labels into which a document can be categorized. This involves defining each **label** as a unique identifier for a category, along with a **description** that functionally defines what the category represents. All pairs of description labels used in the experiments (chapter 5) by the framework are available in the Appendix (Table A.1).

This parameterization allows the framework to systematically organize documents according to predefined categories facilitating efficient retrieval, analysis and decision-making based on data. In particular, the way the framework is built inherently supports this high degree of configurability, allowing it to adapt dynamically to a wide variety of use cases. As a result, users from all backgrounds, regardless of their technical knowledge, can take advantage of the framework effortlessly. By simply specifying their interpretation of the nature or purpose of a document, they can immediately benefit from its capabilities. This combination of structural flexibility and user-centered design makes the framework not only powerful, but also exceptionally intuitive, accessible, ex-

tendable for different domains and widely applicable.

4.2 Document Ingestion

The framework begins by opening the document using *fitz* library from *PyMuPDF* [63]. Then each page will be categorized into one of two groups: those with extractable and valid text and those without. This categorization is used to determine the type of extractor the framework applies to each page, ensuring that the appropriate method is used to extract the page content. For pages containing extractable and valid text, the *PyMuPDF* loader is used, as it has demonstrated robust and reliable performance in text extraction tasks. On the other hand, the pages without valid extractable text are converted to images and passed through the *pytessaract* extractor [64], a wrapper for Google's Tessaract-OCR Engine [65]. Since this can be time-consuming, parallelization was implemented, enabling the handling of multiple pages simultaneously for improved efficiency. At this point, all pages have been processed, and the text has been successfully extracted.

4.3 Retrieval of context

Since the number of pages in each document varies, and the context-window of LLMs is limited, selecting the right content to be provided is crucial, as it can significantly impact the classification results. To address this, the framework was designed to be highly flexible, allowing for different context retrieval methods. Currently, there are four approaches implemented for retrieving document content.

Notation. Let a document D be an ordered list of P pages:

$$D = [p_1, p_2, \ldots, p_P].$$

A context selector returns a set of *retrieved pages* $R \subseteq \{1, \ldots, P\}$ of size at most a budget K, where K is a configurable parameter that limits the number of processed pages.

4.3.1 Strategy 1 - First Pages

The first retrieval method mirrors a natural human approach to assessing a document. When opening a file, a person begins examining the initial N pages before forming a preliminary judgment regarding its content. This strategy operates under the assumption that the most relevant information is positioned towards the beginning of the document. Although conceptually simple, it can be effective in scenarios where documents are structured such that critical information is presented early. Formally, given a document D composed of P ordered pages, the retrieved page set is defined as:

$$R_{\text{First Pages}}(D, K) = \{1, 2, \ldots, K\} \tag{4.1}$$

4.3.2 Strategy 2 - First & Last Pages

The second retrieval method extends the natural human approach by incorporating content from both the beginning and the end of the document. Rather than relying solely on the first N pages (Equation 4.4), this method also retrieves the last M pages (Equation 4.5). The underlying assumption is that, while introductory sections provide essential context, concluding sections often contain summaries, results, or final remarks that can be equally informative for classificatin. By combining both ends of the document (Equation 4.6), this approach aims to capture a broader range of relevant information.

Formally:

$$Limit_{FirstPages}(K) = \lfloor \tfrac{K}{2} \rfloor \tag{4.2}$$

$$Limit_{LastPages}(K) = K - Limit_{FirstPages} \tag{4.3}$$

$$Firsts = \{1, 2, \ldots, \min(Limit_{FirstPages}, P)\} \tag{4.4}$$

$$Lasts = \{\max(1, P - Limit_{LastPages} + 1), \ldots, P\} \tag{4.5}$$

$$R_{\text{First \& Last Pages}}(D, K) = Firsts \cup Lasts \tag{4.6}$$

4.3.3 Strategy 3 - Full Text Vectorization

The Full Text Vectorization strategy uses TF.IDF which is a classic and widely used approach to retrieve relevant content from a corpus of documents [4, 5].

The overall workflow is illustrated in Figure 4.2, where each step of the process is mapped to its corresponding mathematical formulation.

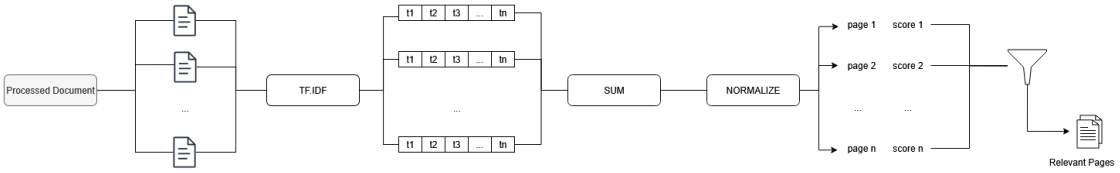

Figure 4.2: TF.IDF pipeline, showing the steps until Top-K page selection.

In this adaptation, the input document D is considered as a collection of pages $\{p_i\}_{i=1}^{P}$, forming the corpus. The objective is to evaluate the importance of terms on each page relative to the other pages within the same document. For this purpose, each page is converted into a TF.IDF vector using the standard TF.IDF formulation:

$$\mathbf{v}_i = \text{TF.IDF}(p_i) \in \mathbb{R}^V, \quad i = 1, \ldots, P \tag{4.7}$$

where V is the size of the vocabulary. The TF.IDF value for term t in page p_i is computed as:

$$\text{TF.IDF}_{t,i} = \text{TF}_{t,i} \cdot \text{IDF}_t \tag{4.8}$$

with Term Frequency (TF):

$$\mathrm{TF}_{t,i} = \frac{f_{t,i}}{\max_k f_{k,i}} \tag{4.9}$$

and Inverse Document Frequency (IDF):

$$\mathrm{IDF}_t = \log_2 \frac{P}{n_t} \tag{4.10}$$

where:

$f_{t,i}$ is the number of occurrences of term t in page p_i

n_t is the number of pages containing t

P is the total number of pages in the document.

Next, the relevance score of each page is obtained by summing the TF.IDF values across all terms:

$$s_i = \sum_{t=1}^{V} \mathrm{TF.IDF}_{t,i}, \quad i = 1, \ldots, P \tag{4.11}$$

The scores are then normalized to ensure comparability:

$$\hat{s}_i = \frac{s_i}{\|\mathbf{s}\|_2} \tag{4.12}$$

Finally, the pages are ranked according to their normalized scores, and the top K most relevant pages are selected:

$$R_{\text{TF.IDF}}(D, K) = \mathrm{TopK}\Big(\{(p_i, \hat{s}_i)\}_{i=1}^{P}, K\Big) \tag{4.13}$$

This procedure allows for identifying the most informative pages of a document based solely on term importance, without requiring external embeddings or pretrained models.

4.3.4 Strategy 4 - Retrieval-Augmented Generation

RAG is a hybrid technique designed to enhance the performance of LLMs by integrating an external knowledge base, in this specific case, the information contained within the document, rather than relying solely on pre-trained LLM knowledge [66, 67, 68].

The overall workflow is illustrated in Figure 4.3, where each step of the process is mapped to its corresponding mathematical formulation.

In this approach, a document D is split into pages $\{p_i\}_{i=1}^{P}$, and each page is converted into an embedding vector using a high-capacity embedding model, specifically *text-embedding-3-large* from OpenAI [69]:

$$\mathbf{e}_i = f_{\text{text-embedding-3-large}}(p_i) \in \mathbb{R}^d, \quad i = 1, \ldots, P \tag{4.14}$$

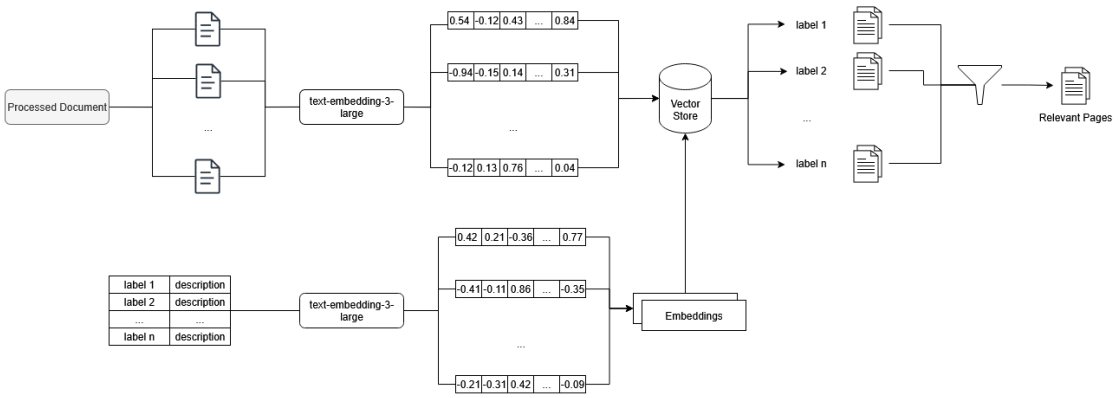

Figure 4.3: RAG pipeline, showing the main steps from document embedding to final Top-K page selection.

Each embedding vector is persisted into a vector store (specifically *In Memory Vector Store* [70]), which serves as a document-specific knowledge base. This enables efficient retrieval and management of embeddings [71].

In parallel, each label l_j, defined at the configuration stage (section 4.1) and associated with a description d_j, is embedded using the same model:

$$\mathbf{e}_{q_j} = f_{\text{text-embedding-3-large}}(d_j) \in \mathbb{R}^d, \quad j = 1, \ldots, L \tag{4.15}$$

The vector store is then queried to retrieve the top K most similar pages for each label, based on cosine similarity:

$$S_j = \text{TopK}\Big(\{(p_i, \cos(\mathbf{e}_i, \mathbf{e}_{q_j}))\}_{i=1}^{P}, K\Big) \tag{4.16}$$

Finally, the resulting candidate lists are combined and refined by applying a global filter that retains only the top K most similar pages overall, regardless of their associated label. This prevents context pollution and avoids an excessive growth in the number of retrieved pages ($K \times L$):

$$R_{\text{RAG}}(D, \{d_j\}_{j=1}^{L}, K) = \text{TopK}\Big(\bigsqcup_{j=1}^{L} S_j, K\Big) \tag{4.17}$$

This approach leverages high-dimensional search capabilities to efficiently manage and retrieve embeddings from the vector store [71]. By querying the *In Memory Vector Store* the semantic richness of the stored data is explored, allowing the most relevant or similar information to be identified and retrieved by measuring the cosine similarity between the stored embeddings and the input query[72, 70]. Thus, this process ensures that the model works with a more relevant and controlled context without compromising its performance.

4.4 Classification

At this stage, all necessary components are assembled for input into the LLM. The framework automatically integrates the contextual data (including possible labels and retrieved pages) with detailed instructions to construct a comprehensive prompt for input into the LLM. This carefully crafted prompt guides the LLM in understanding the task requirements and effectively using the provided context. Indeed, this prompt is built upon three parts.

Firstly, it is crucial to **structure the contextual data** so that the LLM can interpret it effectively, since its performance can be sensitive to prompt formatting [73]. To address this, *XML* was selected as the preferred format, and consequently, both the label-description pairs (from the configuration stage, section 4.1) and the top K pages (obtained in the preceding step, section 4.3) are structured into *XML* tags, ensuring that the information is presented clearly and organized for optimal processing by the LLM.

Detailed instructions are defined by the **system prompt** (available on Appendix, Listing A.1), which guides the model in its decision-making. The prompt begins by assigning the model a specific *persona*, following best practices for prompt design [74, 75]. Instead of giving very specific instructions or detailed explanations of the input data, which could hinder scalability as data volume and complexity grow, the system prompt provides general guidelines for the task, only adding instructions to address edge cases or emphasize important considerations. This approach ensures that the final prompt remains manageable and scalable, regardless of the volume or diversity of the input data.

The concluding element of the prompt is an output parser configured using the **Pydantic Model** framework, ensuring that the model returns the output in the defined structured format and is composed of two components: *rationale*, indicating that the model should explain its reasoning before the decision making; *classification* instructing the model to choose the category that best suits the document, taking into account only the labels given. These components are sorted in this exact order because asking the model to *"think"* or reason first can significantly improve the performance of a model [76] and, consequently, also the classification precision.

Since the context window capacity varies between LLMs, the framework includes a security mechanism that monitors and controls the number of tokens submitted in each request. It first estimates the total size of the prompt, including the *system prompt*, the *output parser*, and the selected content from the document. Then, if the limit is exceeded the framework truncates the context to prevent execution errors.

Finally, and with all the pieces in place, the LLM is invoked to perform the classification task saving the output, composed by the label and the explanation/rationale, in a JSON file.

A pair of input prompt (Listing A.2) and corresponding output response (Listing A.3), illustrating the interaction with the model, can be found in the appendix for reference.

Chapter 5

Experiments

To evaluate the global performance of the proposed framework, a process of collecting and compiling relevant documents was performed, resulting in a dataset tailored to the banking sector. Afterwards, three use cases were designed, in which automatic document classification plays a central role in supporting critical processes within a financial institution. For each use case, a specific subset of data was carefully selected in order to reproduce scenarios that closely mirror real banking operations. Each scenario was then explored through two complementary analyses.

The first is an internal analysis that consists of a comparative evaluation of the retrieval strategies integrated into the framework, as described in section 4.3. This study aims to determine how each strategy influences the overall classification performance and to highlight the trade-offs between accuracy and efficiency.

Then an external analysis with the purpose of benchmarking the proposed framework against other zero-shot classifiers, to assess its relative effectiveness in document classification domain. For this purpose, three methods were selected:

- *Natural Language Inference* (NLI), using the *bart-large-mnli* model trained on the Multi-NLI dataset, which infers the most probable class based on the entailment relationship between the text and the class descriptions [77, 78];
- *Embedding Similarity*, which leverages the semantic distance between embedding vectors generated by the all-MiniLM-L6-v2 model [79, 80];
- *LLM Call* approach, which is the main baseline, consists of a straightforward prompt to the LLM (*GPT-3.5-turbo*, *GPT-4o-mini*, and *GPT-4o*), filling the context window with as much content from the document as possible, along with the class descriptions.

These external approaches were selected for representing distinct strategies, from logic-based models (NLI), to purely vector-based methods (Embedding Similarity), to the direct use of LLMs' interpretative capabilities, thereby enabling a comprehensive and meaningful comparison with the developed framework. This experimental setup allows not only the validation of the retrieval strategies employed in the framework, but also a clear view of its advantages and limitations when compared with existing alternatives.

5.1 Global Dataset

The dataset used for the experiments contains 570 publicly available documents, equally distributed across 19 different classes (30 examples per class). These were gathered through simple Google searches (using a *"filetype=pdf"* filter), from company websites hosting mandatory disclosures, and from the online portals of municipalities and local authorities. The goal was to design a dataset that reflects the diversity of information processed and analyzed within the Portuguese banking sector.

Among the financial documents, the **Trial Balance** and **Annual Report and Accounts** stand out, both of which provide a detailed overview of an entity's financial condition and economic activity. These also include other financial statements as well as performance and asset analysis, thus having critical information for risk evaluation, credit granting and business performance monitoring.

In the real estate domain, the dataset includes the **Land and Urban Registry** and the **Land Registry Certificate**, which are two similar essential documents for characterization, identification, and verification of property ownership and history. In addition, the **Energy Certificate** complements this information by assessing the energy efficiency of real estate assets, and is often required in property purchase, sale processes, mortgage loan, among others.

Within the legal domain, relevant documents include, the **Commercial Registry Certificate**, which certifies the incorporation and legal status of companies, and the **Deed**, which formalizes notarial acts such as purchase agreements or donations. Additionally, The **Loan Agreement** and the **Lease Agreement** establish the terms regulating credit operations and property use, respectively, detailing rights, obligations, and guarantees of the parties involved. Within the same scope, the **Legal Opinion** provides critical interpretative guidance, offering technical analyses of contracts, cases, or issues from a legal perspective, thereby supporting institutional decision-making. Also, the **Corporate Minutes** further contribute by officially documenting company decisions, ensuring transparency and legal enforceability.

The insurance category is comprised of a variety of policies, including **Auto**, **Health**, **Work**, **Life**, and **Multi-risk** insurance policies.

The dataset also incorporates two different reports: the **Property Valuation Report**, which estimates property values, and the **Risk Report**, that analyzes and classifies risk factors associated with operations or investments.

Lastly, an **Other** category that brings together documents that either resemble the previously described types or exhibit unique characteristics, thus enabling the evaluation of the framework's capacity to distinguish documents that don't belong to any predefined class by placing them in a distinct "*third category*".

A year of work experience in the banking sector provided the foundation for constructing this dataset, allowing the integration of a consistent perspective on the document types used across different departments within the industry. The outcome is a dataset that, beyond reflecting this functional diversity, also covers multiple formats (born-digital, scanned, and handwritten) as well

as other nuances, such as the high degree of similarity between documents belonging to different classes.

As illustrated in Figure 5.1, the dataset includes documents ranging from very concise, single-page files to extremely long ones, some exceeding one thousand pages (with a maximum of 1112 pages). On average, each document contains 40 pages, with a median of 12, as also illustrated in the boxplot. The first and third quartiles (5 and 27 pages, respectively) indicate that the majority of documents fall within this range. The first and second thirds (7 and 21 pages, respectively) provide additional detail on the overall distribution.

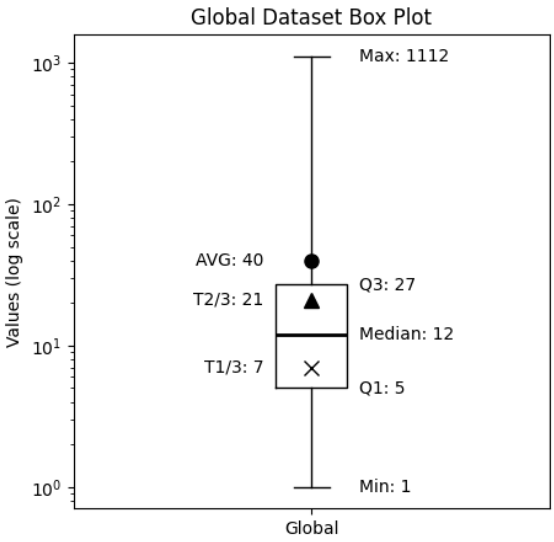

Figure 5.1: Illustration of the variability and dispersion of the global dataset in terms of size (number of pages).

The statistical examination of the *boxplots* by document class, presented in Figure 5.2, reveals pronounced heterogeneity across the dataset. Some classes, like the *Land and Urban Registry*, are composed of short and highly standardized documents, with the majority limited to only two pages. On the other hand, the *Annual Reports and Accounts* are remarkable for their scale, showing a median length of 300 pages and instances surpassing one thousand pages, reflecting the detailed and comprehensive nature of these reports.

Insurance policies, such as *Auto* and *Multi-risk*, also exhibit considerable dispersion, with median lengths above 50 pages and maximums exceeding 100 pages. In contrast, documents like the *Land Registry Certificates*, *Energy Certificates*, and *Lease Agreements* tend to be more concise, with medians below 10 pages.

The differences between medians, maximums, and quartiles observed in the boxplots reveal the presence of outliers, particularly in classes such as *Trial Balance* and *Risk Report*, where very lengthy documents occasionally appear. This variability underscores the importance of flexible and adaptive solutions for automatic classification, able to handle both short, standardized documents and exceptionally long, complex ones.

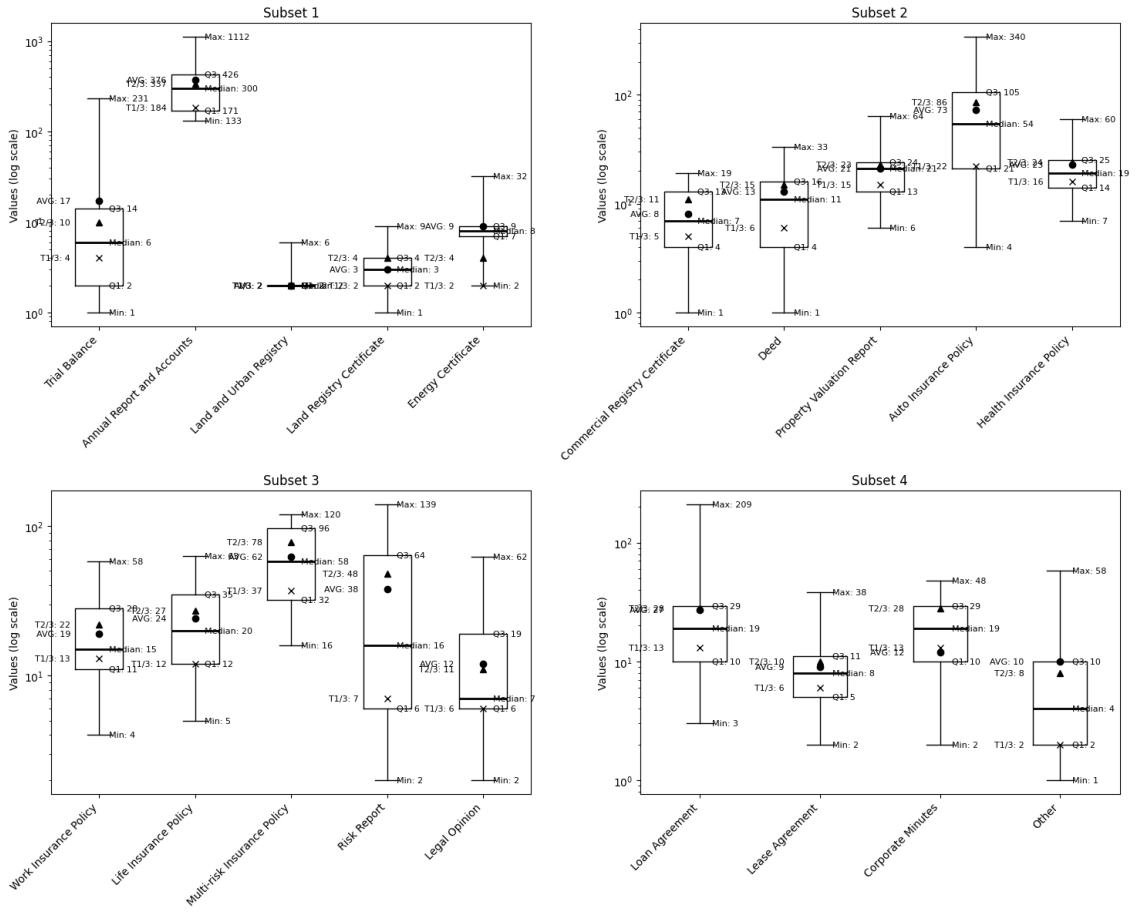

Figure 5.2: Distribution of document length, represented by individual boxplots for each class in the dataset.

5.2 Setup

All experiments involving LLMs were conducted with the *GPT-4o*, *GPT-4o-mini*, and *GPT-3.5-turbo* models, made available by *OpenAI* between 2023 and 2024.

To ensure both consistency and reproducibility, an automated testing pipeline was implemented. This pipeline can handle not only the datasets used in the study but also future ones. Each test case is executed twice, and whenever discrepancies are detected between executions, the pipeline reruns the test five times, adopting as valid the outcome that occurs most often. This approach mitigates the stochastic behavior of language models, as minor variations in output may persist even under zero-temperature configurations.

Regarding the context selection (represented as R in section 4.3 and limited by K) to be included in each model call, an adaptive approach was implemented, based on the statistical distribution of the number of pages in the documents of the given subset. This approach is formalized by the following expression:

In Equation 5.1, R denotes retrieved pages (meaning the ones to be fully processed and sent to the LLM), D_p is the number of pages of the document in scope, $T_{1/3}$ and $T_{2/3}$ are the first and

second thirds of the subset distribution and M is the median.

$$R = \begin{cases} D_p, & \text{if } D_p \leq T_{1/3} \\ \min\left(M,\ \left\lceil T_{1/3} + \dfrac{(D_p - T_{1/3})(M - T_{1/3})}{T_{2/3} - T_{1/3}} \right\rceil\right), & \text{if } T_{1/3} < D_p \leq T_{2/3} \\ T_{2/3}, & \text{if } D_p > T_{2/3} \end{cases} \tag{5.1}$$

This system of equations ensures that short documents are almost totally processed, while long ones are capped at the median threshold, thereby controlling computational cost. For intermediate-length documents, a proportional scaling is applied between the two limits, enabling a balanced representation of content. This strategy prevents biases that could arise from arbitrary truncation and preserves the natural diversity of the documents in the dataset.

To evaluate performance in the multi-label document classification task, the most common metrics were used: *accuracy* (Equation 5.2), *precision* (Equation 5.3), *recall* (Equation 5.4) e *f1-score* (Equation 5.5). Each metric was calculated individually for each class and then macro-averaged, providing an overall assessment that assigns equal importance to all classes, which is appropriate given the balance in the distribution of classes in the dataset.

$$\text{accuracy}_c = \frac{TP_c + TN_c}{TP_c + TN_c + FP_c + FN_c} \qquad \text{macro-accuracy} = \frac{1}{C}\sum_{c=1}^{C} \text{accuracy}_c \tag{5.2}$$

$$\text{precision}_c = \frac{TP_c}{TP_c + FP_c} \qquad \text{macro-precision} = \frac{1}{C}\sum_{c=1}^{C} \text{precision}_c \tag{5.3}$$

$$\text{recall}_c = \frac{TP_c}{TP_c + FN_c} \qquad \text{macro-recall} = \frac{1}{C}\sum_{c=1}^{C} \text{recall}_c \tag{5.4}$$

$$\text{f1-score}_c = 2 \cdot \frac{\text{precision}_c \cdot \text{recall}_c}{\text{precision}_c + \text{recall}_c} \qquad \text{macro-f1-score} = \frac{1}{C}\sum_{c=1}^{C} \text{f1-score}_c \tag{5.5}$$

Notations:

N Number of samples in the dataset.

C Number of classes.

TP_c True positives for class c.

TN_c True negatives for class c.

FP_c False positives for class c.

FN_c False negatives for class c.

accuracy$_c$ accuracy for class c.

precision$_c$ precision for class c.

recall$_c$ recall for class c.

f1-score$_c$ f1-score for class c.

5.3 Experiments by Use Case

To demonstrate the applicability and usefulness of the proposed framework across different banking scenarios, the experiments were organized into three distinct use cases: Document Validation for Business Credit Granting (subsection 5.3.1), Formalization and Legal Compliance of Operations (subsection 5.3.2) and Insurance and Credit Guarantee Document Management (subsection 5.3.3). This approach aims to highlight how the solution can be tailored to concrete business needs, taking into account both data variability and strategic objectives. By structuring the evaluation around these use cases, the study can evaluate the performance of the examined methods under realistic and distinct conditions. This enables a broad assessment of their potential to address diverse challenges in the sector.

5.3.1 Document Validation for Business Credit Granting

In the corporate credit approval process, document validation represents a critical step for risk mitigation and meeting regulatory compliance. The banking process demands rigorous verification of the documents submitted by companies, ensuring that all necessary information is present, up to date, and compliant with both legal and internal standards. In this context, automatic document classification becomes essential to initiate and accelerate the analysis process, reduce the probability of human error, and enhance operational efficiency. By automating this step, banking institutions are able to deliver faster responses to client requests while maintaining high standards of accuracy and compliance, which are vital for the sustainability and security of credit operations.

Dataset

The subset for this use case is composed of financial, commercial, and real estate documents, namely the *Annual Report and Accounts*, *Trial Balance*, *Commercial Registry Certificate*, *Land Registry Certificate*, *Land and Urban Registry*, *Property Valuation Report*, *Loan Agreement*, and *Risk Report*.

Based on this subset, page retrieval methods were applied using the equation in the previous section (Equation 5.1), with the values appropriate to the nature of the subset in question in which: $T_{1/3} = 4$, $T_{2/3} = 17$ e $M = 8$.

The defined Equation 5.6 ensures full processing of short documents (up to four pages), proportional adjustment for medium-length documents (between eight and seventeen pages, up to twenty processed), and a limit of seventeen pages for long documents, as shown in Figure 5.3. Also, the Table 5.1 presents five practical examples of its application.

$$R = \begin{cases} D_p, & \text{if } D_p \leq 4 \\ \min\left(8,\ \left\lceil 4 + \frac{(D_p - 4)(8-4)}{17-4} \right\rceil\right), & \text{if } 4 < D_p \leq 17 \\ 17, & \text{if } D_p > 17 \end{cases} \tag{5.6}$$

Nº of pages in the document	Processed Pages (K)
3	3
6	5
15	7
30	17
1112	17

Table 5.1: Examples of applying the page recovery equation for the use case of document validation in order to obtain the value of K.

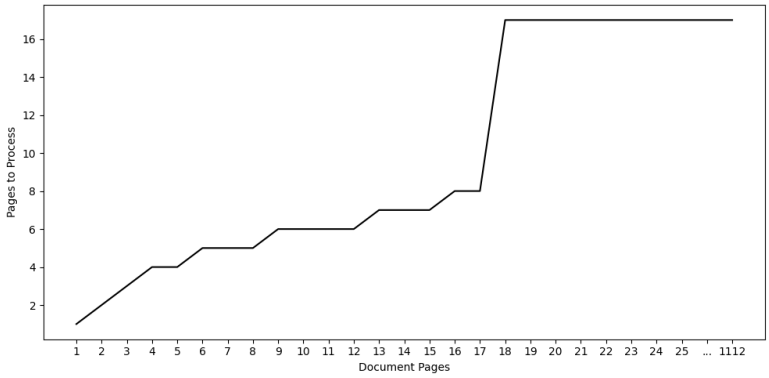

Figure 5.3: Representation of the page recovery equation for document validation, showing the number of pages processed as a function of document length.

Retrieval Strategies Comparison

Overall, when examining Table 5.2, the *RAG* strategy proves to be the most robust and generalizable across the different models. The *Firsts* strategy delivers competitive results and even outperforms *RAG* in the case of *GPT-3.5-turbo*, suggesting that for this specific use case, smaller models may benefit more from simpler inputs than from semantic retrieval. The *Firsts and Lasts* variant achieves results close to the previous strategies, specially in the larger model. Finally, the *TF.IDF* strategy consistently emerges as the weakest, with significant performance drops, particularly in the *GPT-3.5-turbo* model.

Table 5.2: Summary of results for each model and retrieval method in the document validation use case.

		Retrieval Mode			
Model	Metric	RAG	Firsts	Firsts and Lasts	TF.IDF
GPT-4o	accuracy	**99.42%**	99.01%	**99.42%**	99.10%
	precision	**97.51%**	95.61%	97.45%	95.94%
	recall	**97.41%**	95.56%	**97.41%**	95.93%
	f1-score	97.36%	95.49%	**97.39%**	95.75%
***GPT-4o* mini**	accuracy	**98.85%**	98.68%	98.68%	98.27%
	precision	**95.12%**	94.72%	94.71%	92.99%
	recall	**94.82%**	94.08%	94.07%	92.22%
	f1-score	**94.81%**	94.10%	94.10%	92.18%
GPT-3.5-turbo	accuracy	96.17%	**96.54%**	96.05%	94.40%
	precision	**85.60%**	85.27%	84.12%	79.70%
	recall	82.59%	**84.45%**	82.22%	74.82%
	f1-score	82.62%	**83.32%**	81.33%	74.59%

Although the overall analysis provides a good indication of performance, it is also essential to evaluate the framework's performance in more specific dimensions, namely through the analysis

of particular cases.

As illustrated in Figure 5.4, when analyzing a case with a high degree of similarity, between *Land Registry Certificates* and *Land and Urban Registry*, it becomes clear that *GPT-4o* is the most suitable model to distinguish them correctly regardless of the retrieval strategy employed. The same pattern is observed when analyzing the *GPT-4o-mini* line, yet here, the framework consistently fails, misclassifying the same three documents in every circumstances. In the case of *GPT-3.5-turbo*, a substantial performance degradation is observed, with around ten additional errors compared to the fourth-generation models. Although *RAG* slightly reduces the error rate, the remaining strategies fail to capture sufficiently discriminative information. This gap underscores the superior ability of fourth-generation models in capturing deeper semantic nuances, a critical competence when small variations separate documents from different classes which is a common scenario in the banking sector.

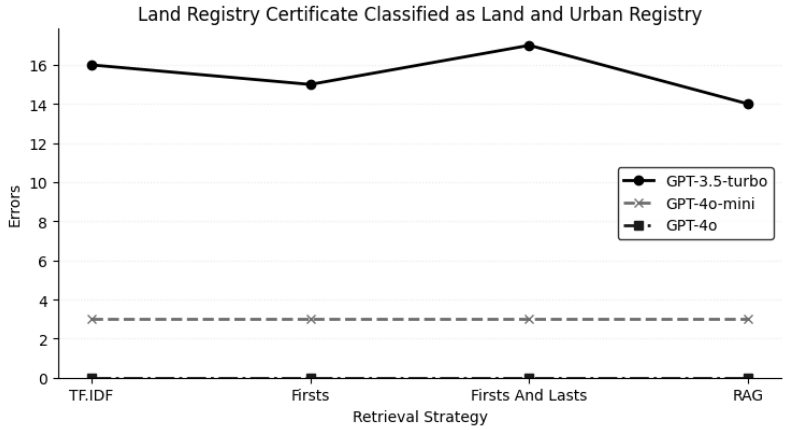

Figure 5.4: Representation of the number of Land Registry Certificates classified as Land and Urban Registry.

Within the residual *Others* category, defined by the exclusion of primary categories, Figure 5.5 shows that the fourth-generation models maintain high rejection rates for out-of-scope documents across all strategies. For *GPT-4o*, the *Firsts and Lasts* strategy stands out as the most effective in this specific dimension, despite not achieving the highest performance overall. *GPT-4o-mini* shows a more balanced behavior across strategies, although *TF.IDF* consistently lags behind. The most severe decline appears with *GPT-3.5-turbo*, where the variation is most pronounced and only the *RAG* strategy mitigates the level of decline, thus maintaining a highly competitive result compared to the other strategies.

Still within the residual *Others* category, but considering the opposite scenario, which is when the framework incorrectly assigns the *Others* label to a document (False Positive), the stacked barplot in Figure 5.6 shows that for stronger models such as *GPT-4o*, the differences between strategies are minimal, with *RAG* emerging as the most consistent option and *Firsts* as the most error-prone. In *GPT-4o*-mini, discrepancies also remain limited, but in *GPT-3.5-turbo* the choice of strategy becomes decisive. The *TF.IDF* strategy underperforms sharply, producing two to three

times more false positives than its counterparts. On the opposite side, *Firsts* strategy, aligned with its strong results in the overall metrics, turns out to be unexpectedly conservative, thus substantially limiting incorrect assignments.

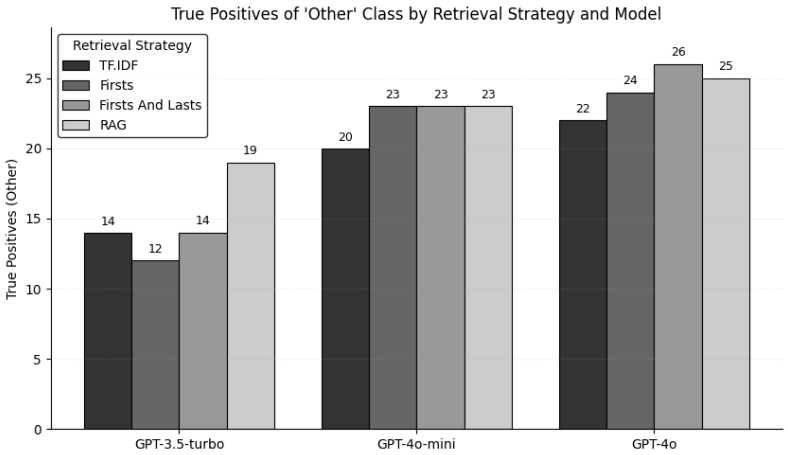

Figure 5.5: Number of true positives in the Other class obtained in different combinations of retrieval strategies and language models for the document validation use case.

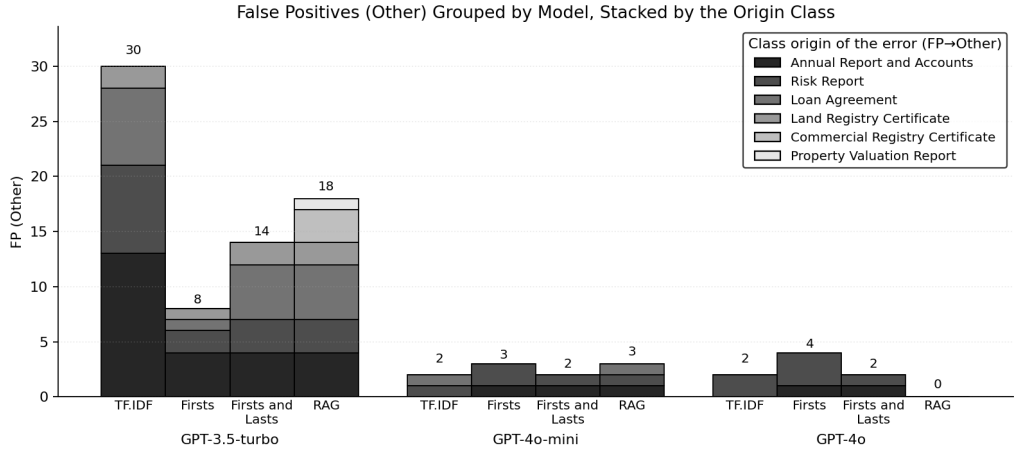

Figure 5.6: Number of false positives in the Other class obtained in different combinations of retrieval strategies and language models for the document validation use case.

Finally, the comparison of token consumption across strategies (Figure 5.7) highlights an interesting balance between cost and performance. *TF.IDF* emerges as the least efficient option, in addition to yielding the weakest performance, it consumes on average 27.7% more tokens than the most economical strategy. By contrast, *RAG* proves to be the most robust in terms of results, with a token cost 11% above the minimum. This trade-off makes it particularly appealing in scenarios where classification quality takes priority. The *Firsts* and *Firsts and Lasts* strategies occupy a middle ground. *Firsts and Lasts* is the most economical reference point, while *Firsts* consumes only 7.8% more. Both demonstrated balanced performance, making them a suitable solution for large-scale contexts where computational efficiency is a significant factor.

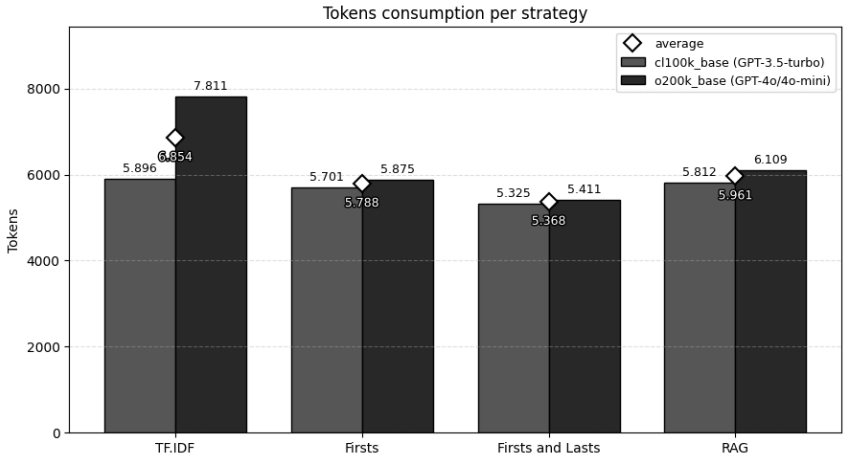

Figure 5.7: Representation of token consumption by retrieval strategy for each token encoder in the document validation use case.

A noteworthy observation that emerges from this analysis, is that instead of the expected reduction in token consumption when moving from *cl100k_base* (the encoder used by *GPT-3.5-turbo*) to *o200k_base* (the encoder of fourth-generation models), the opposite occurs. The explanation lies in the framework's safety mechanism (described in section 4.4), which truncates part of the context once the model limit is reached. The difference is particularly pronounced in *GPT-3.5-turbo*, which supports a maximum context window of 16k tokens, compared to almost 125k tokens in the fourth-generation models.

Even though the previous analysis provides useful insights into costs and consumption, the varying prices across models require a direct calculation of expenses. Accordingly, Figure 5.8 shows the evolution of prices in USD, based on the reference values listed in Table A.2.

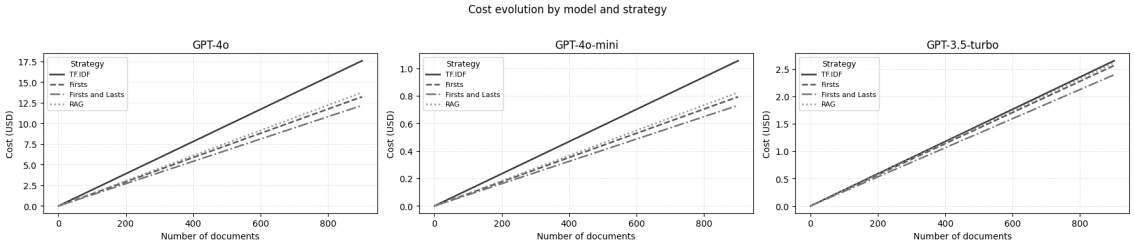

Figure 5.8: Representation of price trends for different combinations of model and strategy in the document validation use case.

It can be observed that *GPT-4o* stands out as the most expensive model, thus, showing a higher cost escalation. Employing this model and the *TF.IDF* strategy (the one that consumes the most tokens) leads to the most expensive combination, with prices rising to just over 17$ for approximately 900 documents. The remaining retrieval strategies, grow at a slower pace, ranging between 11$ and 14$. On the other hand, *GPT-4o-mini*, proves to be highly cost-efficient, with expenses staying below 1$, saving roughly 95% in comparison to *GPT-4o*. Finally, GPT-3.5-turbo is at an intermediate level, ranging from 2.3$ to 2.7$, with minimal variations between strategies.

Benchmarking Against Other Zero-Shot Classifiers

The analysis of Table 5.3 shows that the comparison between the proposed framework and the different zero-shot approaches reveals a consistent trend. The *NLI* approach is practically useless in this setting (*f1-score* of 19.2%), and while *Embedding Similarity* performs somewhat better (*f1-score* of 53%), it remains far below the other alternatives. The strategy of filling the entire context window (*LLM Call*) becomes very competitive employed with state-of-the-art models, particularly *GPT-4o* and its *mini* variant. In these cases, performance is very high (*accuracy* above 98% and *f1-score* above 93%), yet still doesn't surpass the proposed framework, which consistently scores 1 to 2 percentage points higher in *f1-score*. This demonstrates that the additional structuring provided by the framework pays off, even when the underlying model is already highly capable.

Table 5.3: Results summary for the different zero-shot classifiers in the document validation use case.

			Simple LLM Call		
Metric	NLI	Embedding Similarity	*GPT-4o*	*GPT-4o*-mini	GPT-3.5-turbo
accuracy	83.38%	90.86%	99.18%	98.60%	93.75%
precision	26.97%	69.19%	96.28%	94.25%	83.18%
recall	25.18%	58.89%	96.30%	93.70%	71.85%
f1-score	19.24%	53.00%	96.24%	93.73%	73.50%

This scenario shifts significantly with the use of *GPT-3.5-turbo*. In this case, direct calls to the model degrade performance substantially, with an *f1-score* of 73.5%, while the framework remains at a much more competitive level (around 82–83%). This highlights that solution robustness is shaped not only by the capacity of the LLM but also by the way its reasoning process is guided.

The analysis of the *Land Registry Certificate* scenario, classified under *Land and Urban Registry*, further reinforces this interpretation. For highly similar documents, the framework proved to be more reliable, preventing errors that even state-of-the-art models (*GPT-4o* and *mini*) could not avoid under the *LLM Call* strategy. Moreover, with the smaller model, this approach produced 19 errors, representing 2 to 5 more than those observed with the framework.

Regarding the *Others* category, for true positives this approach is fairly competitive, reaching outcomes that are not superior but remain closely aligned. In contrast, when examining false positives, the *LLM Call* performed very poorly with the smallest model, producing 16 additional errors than *TF.IDF* and 38 more than *Firsts*, which represent the weakest and strongest strategies of the framework in this context, respectively.

When efficiency is taken into account, the disparity becomes even more evident. Figure 5.9 shows that, while the framework maintains a contained growth in average token consumption (+10.9% between *GPT-3.5-turbo* and *GPT-4o/mini*), the *LLM Call* strategy exhibits a disproportionate increase, rising from 6436 to over 21000 tokens on average. This increase corresponds to +227%, indicating that the solution doesn't scale sustainably. It is worth noting that

in *cl100k_base* (*GPT-3.5-turbo*) setting, the difference relative to the framework was relatively small (+13.3%), but when *o200k_base* (*GPT-4* family) is employed this difference increases to an impressive +234%. Thus, even when performance metrics appear competitive, the associated average cost makes *LLM Call* impractical for production scenarios.

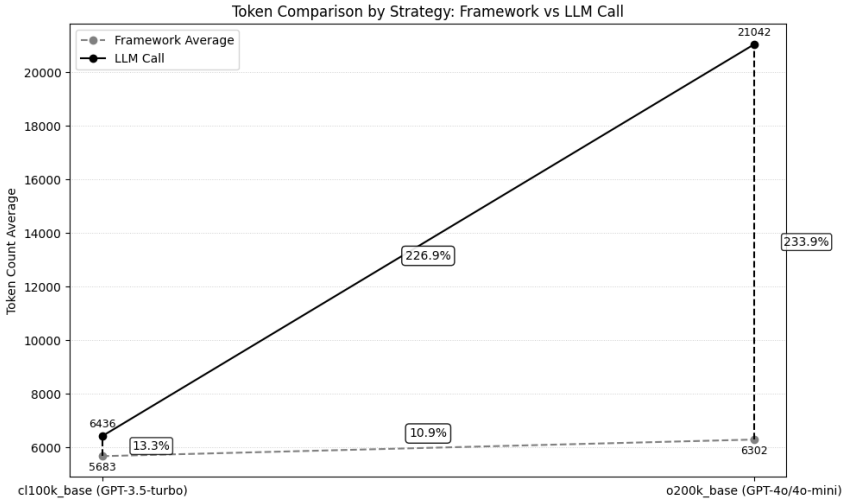

Figure 5.9: Representation of average token consumption for the framework and LLM Call approaches in the document validation use case.

The comparison between the proposed framework and the baseline strategy (*LLM Call*) highlights substantial cost reductions, particularly for the most expensive model, as can be seen in Figure 5.10. For *GPT-4o*, the direct use of the model without tailored context retrieval surpasses the 40$ for 900 documents, while the framework on average, lowers this to near 14$ constituting a saving of more than 65%. A similar proportional reduction is observed with *GPT-4o-mini*, even though with smaller expenses, where the framework keeps costs below 1$ compared to almost 3$ for the baseline. Finally, *GPT-3.5-turbo* shows a slightly different pattern, with the difference between the two approaches being smaller, closely following the token consumption analysis. In this case, the structure offers only a modest improvement, with savings of approximately 15%.

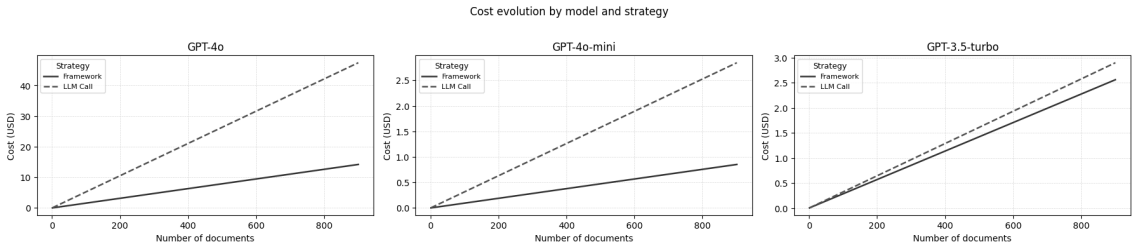

Figure 5.10: Representation of cost evolution for the average framework and LLM Call approaches in the document validation use case.

Discussion

The analysis of the document validation use case for corporate credit approval clearly demonstrates the advantages of the proposed framework over conventional zero-shot approaches. The first dimension to highlight is effectiveness: results show that the framework consistently outperforms some of the baseline alternatives, such as *NLI* and *Embedding Similarity*. When compared with the *LLM Call*, not only matches but in many scenarios even surpasses its performance, particularly in more demanding contexts (such as high degree of similarity) or with smaller models.

The second dimension concerns costs, where the framework proves to be more cost-efficient by effectively balance performance and token consumption. Strategies such as *RAG*, although requiring a moderate increase in resources (+11% compared to the minimum), deliver performance gains that justify the investment. In contrast, the *TF.IDF* strategy is doubly penalizing, with both poorer results and higher consumption (+27.7%). The integration of actual model prices provides a different perspective on costs, with *GPT-4o* exhibiting very steep cost growth and *GPT-4o-mini* proving to be highly efficient, keeping costs much lower even at larger scales. The smaller model, lies in between, showing relatively stable costs with little sensitivity to strategy. Yet, its consistently weaker performance undermines its attractiveness when balancing cost against results. On the other hand, the comparison with the baseline (LLM Call) shows even clearer differences. Despite being competitive in raw metrics, in the most advanced models, the baseline proves unsustainable at scale, consuming up to +227% of tokens and causing significantly higher expenses (+61%) without gains in terms of results. Thus, it can be concluded that the structure offers a much more interesting balance between performance and cost, since it achieves better metrics, consumes significantly fewer tokens, and consequently significantly reduces monetary costs.

In the banking domain, these findings are particularly significant. Credit validation processes demand accuracy, speed, and compliance with legal requirements. The proposed framework accelerates this first stage of validation, improving quality, lowering the risk of human error, and enhancing trust in the results. Coupled with its efficiency in processing costs, the framework is well-suited for production environments, where hundreds of heterogeneous documents must be assessed daily. In this way, the solution not only improves operational efficiency but also supports the sustainability of credit approval workflows, fostering faster and more reliable decisions.

5.3.2 Formalization and Legal Compliance of Operations

The formalization of banking operations, including financing, leasing, corporate restructuring, among others, relies on a rigorous legal assessment of the associated documentation. In this context, banks must guarantee that all necessary documents are correctly provided and compliant with applicable legal standards. Automatic document classification becomes a key tool for legal and compliance teams, allowing them to quickly determine both the nature and the legal scope of the documents received. This automation helps reduce compliance errors, safeguard the legal validity of operations, and foster informed, efficient, and regulation-aligned decision-making.

Dataset

This subset covers legal and corporate documents, such as the *Deed*, *Commercial Registry Certificate*, *Loan* and *Lease Agreement*, *Legal Opinion* and *Corporate Minutes*.

In this use case, page retrieval methods were applied using the equation presented (Equation 5.1), with the values of $T_{1/3} = 6$, $T_{2/3} = 13$ e $M = 8$.

In practice, Equation 5.7 guarantees that short documents (no more than six pages) are fully processed, while medium-length documents (seven to thirteen pages) are scaled proportionally up to a ceiling of thirteen. Longer documents (above thirteen pages) are likewise limited to this threshold, as illustrated in Figure 5.11 and exemplified in Table 5.4.

$$R = \begin{cases} D_p, & \text{if } D_p \leq 6 \\ \min\left(8,\ \left\lceil 6 + \frac{(D_p - 6)(8 - 6)}{13 - 6} \right\rceil\right), & \text{if } 6 < D_p \leq 13 \\ 13, & \text{if } D_p > 13 \end{cases} \tag{5.7}$$

Nº of pages in the document	**Processed Pages** (K)
3	3
8	7
12	8
25	13
209	13

Table 5.4: Examples of applying the page recovery equation for the formalization and legal compliance use case in order to obtain the value of K.

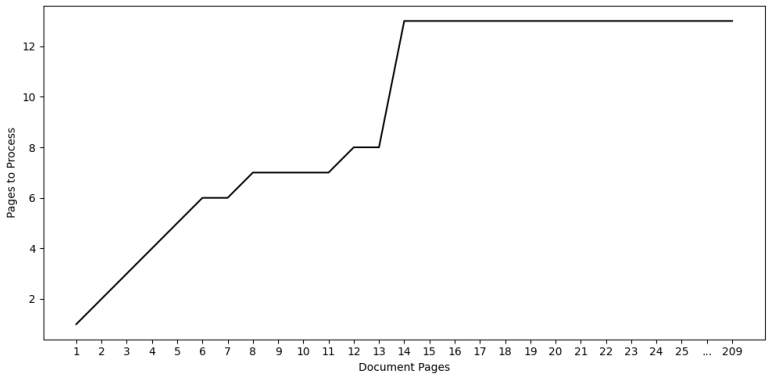

Figure 5.11: Representation of the page recovery equation for the formalization and legal compliance use case, showing the number of pages processed by document length.

Retrieval Strategies Comparison

According to Table 5.5, fourth-generation models, and *GPT-4o* in particular, achieve near-perfect results across all metrics, regardless of the strategy applied. Even so, the superiority of the *RAG* strategy (also observed in section 5.3.1) is confirmed, as it consistently extracts the maximum potential of the models in every scenario. In contrast, *TF.IDF* once again delivers the weakest outcomes, reinforcing its position as the least competitive approach. The *Firsts* and *Firsts and Lasts* strategies fall into an intermediate tier, performing closely to one another and without notable advantages over each other. This trend appears to hold across contexts: with stronger models, differences between strategies become marginal, whereas with smaller models (such as *GPT-3.5-turbo*), the strategy selection gains importance, with *RAG* emerging as the most reliable in maintaining superior performance.

Table 5.5: Table with summarized results for each combination of model and retrieval method in the formalization and legal compliance use case.

Model	Metric	Retrieval Mode			
		RAG	Firsts	Firsts and Lasts	TF.IDF
GPT-4o	accuracy	**99.46%**	99.05%	98.91%	98.50%
	precision	**98.32%**	96.86%	96.48%	95.58%
	recall	**98.10%**	96.67%	96.19%	94.76%
	f1-score	**98.11%**	96.66%	96.21%	94.87%
***GPT-4o* mini**	accuracy	**98.23%**	97.96%	98.10%	97.14%
	precision	**93.95%**	93.37%	93.61%	91.28%
	recall	**93.81%**	92.86%	93.33%	90.00%
	f1-score	**93.73%**	92.64%	93.28%	90.07%
GPT-3.5-turbo	accuracy	**93.88%**	93.47%	93.67%	91.43%
	precision	**83.80%**	81.88%	82.87%	78.21%
	recall	**78.57%**	77.14%	77.62%	70.00%
	f1-score	**78.45%**	76.68%	77.68%	70.17%

Despite the positive overall results, error analysis highlights the legal domain as one of the most challenging, mainly due to the heavy use of technical jargon as well as lexical overlap across document types. Among the most challenging classes are the *Deeds*, which are frequently misclassified as *Corporate Minutes* or *Loan/Lease Agreements*, as might be seen in Figure 5.12, given their shared notarial language and contractual clauses. In addition, *Legal Opinions* also exhibit high confusion rates with *Loan/Lease Agreements* and *Corporate Minutes*, as is evident in Figure 5.12, underscoring their semantic similarity. By contrast, *Corporate Minutes* and *Loan/Lease Contracts* exhibit only minor misclassification patterns.

Analyzing the Figure 5.12, it is interesting to note that once again the use of *GPT-4o* resolves potential issues, proving to be immune to the misclassification of these two types.

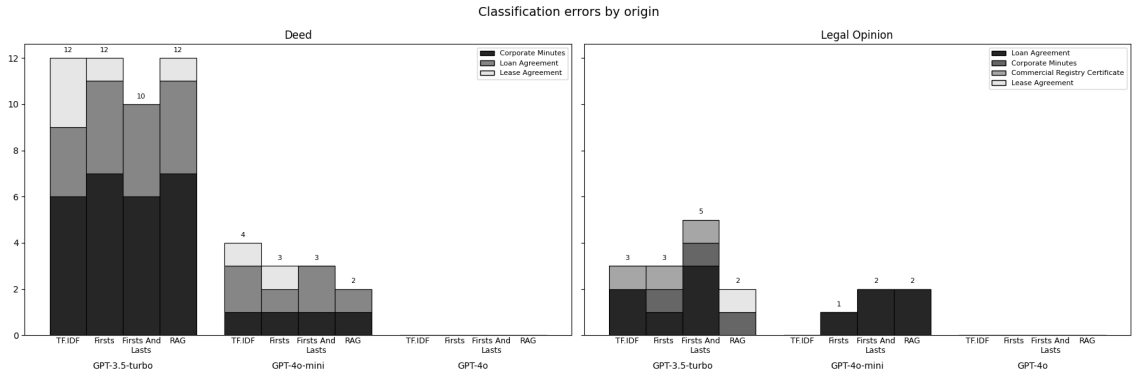

Figure 5.12: Comparison of classification errors by type of document source, highlighting differences between models in cases of Deeds and Legal Opinions.

In the case of *Deeds*, this class proved to be the most problematic for the two alternative models. The smaller model was particularly prone to confusion, repeatedly misclassifying these

documents as *Corporate Minutes*, *Loan Agreements* or *Lease Agreements*. This pattern was consistent across all retrieval strategies, with one exception: the *Firsts and Lasts* strategy, which reduced the confusion to just two categories and completely eliminated misclassifications as *Lease Agreements*, thus making it the best strategy for this task.

In addition, *Legal Opinions* are another recurring source of error, as they describe contractual operations but in an analytical tone, thus inducing misclassifications. Furthermore *Corporate Minutes* show only residual errors but can still be confused with *Deeds* or *Certificates*.

This analysis pattern reflects the difficulty that smaller models have in handling lexical redundancy and formal complexity, which are main characteristics in these document types.

The analysis of rejection cases (*Other* – True Positives) confirms the robustness of the framework, particularly in fourth-generation models, where performance is nearly perfect (Figure 5.13). In the smaller model, however, some degradation is observed, mainly in core document types such as *Deeds* and *Legal Opinions*, which are more frequently misclassified as *Other*, thereby reducing overall reliability.

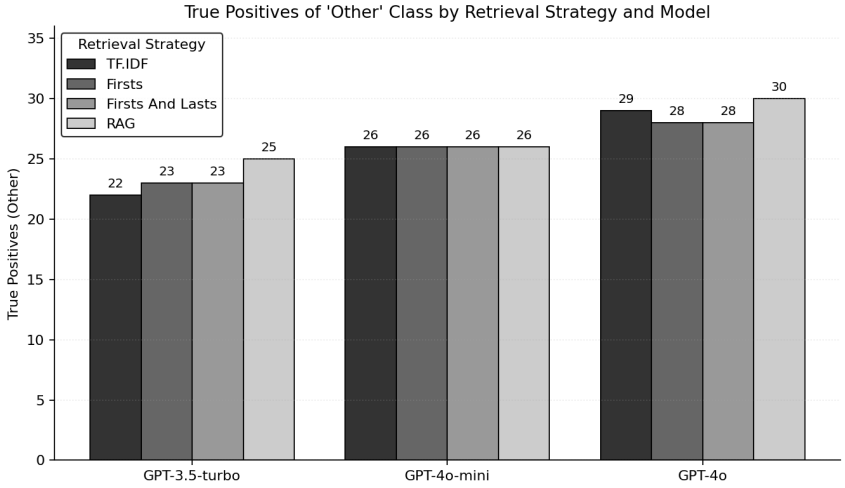

Figure 5.13: Number of true positives in the Other class obtained in different combinations of retrieval strategies and language models for the formalization and legal compliance use case.

The analysis of Figure 5.14 highlights a distinctive phenomenon where the class *Other* effectively works as a "waste bin" for misclassifications, particularly in the less robust model (*GPT-3.5-turbo*). When the framework using this model struggles to confidently assign a specific label, tends to fall back on *Other*, which explains the disproportionately high number of false positives in this category compared with the rest. Consequently, instead of a more balanced distribution, 'Other' results in a high number of false positives, which explains why there are not many more errors than those shown in Figure 5.12 and Figure 5.14. Similarly, *Deeds* and *Legal opinions* are once again the most affected categories, making them the most difficult to distinguish.

In this context, the *Other* category emerges as an absorbing class, capturing a large portion of the misclassifications and thereby preventing the errors from being more evenly distributed across the remaining classes.

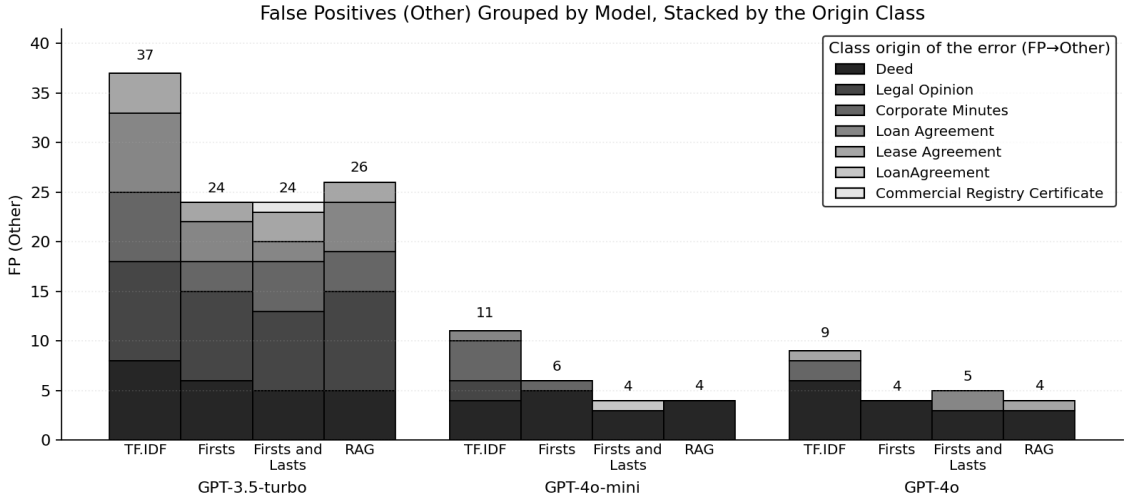

Figure 5.14: Number of false positives in the Other class obtained in different combinations of retrieval strategies and language models for the formalization and legal compliance use case.

Regarding the tokens consumption (Figure 5.15) analysis the *TF.IDF* remains the least attractive option, since it has the highest average consumption (+9% relative to the most economical approach) yet it doesn't offer any performance benefits to offset this overhead. By contrast, RAG positions itself as the most reliable strategy. Its slightly higher cost (+4%) is justified by a clear reduction in errors within critical classes, making it particularly valuable in contexts where reliability is essential. *Firsts and Lasts* emerges as the most economical option, serving as a reference point, while *Firsts* consumes only slightly more (+4.9%). Both deliver strong results, especially with more capable models, making them suitable choices whenever computational cost savings are a priority.

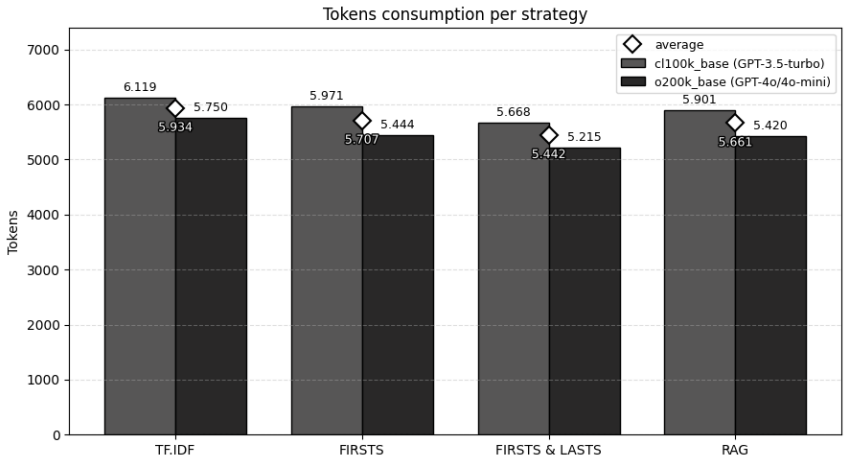

Figure 5.15: Representation of token consumption by retrieval strategy for each token encoder in the formalization and legal compliance use case.

In contrast to the findings in subsection 5.3.1, for this subset the gap between *cl100k_base* and *o200k_base* is minimal and constant, maintaining the expected effect of token savings when using

o200k_base. This occurs because the nature of the documents that compose this subset is smaller (fewer pages), thus preventing the framework's safety mechanism, responsible for truncating part of the context once the model limit is exceeded, from being triggered. As a result, token consumption remains stable across models, and the differences observed between strategies reflect only their inherent characteristics.

Looking at monetary costs, the patterns align closely with the token usage analysis. As illustrated in the Figure 5.16 *GPT-4o* remains the most expensive option, with values reaching nearly 14$ for 900 documents. Unlike the previous subset (section 5.3.1), where the *TF.IDF* strategy was clearly the most expensive, in this scenario the differences between the strategies are relatively small, ranging between 1$ and 2$. Once again, *GPT-4o-mini* stands out as the most economical choice, with residual costs below 1$, making it particularly attractive whenever financial efficiency is a priority. *GPT-3.5-turbo* falls in the middle, at around 2.50$, with minimal variation between strategies. The stability of the different strategies is largely due to the smaller documents in this subset, which limit context truncation and support a nearly linear cost trend. Overall, the pricing analysis reinforces that the choice of model is also a decisive factor, while the framework keeps scalability under control and secures a solid trade-off between performance and cost.

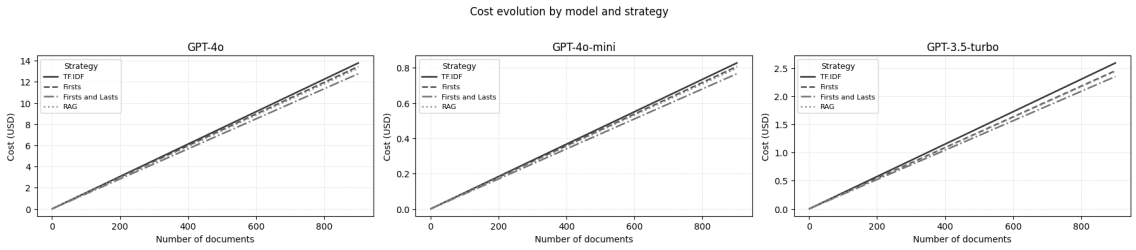

Figure 5.16: Representation of price trends for different combinations of model and strategy in the formalization and legal compliance use case.

Benchmarking Against Other Zero-Shot Classifiers

Table 5.6 confirms the same trend observed in the previous use case (section 5.3.1). The *NLI* method proves unusable, while *Embedding Similarity*, though substantially better, still falls far behind the remaining approaches. Regarding the *LLM Call* strategy, the use of top-tier models narrows the gap with the framework but never surpasses it. Both *GPT-4o* and its *mini* variant achieve very strong results (*f1-score* of 97.6% and 94.1%), yet remain slightly below the framework, which retains an advantage and greater consistency. The sharpest contrast is seen when *GPT-3.5-turbo* is employed, where the direct call collapses to an *f1-score* of 70.8%, compared to framework values ranging from 80–90%. This outcome shows that the competitiveness of *LLM Call* depends heavily on the model generation.

The specific errors make this weakness even more evident. In *Deeds* and *Legal Opinions* which are classes traditionally difficult to classify (as discussed earlier), when the *LLM Call* is used with the smaller model, it tends to redirect documents to the *Others* category, thereby masking

Table 5.6: Results summary for the different zero-shot classifiers in the formalization and legal compliance use case.

Metric	NLI	Embedding Similarity	Simple LLM Call		
			GPT-4o	*GPT-4o*-mini	GPT-3.5-turbo
accuracy	85.71%	87.76%	99.32%	98.37%	90.82%
precision	00.00%	67.30%	97.81%	94.38%	81.11%
recall	00.00%	57.14%	97.62%	94.29%	67.62%
f1-score	00.00%	52.85%	97.64%	94.14%	70.79%

prediction failures. The impact is particularly severe in the *Others* category, where, although this approach achieves competitive results with fourth-generation models, in *GPT-3.5-turbo* false positives rise sharply to 51 errors, exceeding the framework's worst setup by 14 errors and its best by 25, thus far surpassing any framework variant.

Regarding the efficiency, the contrast between strategies once again becomes evident. As show in Figure 5.17, the framework achieves stable average token usage and even a reduction with fourth generation models (–7.7%). In the other hand, *LLM Call* grow from 6438 to 8309 tokens, an increase of +29.1%. The relative difference compared to the framework, which stood at just +8.9% in *cl100k_base* (*GPT-3.5-turbo*), escalates to more than +52% in *o200k_base* (*GPT-4o/mini*).

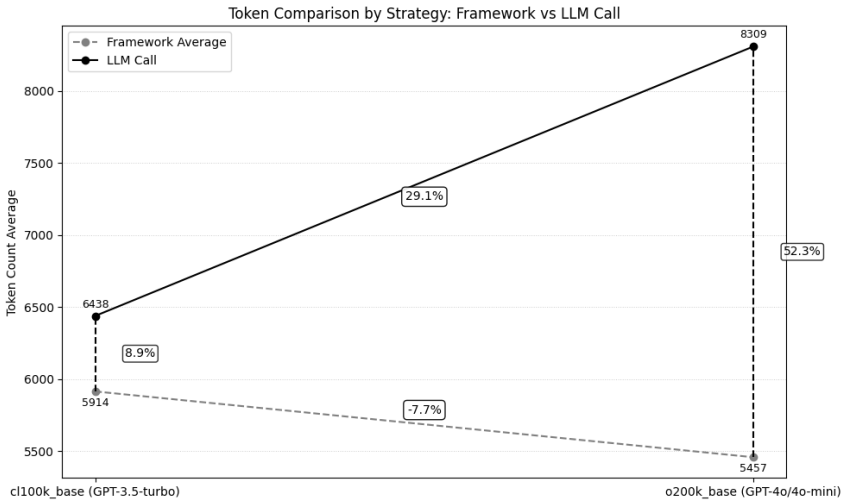

Figure 5.17: Representation of average token consumption for the framework and LLM Call approaches in the formalization and legal compliance use case.

Although this growth is significant, it doesn't reach the magnitude observed in the previous use case (section 5.3.1), where consumption exploded beyond +200%. The explanation lies in the very nature of the legal subset, which is smaller in terms of page count, limiting the escalation of cost. Even so, the trend remains clear: the framework scales in a controlled manner, whereas the *LLM Call* becomes increasingly expensive as model complexity grows.

Within the legal subset, the contrast between the framework and the baseline (LLM Call) again

points to efficiency improvements (Figure 5.18), though less marked than in the previous case (section 5.3.1). For *GPT-4o*, the direct use of the LLM, without any optimization of the context retrieved, reaches around 19$ for the same 900 documents, while the framework on average lowers this value to approximately 14$, therefore saving roughly 30%. A similar proportional reduction is seen with *GPT-4o-mini*, where costs fall from just over 1.1$ to under 0.8$. By contrast, for *GPT-3.5-turbo* the gap between baseline and framework is narrower, as total expenses are lower and the cost curves almost overlap, though the framework still delivers a modest gain.

Another relevant observation is that *GPT-4o* is cheaper here than in the previous subset, a result of the shorter length of legal documents, which avoids pushing the framework's safeguards to the maximum context size and therefore keeps token usage under control.

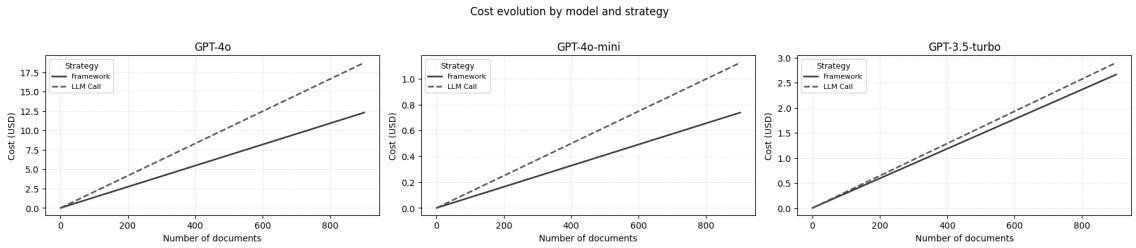

Figure 5.18: Representation of cost evolution for the average framework and LLM Call approaches in the formalization and legal compliance use case.

Discussion

In the legal domain, the analysis confirms that the proposed framework offers clear advantages in both effectiveness and cost-efficiency, consistently outperforming the evaluated baselines. These results align with the findings from the previous use case (section 5.3.1), reinforcing the framework's ability to generalize across domains. While *LLM Call* proves competitive when applied to state-of-the-art models, its performance is compromised by lower reliability in critical classes and by substantially higher resource consumption. By contrast, the framework combines accuracy and efficiency, establishing itself as a more robust and sustainable solution.

Regarding costs, in the legal subset, the framework demonstrates again greater efficiency by keeping token consumption stable and predictable, unlike the *LLM Call* strategy, where costs scale with more advanced models. Although the increase is less pronounced than in the credit subset, it remains significant, with percentages around +52% for *GPT-4 family* compared to the framework, versus +8.9% with *GPT-3.5-turbo*. Among the internal strategies, *RAG* stands out for its balance between performance and cost, justifying a small increase in consumption, while *Firsts and Lasts* remains the most economical option. In terms of pricing, the same trends are confirmed, with *GPT-4o* as the most expensive model, *GPT-4o-mini* as the most efficient, and *GPT-3.5-turbo* in an intermediate position. Compared to the baseline, it is clear that the framework ensures lower costs and, at the same time, better performance, offering a much more favorable trade-off between performance and cost.

Applied to the use case of legal formalization and compliance, these findings have clear practical significance. Legal and compliance teams depend on the timely and accurate classification of contracts, deeds, and legal opinions in order to verify their compliance with regulatory and corporate standards. The framework reduces compliance risks, accelerates document screening, and enhances process consistency, all of which are crucial for regulatory assurance. In addition, efficient cost management ensures scalability, even in environments where thousands of documents require daily review.

5.3.3 Insurance and Credit Guarantee Document Management

Managing insurance and credit guarantee documentation is a key element in the banking sector's risk mitigation strategy. The ability to quickly and accurately identify and validate such documents is essential to confirm that the guarantees and coverages tied to credit operations are in place. Automatic document classification enables a faster analysis of relevant records, supporting the operationalization of banking products and the efficient management of guarantees. This automated process not only increases response times and processing capacity but also strengthens the robustness and reliability of risk assessment.

Dataset

This final subset consists of documents related to insurance and evaluations, namely the *Auto*, *Health*, *Work*, *Life* and *Multi-risk* Insurance Policies, *Property Valuation Report*, and *Risk Report*. The length of these documents ranges from 1 to 340 pages, with an average of 31 pages and a median of 18. The first third contains documents of up to 11 pages, the second third up to 27 pages, while the first quartile is at 8 pages and the third quartile at 36 pages. As illustrated in Figure 5.2, the subset includes a significant number of short documents, but also a considerable proportion of medium- to large-sized ones.

The page retrieval methods were applied using the equation presented (Equation 5.1), with the values of $T_{1/3} = 11$, $T_{2/3} = 27$ e $M = 18$.

Operationally, Equation 5.8 guarantees that concise documents (no more than eleven pages) are fully processed, documents of intermediate length (from twelve to twenty-seven pages) are scaled proportionally up to that limit, and longer documents (above twenty-seven pages) are capped at the same threshold. The rule is illustrated in Figure 5.19 and exemplified in Table 5.7.

$$R = \begin{cases} D_p, & \text{if } D_p \leq 11 \\ \min\left(18,\ \left\lceil 11 + \dfrac{(D_p - 11)(18 - 11)}{27 - 11} \right\rceil\right), & \text{if } 11 < D_p \leq 27 \\ 27, & \text{if } D_p > 27 \end{cases} \tag{5.8}$$

Nº of pages in the document	Processed Pages (K)
7	7
15	13
22	16
40	27
340	27

Table 5.7: Examples of applying the page recovery equation for the insurance use case in order to obtain the value of K.

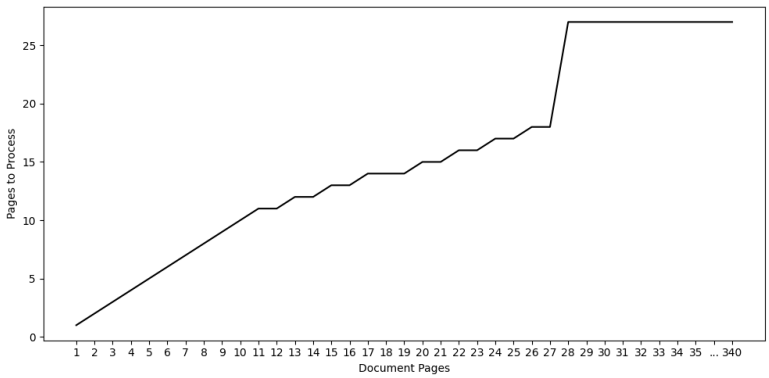

Figure 5.19: Representation of the page recovery equation for the insurance use case, showing the number of pages processed by document length.

Retrieval Strategies Comparison

In the use case of Insurance and Credit Guarantee Document Management, the framework achieves outstanding results in nearly all scenarios (Table 5.8). Fourth-generation models, particularly *GPT-4o*, deliver almost perfect performance (accuracy close to 99.6%, f1-score around 98.5%), with the *RAG* and *Firsts and Lasts* strategies sharing the lead. Although *TF.IDF* is systematically the weakest strategy (as already observed in section 5.3.1 and section 5.3.2), its scores remain high, suggesting that this domain is, overall, less complex than others previously analyzed. *GPT-4o-mini* shows greater sensitivity to retrieval strategy, with *RAG* emerging as the strongest and *TF.IDF* suffering a significant decline. Even with *GPT-3.5-turbo*, the most limited of the models, performance remains high (*f1-score* up to 91% across all strategies), reinforcing the idea that insurance-related classes display clearer structural and semantic distinctions compared to other domains, such as financial reports.

The per classes analysis reinforces these findings. A key observation relates to *Insurance Policies*, which are theoretically challenging to separate because of structural resemblance, overlapping technical terminology, and subtle distinctions. Despite this, the framework achieved near-perfect performance, even in the weaker models, with only residual misclassifications. This result is especially noteworthy as it demonstrates that the integration of retrieval strategies with LLMs effectively addresses semantically and structurally similar documents, surpassing the constraints commonly faced by traditional classification approaches.

A second noteworthy case involves the *Risk Reports*, which reveal an unexpected pattern. Though it was to be expected that the fourth-generation models would clearly stand out, *GPT-4o-mini*, for the first time, recorded lower performance than *GPT-3.5-turbo* in certain scenarios (Figure 5.20). The explanation appears to lie in the lexical overlap between *Risk Reports* and *Insurance Policies*, which leads the intermediate model to semantic confusions, spreading errors across several incorrect classes. Paradoxically, *GPT-3.5-turbo* showed greater caution, often redirecting these documents to the *Other* class, thus reducing direct confusions but at the cost of a

Table 5.8: Table with summarized results for each combination of model and retrieval method.

Model	Metric	Retrieval Mode			
		RAG	Firsts	Firsts and Lasts	TF.IDF
GPT-4o	accuracy	**99.67%**	99.59%	**99.67%**	99.51%
	precision	98.59%	98.41%	**98.69%**	97.96%
	recall	**98.52%**	98.15%	**98.52%**	97.78%
	f1-score	**98.53%**	98.16%	**98.53%**	97.74%
***GPT-4o* mini**	accuracy	**99.34%**	99.01%	98.60%	99.01%
	precision	**97.24%**	95.70%	94.50%	95.98%
	recall	**97.04%**	95.56%	93.70%	95.56%
	f1-score	**96.97%**	95.36%	93.30%	95.32%
GPT-3.5-turbo	accuracy	**98.77%**	98.56%	98.44%	98.15%
	precision	**95.32%**	94.82%	94.45%	93.22%
	recall	**94.44%**	93.33%	92.96%	91.48%
	f1-score	**94.52%**	93.67%	93.15%	91.95%

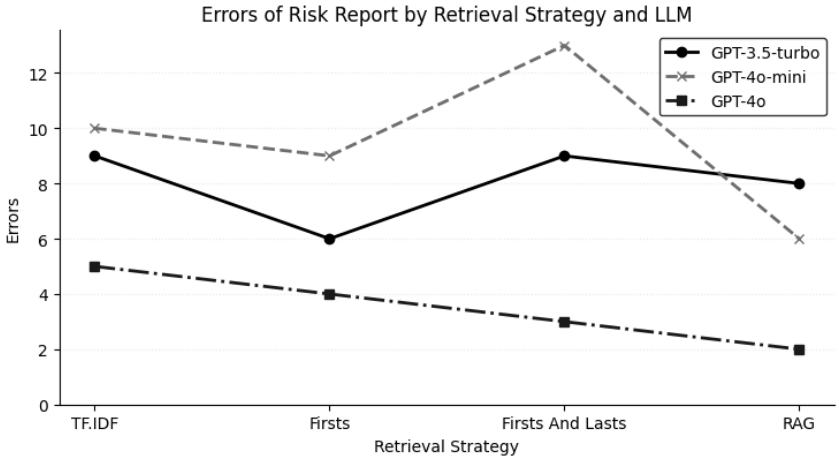

Figure 5.20: Representation of the number of errors within the risk report class.

higher rejection rate. Only the RAG strategy maintained, even by a small margin, the expected hierarchy of performance between models, indicating that its use played a differentiating role in *GPT-4o-mini*. With greater contextual reasoning capacity, *GPT-4o* is able to minimize these ambiguities and stands out clearly from the rest.

For the *Energy Certificates*, the fourth generation models performance was perfect, without any single error. By contrast, *GPT-3.5-turbo* produced a significant number of errors, ranging from 2 to 7 depending on the strategy. In this case, *RAG* once again demonstrated its robustness, reducing errors by about 70% compared to the *Firsts* strategy, which proved to be the weakest. This example highlights how strategy selection becomes essential in less capable models.

When examining the rejection of out-of-scope documents (*Other* – True Positives), *GPT-4o* delivers almost flawless results across all strategies (Figure 5.21). For *GPT-4o-mini* and *GPT-3.5-turbo*, *RAG* again emerges as the most reliable option, maintaining high performance. Yet,

the False Positives analysis shows that *Risk Reports* remain the most error-prone class, regardless of model or strategy (Figure 5.22). Within this setting, *RAG* proves to be the most consistent approach, whereas *Firsts and Lasts* and *TF.IDF* suffer notable drops in performance in both intermediate and smaller models.

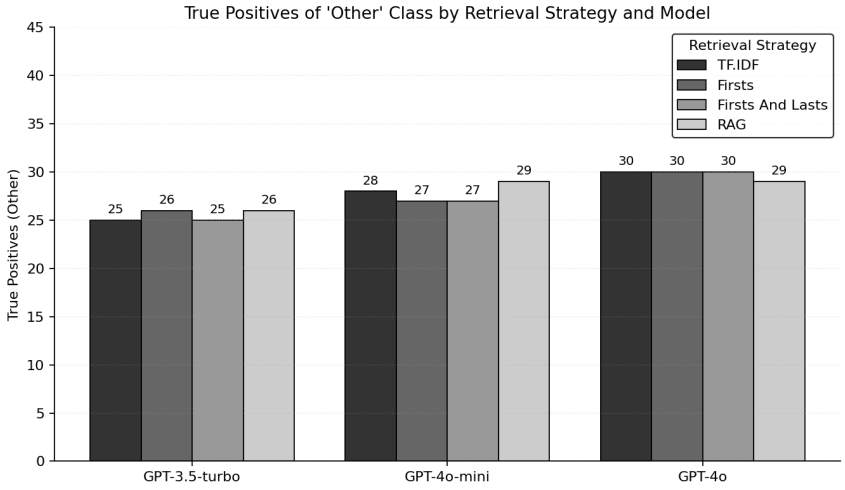

Figure 5.21: Number of true positives in the Other class obtained in different combinations of retrieval strategies and language models for the insurance use case.

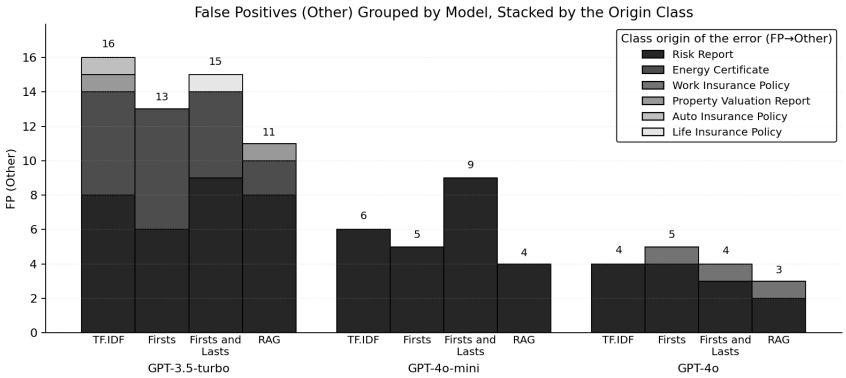

Figure 5.22: Number of false positives in the Other class obtained in different combinations of retrieval strategies and language models for the insurance use case

The token consumption analysis (Figure 5.23) highlights significant differences in efficiency. As in previous cases, the *TF.IDF* strategy again proves the least economical, consuming on average 6.4% more than the lightest option, without any corresponding performance gain. *RAG*, while requiring a modest additional cost (+3.3%), remains competitive thanks to its overall consistency. The *Firsts* and *Firsts and Lasts* strategies confirm themselves as the most efficient, repeating the pattern observed in other use cases. *Firsts and Lasts* sets the lowest reference value, while *Firsts* shows only a residual increase (+1.1%). These two strategies therefore emerge as the most suitable for large-scale scenarios, where controlling computational costs is critical.

It should be highlighted that, as in section 5.3.1, the framework's safety mechanism has an im-

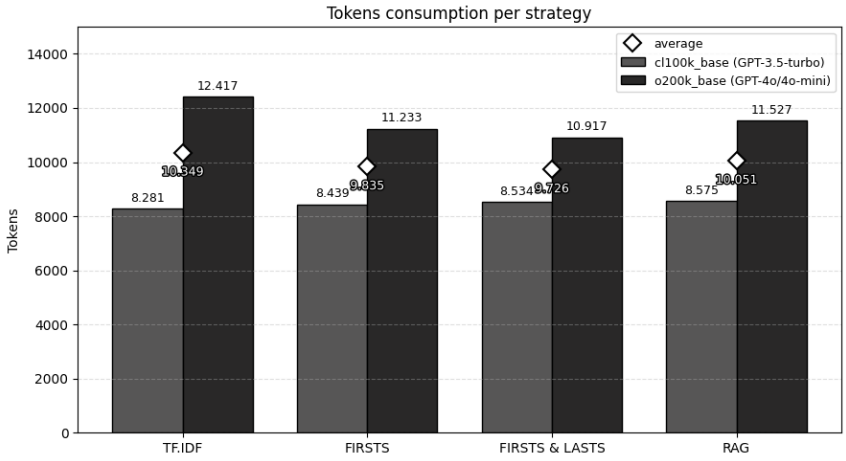

Figure 5.23: Representation of token consumption by retrieval strategy for each token encoder in the insurance use case.

pact by truncating part of the context in *cl100k_base* whenever the limit is exceeded. This explains why, contrary to expectations, consumption is more contained in this model. In *o200k_base*, however, this restriction doesn't apply, resulting in higher values that more closely reflect the actual consumption of the strategies.

In the insurance subset, the cost differences between strategies are quite small, especially when compared to the credit (section 5.3.1) and legal (section 5.3.2) subsets. As observed in Figure 5.24, *GPT-4o* continues to be the most costly option, reaching nearly 26$ for 900 documents, whereas *GPT-4o-mini* once again proves to be the most economical, never exceeding 1.7$. *GPT-3.5-turbo* takes an intermediate position, with expenses of about 3.8$. Yet, despite the absolute gaps between models, the four strategies produce almost identical cost curves, with only minor deviations. This indicates that, in this subset, document characteristics yield an almost homogeneous effect of the strategies on pricing, leaving model choice as the decisive factor for this component.

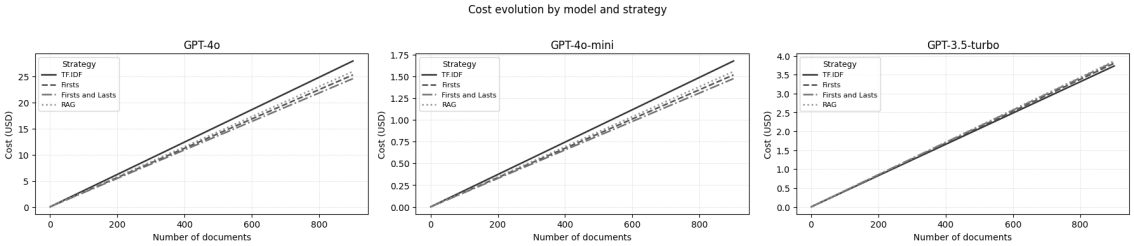

Figure 5.24: Representation of price trends for different combinations of model and strategy in the insurance use case.

Benchmarking Against Other Zero-Shot Classifiers

This use case confirms the patterns seen in previous domains (section 5.3.1 and section 5.3.2), where direct LLM calls are only competitive when supported by top-tier models. *GPT-4o* and its mini variant achieve high metrics (*f1-scores* of 97.5% and 96.2%), close to the framework, but

Table 5.9: Results summary for the different zero-shot classifiers in the insurance use case.

Metric	NLI	Embedding Similarity	Simple LLM Call		
			GPT-4o	*GPT-4o*-mini	GPT-3.5-turbo
accuracy	88.89%	88.48%	99.42%	99.18%	94.61%
precision	00.00%	66.23%	97.90%	96.56%	88.73%
recall	00.00%	48.15%	97.41%	96.30%	75.56%
f1-score	00.00%	45.25%	97.45%	96.22%	78.37%

never surpassing it. The scenario changes completely with *GPT-3.5-turbo*, where performance drops to an *f1-score* of 78.4%, showing that the strategy is not resilient with less powerful models. The remaining external baselines offer no viable alternative: *Embedding Similarity* reaches only 45.2% *f1-score*, and *NLI* fails entirely (*f1-score* = 0%).

A detailed error analysis reveals some weaknesses, however, in core insurance documents the framework maintains stable classifications with very few mistakes, even in *GPT-3.5-turbo*. By contrast, *LLM Call* accumulates 17 errors in this configuration, compared with only two in *GPT-4o* and one in the mini.

For the risk report documents, *LLM Call* produced 5 errors with *GPT-4o* and 7 with *GPT-4o-mini*, reflecting the pattern seen with the strongest strategy of the framework, *RAG*. However, when the smallest model is employed, the errors rise to 19, far exceeding any framework configuration.

In the *Others* category, both the framework and *LLM Call* achieve similar results in true positives. Yet in false positives, *GPT-3.5-turbo* under *LLM Call* collapses once again, with 50 errors compared to only 11–16 in the framework.

The comparison with the framework makes this conclusion even clearer (Figure 5.25). While the *GPT-3.5-turbo* model had almost identical token usage (+0.8%), the gap expanded sharply to +66.1% in the *GPT-4* family, clearly indicating that the strategy is failing to scale sustainably. This increase, although less dramatic than in the case of credit use (where it exceeded +200%, section 5.3.1) but slightly higher than in the legal use case (the lower, with +52%, section 5.3.3), confirms the recurring pattern in which the structure preserves stability and the *LLM Call* becomes increasingly expensive. This discrepancy is also partly explained by the insurance subset itself, which is less extensive and less dependent on contextual depth than credit data, thereby reducing the overall cost impact.

In the insurance subset, the comparison between the framework and the baseline (LLM Call) again shows significant differences, especially in the more expensive models (Figure 5.26). In *GPT-4o*, the baseline cost exceeds 40$ for 900 documents, while the framework reduces this amount to around 26$, ensuring savings of close to 40%. The same pattern is observed in *GPT-4o-mini*, where costs drop from over 2.5$ to approximately 1.6$. In *GPT-3.5-turbo*, however, the differences are marginal, with both scenarios hovering around 3.8$, reflecting a limited impact of the framework on this model. In short, this subset also confirms that the framework enables

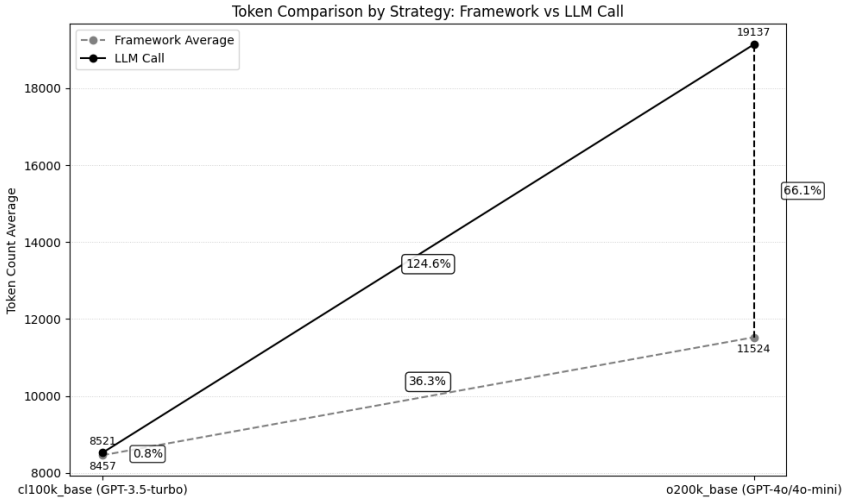

Figure 5.25: Representation of average token consumption for the framework and LLM Call approaches in the insurance use case.

significant efficiency gains when applied to more expensive models, mitigating the baseline cost escalation.

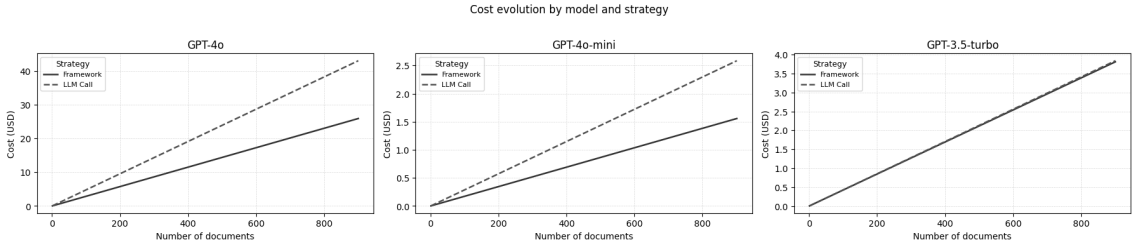

Figure 5.26: Representation of cost evolution for the average framework and LLM Call approaches in the insurance use case.

Discussion

In the domain of insurance and credit guarantees, the results confirm the high effectiveness and efficiency of the framework, following the same pattern already observed in the credit and legal use cases. In terms of performance, it achieved near-perfect metrics with fourth-generation models (*GPT-4o* and *mini*), reaching *accuracy* above 99% and *f1-scores* above 98%. Even with *GPT-3.5-turbo*, a more limited model, performance remained robust (*f1-score* up 91% across all strategies), showing that this subset is structurally more distinct than other use cases, which facilitates automatic classification. Typically complex situations, such as distinguishing between policies from different branches, were handled correctly by the framework.

From a cost-efficiency perspective, the framework proves highly competitive. *RAG* remains the most robust strategy, justifying its slight increase in tokens consumption (+3.3%) with a substantial reduction in errors for classes such as Energy Certificates. The *Firsts* and *Firsts and Lasts* strategies stand out as the most efficient, combining minimal consumption with solid performance,

whereas *TF.IDF* once again emerges as the least attractive option, with higher usage (+6.4%) and no relevant performance gains. When actual prices are considered, the differences across strategies are marginal in this subset, making model selection the decisive factor. *GPT-4o* appears as the most expensive (around 26$), *GPT-4o-mini* remains the cheapest (close to 1.6$), and *GPT-3.5-turbo* sits in between (approximately 3.8$). Compared with the baseline (*LLM Call*), the contrast becomes even clearer. Although it also achieves good metrics with the more advanced models, its costs scale unsustainable, surpassing 40$ for *GPT-4o*, versus only 26$ with the framework. Thus, even in the insurance subset, the framework confirms a much more favorable trade-off between performance and cost, delivering consistent efficiency gains.

When applied to a practical banking use case, the advantages are immediate. As in the credit validation (section 5.3.1) and legal compliance scenarios (section 5.3.2), the framework ensures fast, reliable, and scalable processing which is a critical step for mitigating risks and ensuring the validity of credit operations in the management of insurance and guarantee documents. Beyond reducing errors that could compromise coverage or the activation of guarantees, the framework also controls computational costs predictably, making it suitable for large-scale production environments where thousands of policies, reports, and certificates must be processed daily.

5.4 Cross-Use Case Analysis

5.4.1 Global Metrics

The global assessment of metrics highlights a consistent trend across all three use cases. The *GPT-4* family stands out, with *GPT-4o* consistently reaching *accuracy* and *f1-scores* above 98%. *GPT-4o-mini* delivers comparable results, with only minor declines, whereas *GPT-3.5-turbo* reveals evident weaknesses in more demanding contexts, especially when classes share strong semantic proximity.

With respect to retrieval strategies, *RAG* remains the most robust solution across all contexts, offering the best balance between performance and computational cost. *Firsts and Lasts* emerges as the most economical alternative, delivering competitive results and proving particularly attractive in large-scale scenarios. *TF.IDF*, by contrast, confirms its weakness, combining lower metrics with higher consumption.

When compared with zero-shot baselines such as *NLI*, *Embedding Similarity*, or direct *LLM calls*, the framework is consistently better. This advantage is most evident over *NLI* and *Embedding Similarity*, while in direct *LLM calls* the difference translates into modest but steady gains with top-tier models. In lower-capacity models, however, the gap becomes substantial, reaching more than ten percentage points in *f1-score*.

Finally, when efficiency is taken into account, the framework proves more sustainable as token consumption grows in a predictable and controlled way, in contrast to the sharp escalation seen with *LLM Call*, which more than doubles the average cost. This behavior reinforces the practical applicability of the solution in production scenarios.

5.4.2 Shared Classes Analysis

The cross-class analysis of documents shared between subsets makes it possible to understand how much context influences model performance. The purpose of this analysis is therefore to assess how the same class may be easy in one scenario yet challenging in another, depending on the semantic proximity of its neighboring documents.

In the case of the *Commercial Registry Certificate*, in the credit subset it appeared alongside reports and risk analyses (section 5.3.1), which overlap less in jargon. In the legal subset (section 5.3.2), however, it was placed among deeds and minutes, where the terminological density is much higher. In terms of performance, the framework showed a higher error rate in the credit subset, ranging from 0 to 2 errors with *GPT-4o-mini* and 0 to 3 with *GPT-3.5-turbo*. In contrast, in the legal subset, errors were minimal, with only a single misclassification (using the smaller model) observed across all runs. Interestingly, this pattern is reversed for the *LLM Call* strategy, where the credit subset contained 3 errors and in the legal subset raised to 5, making it the least favorable scenario. Notably, *GPT-4o* proved entirely robust across all strategies, achieving flawless performance regardless of context. In other words, context influenced not only the difficulty of the task but also the relative advantage of each approach.

The *Loan Agreement* follows a distinct pattern. In the credit subset (section 5.3.1), the distinction from financial reports was relatively straightforward, with only residual errors in the top models, both within the framework and through direct LLM Call. By contrast, *GPT-3.5-turbo* proved to be less reliable, yielding between 1 and 7 errors in the framework and consistently 7 in *LLM Call*. In the legal subset (section 5.3.2), however, the presence of leases and minutes introduced additional ambiguity. Some combinations in the framework obtained three errors when using *GPT-4o*, whereas in the credit subset performance had been flawless. The smaller model fared worse, with errors ranging from 2 to 9, marking a clear decline compared with the previous case. For the *LLM Call*, unlike what happened with the previous document, the pattern remains the same, where the framework obtains the same number of errors for the larger models, and 2 additional errors for the smaller model. This case illustrates how the density of semantically similar documents amplifies the difficulty of the task.

Through these two comparisons, we can see how the same contexts (credit and legal) have different impacts on different documents.

In addition, the *Property Valuation Report* stood out as the class most immune to context. In credit (section 5.3.1), surrounded by certificates and contracts, and in insurance (section 5.3.3), among policies and risk reports, the framework and the *LLM Call* with top models consistently delivered near-zero errors across all models. The contrast appeared in *LLM Call* was employed with the smaller model obtaining six errors in credit subset and twice as many (12) in insurance subset. In essences, while the framework remained stable, *LLM Call* showed greater vulnerability when the document was placed in a denser semantic context.

By distinction, the *Risk Report*, illustrates the opposite case: it is the class most sensitive to context. In credit (section 5.3.1), performance was acceptable, though *GPT-3.5-turbo* still pro-

duced up to nine errors. In insurance, however, its proximity to policies and valuation reports caused clear degradation: up to 13 errors in the framework (*GPT-4o-mini*) and 19 in *LLM Call* (*GPT-3.5-turbo*). This confirms that certain analytical documents become significantly harder to isolate when their vocabulary overlaps with that of several neighboring classes.

In sum, this analysis shows that performance depends not only on the model's competence or the retrieval strategy, but also on the documentary context in which a class is embedded. The framework demonstrates greater stability, particularly with fourth-generation models, yet classes such as the Risk Report illustrate how context can amplify challenges, making the classification problem less a matter of isolated difficulty and more one of contextual generalization.

5.4.3 Out-of-Scope Detection

The detection of out-of-scope (OOS) documents is a critical aspect in real-world scenarios, since many incoming files may not belong to the target classes. The introduction of a residual class (as shown in section 5.1) opens the possibility to the system handle these cases differently (for example, by adding a human in the loop), also ensuring, that the system doesn't force unsuitable assignments. This rejection mechanism not only strengthens operational robustness but also lowers the likelihood of critical mistakes, especially in high-stakes areas like credit, legal, and insurance.

In the proposed framework, the rejection factor doesn't result from a probabilistic threshold applied to internal scores, but rather from explicit instructions incorporated into the system prompt (available in Listing A.1). The protocol establishes that whenever the LLM doesn't find an unambiguous match between the content of a document and the formal descriptions of the in-scope classes, the output should be the *Other* class. In addition, three specific conditions for rejection are also imposed:

- ambiguous texts or those with insufficient information should be rejected;
- the mere occurrence of keywords or the title of a class is not sufficient criteria for acceptance;
- only when all elements of a class description are satisfied can the document be assigned to that class.

Thus, the rejection factor is governed by a prompting rule-based protocol, which aims to prioritize security (minimizing false positives) over total coverage.

Although the framework doesn't rely on classical probabilistic calibration methods (e.g., Platt scaling, isotonic regression), consistency is achieved through the standardization and refinement of the prompt. The same system prompt (Listing A.1) is used and applied acrross all evaluated combinations and domains, ensuring a homogeneous rejection criterion. This system prompt was developed iteratively within a validation subset to mitigate potential biases, such as the model's initial tendency to accept documents based solely on the superficial repetition of terms related to the class. Functionally, this process mirrors the role of statistical calibration: rather than adjusting

numeric scores, it adjusts the instruction's wording to align rejection outcomes with the desired behavior. The outcome is a rejection mechanism that is robust, predictable, and comparable across different classification strategies and language models.

With the rejection protocol formally defined, the subsequent analysis addresses the behavior of the *Other* class, with particular emphasis on the distribution of true and false positives.

Since both the proposed framework and the *LLM Call* rely on the same system prompt, the rejection protocol is identical across the two approaches. As a result, the number of true positives remains closely aligned, particularly in larger models. In the legal domain, both reach 29–30 correct rejections, while in insurance the values remain within 28–30. In the credit domain, differences were also marginal (22–26 in the framework versus 24 in the *LLM Call*). For the smaller model (*GPT-3.5-turbo*), the framework did not substantially improved in comparison to the *LLM Call*.

The decisive difference arises in false positives. Here, the structural support of the framework together with the qualitative calibration of the instructions, imposes greater rigor on the rejection process, reducing the model's propensity to misattribute the *Other* class.

In fourth-generation models, errors were residual (0–5 in the framework, compared to up to 7 in the *LLM Call*), concentrated in only a few borderline classes. However, with *GPT-3.5-turbo*, the difference becomes substantially more pronounced. In the credit subset, the framework limited false positives to 8–30, against 46 in the *LLM Call*, corresponding to a reduction between 35% and 83%. A similar pattern emerged in the legal subset, with the framework yielding 24–37 errors versus 51 in the *LLM Call*, amounting to a reduction of 27% to 53%. Finally, in the insurance subset, the framework produced only 11–16 errors compared to 58 in the *LLM Call*, representing the largest gap, with reductions ranging from 72% to 81%.

Overall, the findings confirms that the baseline doesn't even match the performance of the weakest framework strategy, TF.IDF. Moreover, the results indicate that, despite sharing the same rejection protocol, the *LLM Call* tends to use the *Other* class less judiciously, especially in smaller models. This limitation is exacerbated by its strategy of filling the entire context window, which the model often fails to handle effectively. As a consequence, its predictions become less certain and the rejection protocol is triggered unnecessarily more often. In contrast, the structure provides a smaller and more representative context, which better guides the model's decision and avoids arbitrary rejections.

In conclusion, the comparison reveals that both approaches achieve nearly identical results in terms of true positives, but diverge sharply in the handling of false positives. The *LLM Call* frequently misuses the rejection protocol, leading to inflated error rates, particularly when operating with smaller models. By contrast, the framework employs structured retrieval strategies that constrain this behavior, limiting false positives and establishing performance across domains. The evidence therefore indicates that the real advantage of the framework lies not in raising the number of correct rejections, but in avoiding excessive and unwarranted ones.

Chapter 6

Conclusion

6.1 Conclusions

This work presents a comprehensive review of the current state-of-the-art in document classification, examining both traditional and modern approaches based on LLMs, as well as, highlighting and discussing persistent limitations that hinder their real-world applicability. Although traditional methods are historically relevant, they continue to rely heavily on technical knowledge for configuration and implementation. These methods make simplified assumptions (e.g., Naïve Bayes) and are computationally expensive. They also exhibit poor generalization, especially in the presence of unbalanced or small datasets. Meanwhile, modern LLM-based systems, despite improvements in performance and explainability, often remain tied to specific domains, require annotated data, and struggle to adapt to the structural and semantic diversity found in real-world documents. The lack of frameworks that combine zero-shot capabilities, domain independence, and configurability by non-technical users remained a critical gap in the field. In order to address these challenges, it was proposed a framework that integrates zero-shot capabilities with LLMs, enabling accurate classifications without the need for extensive, pre-curated corpora, while ensuring flexibility for multiple domains, ease of use by non-experts, and computational efficiency.

In terms of model comparison, *GPT-4o* proved to be the most robust model, consistently achieving *precision* and *f1-score* metrics above 98%, albeit at a high cost. *GPT-4o-mini* was a optimum balance between performance and efficiency, consistently outperforming all combinations that employ the smaller model *GPT-3.5-turbo*, whose lower performance and higher costs make it unattractive to use.

Among the retrieval strategies evaluated, *RAG* stood out for its consistency and balance between quality and cost, justifying the additional investment in scenarios that require high precision, for example in distinguishing documents with a high degree of similarity (Land Registry Certificates - Land and Urban Registry for the credit subset) and in classifying easily confused documents within the subsets (Deeds and Legal Opinions in the legal subset; Risk Report in the insurance subset). The *Firsts* strategy and *Firsts and Lasts* approach presented intermediate results with quite acceptable performance and were the most economical and, therefore, suitable for large-scale scenarios. The *TF.IDF*-based strategy proved to be the least effective, associated with

higher token consumption and consequently unnecessary cost increases.

In comparisons with the baselines, both *NLI* and *Embedding Similarity* revealed severe limitations, proving to be unable to compete with the strategies integrated into the framework. On the other hand, *LLM Call* demonstrated some competitiveness when paired with fourth-generation models, but at the cost of highly unpredictable and substantially higher expenses than those observed with the framework. With the smaller model (GPT-3.5-turbo) employed, its performance has declined significantly, continuing to consume more tokens and consequently higher costs. Overall, there is no compelling reason to adopt any of these baselines against the proposed framework, since it has consistently higher success rates at a lower cost.

This assessment confirmed the robustness of the framework, which stood out for its superiority over baselines, cost predictability, and scalability in real-world scenarios. The *RAG + GPT-4o-mini* combination proved to be the most balanced setup, achieving results nearly on par with the best-performing model (*GPT-4o*) while keeping costs well below.

Additionally, the case studies across credit validation, legal compliance, and insurance classification demonstrated the framework's generalization ability. In each domain, it delivered superior performance, ensured cost-efficiency, and proved scalable in production environments where thousands of documents are processed daily. These results underline the framework's practical value. Together, these findings position the framework not only as a technical contribution but also as a practical tool ready for real-world adoption in environments where accuracy, efficiency, compliance and sustainability cannot be compromised.

6.2 Future Work

The present study opens multiple avenues for future research. A particularly promising direction involves the evaluation of the framework with open-source language models. This approach was not included in the current research due to the complexity involved in allocating and configuring computational resources, as the primary goal was to deliver a solution that is easily configurable and deployable. In this same thread, it would also be valuable to examine how well the framework generalizes to models beyond OpenAI, particularly those from Google's Gemini and Anthropic's Claude models.

Another promising direction for future work lies in studying the impact of varying key parameters of the framework. This includes testing different embedding models (like *text-embedding-3-small* or *text-embedding-ada-002*), experimenting with alternative values of K (for the number of pages retrieved), exploring distinct chunking strategies (like by paragraph, sliding windows, etc), or even adopting different formats for context organization such as JSON instead of XML. Such an investigation could provide valuable insights into how these design choices influence both performance and cost-efficiency. However, this analysis was left outside the present scope, as the focus of this work was to demonstrate the overall feasibility and practical benefits of the framework, rather than to optimize each individual parameter.

Given that the framework was developed within the banking sector, a natural extension of

this work involves testing its applicability in other document-intensive domains. Such evaluations would allow for a more comprehensive assessment of its robustness and contextual adaptability.

One constraint of this study was the limited dataset size (30 examples per class), which made it unfeasible to apply supervised learning models such as Support Vector Machine or Random Forests. In this way, future research should prioritize the collection and annotation of a larger dataset, thereby enabling direct and fair comparisons between LLMs and traditional ML approaches.

Finally, an important aspect that deserves further attention is the absence of an explicit confidence measure to go along with the explanation provided by the framework. Exploring strategies to estimate or infer such information (for example from *logprobs*) could significantly enhance the system's reliability in sensitive application domains.

Glossary

AI Artificial Intelligence. ix, 1, 3, 5–7, 9, 10, 15, 16

BPNNs Backpropagation Neural Networks. 8, 16

CNNs Convolutional Neural Networks. 8, 16

IA Inteligência Artificial. iv

K-NN K-Nearest Neighbors. 7

LLM Large Language Model. vi, vii, 3, 5, 11–13, 17, 22, 24, 28, 35, 44, 54, 57

LLMs Large Language Models. v, vi, viii, 2–6, 10–13, 15–18, 20, 22, 24, 25, 28, 46, 57, 59

LoRA Low-Rank Adaptation. 11

MESC Multi-stage Encoder-based Supervised with Clustering. 12

ML Machine Learning. 7, 9, 15, 59

NLI Natural Language Inference. 25

NLP Natural Language Processing. 2, 6, 9

NNs Neural Networks. 7–10

NT Naive Tree. 8

OCR Optical Character Recognition. 8, 9, 13, 16, 20

PLN Processamento de Linguagem Natural. v

RAG Retrieval-Augmented Generation. vi, 13, 16, 22

RLHF Reinforcement Learning from Human Feedback. 11

RNNs Recurrent Neural Networks. 8, 16

ROPE Requirement-Oriented Prompt Engineering. 10

TF.IDF Term Frequency-Inverse Document Frequency. iv, 6, 8, 21

VLMs Visual Language Models. 13

VSM Vector Space Model. 7

Bibliography

[1] Alan M. Turing. "Computing machinery and intelligence". In: *Mind* 59.236 (1950), pp. 433–460. DOI: 10.1093/mind/LIX.236.433.

[2] John McCarthy, Marvin L. Minsky, Nathan Rochester, and Claude E. Shannon. "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955". In: *AI Magazine* 27.4 (2006), p. 12. DOI: 10.1609/aimag.v27i4.1904.

[3] H. Borko and M. Bernick. "Automatic Document Classification". In: *Journal of the ACM* 10.2 (1963), pp. 151–162. ISSN: 0004-5411. DOI: 10.1145/321160.321165.

[4] Karen Sparck Jones. "A Statistical Interpretation of Term Specificity and Its Application in Retrieval". In: *Journal of Documentation* 28.1 (1972), pp. 11–21. ISSN: 0022-0418. DOI: 10.1108/eb026526.

[5] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of Massive Datasets*. 3rd. USA: Cambridge University Press, 2020. ISBN: 1107077230.

[6] Fabrizio Sebastiani. "Machine learning in automated text categorization". In: *ACM Computing Surveys (CSUR)* 34.1 (2002), pp. 1–47. ISSN: 1557-7341. DOI: 10.1145/505282.505283.

[7] European Commission, Joint Research Centre, B Delipetrev, C Tsinaraki, and U Kostić. *AI watch, historical evolution of artificial intelligence – Analysis of the three main paradigm shifts in AI*. Publications Office, 2020. DOI: 10.2760/801580.

[8] Chidanand Apté, Fred Damerau, and Sholom Weiss. "Knowledge Discovery for Document Classification". In: *Proceedings of the 2nd International Conference on Knowledge Discovery in Databases (AAAIWS'93)*. AAAI Press, 1993, pp. 326–336. URL: https://dl.acm.org/doi/10.5555/3000767.3000799.

[9] Zhou Yong, Li Youwen, and Xia Shixiong. "An Improved KNN Text Classification Algorithm Based on Clustering". In: *Journal of Computers* 4 (2009). ISSN: 1796203X. DOI: 10.4304/jcp.4.3.230-237.

[10] Kansheng Shi, Lemin Li, Haitao Liu, Jie He, Naitong Zhang, and Wentao Song. "An improved KNN text classification algorithm based on density". In: *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*. 2011, pp. 113–117. DOI: 10.1109/CCIS.2011.6045043.

[11] Yong Wang, J. Hodges, and Bo Tang. "Classification of Web documents using a naive Bayes method". In: *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*. 2003, pp. 560–564. DOI: 10.1109/TAI.2003.1250241.

[12] Yanfang Wang, James Hodges, and Binqiang Tang. "Classification of Web Documents Using a Naive Bayes Method". In: *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2003, pp. 560–564. DOI: 10.1109/TAI.2003.1250241.

[13] Loc Nguyen. "A Proposal of Discovering User Interest by Support Vector Machine and Decision Tree on Document Classification". In: *2009 International Conference on Computational Science and Engineering*. Vol. 4. 2009, pp. 809–814. DOI: 10.1109/CSE.2009.112.

[14] Jiang Su and Harry Zhang. "A fast decision tree learning algorithm". In: *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*. AAAI'06. Boston, Massachusetts: AAAI Press, 2006, pp. 500–505. ISBN: 9781577352815. URL: https://dl.acm.org/doi/10.5555/1597538.1597619.

[15] Cheng Hua Li and Soon Choel Park. "An efficient document classification model using an improved back propagation neural network and singular value decomposition". In: *Expert Systems with Applications* 36.2, Part 2 (2009), pp. 3208–3215. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2008.01.014.

[16] Larry Manevitz and Malik Yousef. "One-class document classification via Neural Networks". In: *Neurocomputing* 70.7 (2007). Advances in Computational Intelligence and Learning, pp. 1466–1481. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2006.05.013.

[17] Amy J.C. Trappey, Fu-Chiang Hsu, Charles V. Trappey, and Chia-I Lin. "Development of a patent document classification and search platform using a back-propagation network". In: *Expert Systems with Applications* 31.4 (2006). Computer Supported Cooperative Work in Design and Manufacturing, pp. 755–765. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2006.01.013.

[18] Simon J. D. Prince. "Chapters 1, 3, 4, 12". In: *Understanding Deep Learning*. The MIT Press, 2024, pp. 207–239. ISBN: 9780262048644. URL: http://udlbook.com.

[19] Hubert L. Dreyfus. *Alchemy and Artificial Intelligence*. Santa Monica, CA: RAND Corporation, 1965. URL: https://www.rand.org/content/dam/rand/pubs/papers/2006/P3244.pdf.

[20] Michael Wooldridge. *A Brief History of Artificial Intelligence: What It Is, Where We Are, and Where We Are Going*. 1st. Flatiron Books, 2021. ISBN: 1250770742.

[21] A. Toosi, A. G. Bottino, B. Saboury, E. Siegel, and A. Rahmim. "A brief history of AI: How to prevent another winter (A critical review)". In: *PET Clinics* 16.4 (2021), pp. 449–469. ISSN: 1556-8598. DOI: 10.1016/j.cpet.2021.07.001.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://dl.acm.org/doi/10.5555/3295222.3295349.

[23] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. *A Survey of Large Language Models*. 2025. DOI: 10.48550/arXiv.2303.18223.

[24] Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. "A Survey on Text Classification: From Traditional to Deep Learning". In: 13.2 (Apr. 2022). ISSN: 2157-6904. DOI: 10.1145/3495162.

[25] Lingfei Fan, Linyi Li, Zhiyuan Ma, Sukyoung Lee, Hongming Yu, and Libby Hemphill. "A bibliometric review of large language models research from 2017 to 2023". In: *arXiv* (2023). DOI: 10.48550/arXiv.2304.02020.

[26] D. Ver Meer. *Number of ChatGPT users and key stats (December 2024)*. Accessed: January 2025. Dec. 2024. URL: https://www.namepepper.com/chatgpt-users.

[27] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era". In: *CoRR* (2017). DOI: 10.48550/arXiv.1707.02968.

[28] R. T. Dattola. "A Fast Algorithm for Automatic Classification". In: *Information Technology and Libraries* 2.1 (1969), pp. 31–48. DOI: 10.6017/ital.v2i1.3797.

[29] John V. Guttag. *Introduction to Computation and Programming Using Python*. 2nd. The MIT Press, 2016. ISBN: 9780262542364. URL: https://mitpress.mit.edu/9780262542364/introduction-to-computation-and-programming-using-python/.

[30] Hui Gao, Jun Jiang, Li She, and Yan Fu. "A New Agglomerative Hierarchical Clustering Algorithm Implementation based on the Map Reduce Framework." In: *JDCTA* 4 (June 2010), pp. 95–100. URL: https://www.researchgate.net/publication/220670275_A_New_Agglomerative_Hierarchical_Clustering_Algorithm_Implementation_based_on_the_Map_Reduce_Framework.

[31] Peter Flach. *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*. Cambridge, UK: Cambridge University Press, 2012. ISBN: 9781107422223.

[32] Guoqiang Zhong, Hui Yao, Yutong Liu, Chen Hong, and Tuan Pham. "Classification of photographed document images based on deep-learning features". In: *Eighth International Conference on Graphic and Image Processing (ICGIP 2016)*. Ed. by Yulin Wang, Tuan D. Pham, Vit Vozenilek, David Zhang, and Yi Xie. Vol. 10225. International Society for Optics and Photonics. SPIE, 2017, p. 102250X. DOI: 10.1117/12.2266984.

[33] C.P. Sumathi, T Santhanam, and G Devi. "A Survey On Various Approaches Of Text Extraction In Images". In: *International Journal of Computer Science and Engineering Survey* 3 (Sept. 2012). URL: https://www.researchgate.net/publication/267861120_A_Survey_On_Various_Approaches_Of_Text_Extraction_In_Images.

[34] Rishabh Mittal and Anchal Garg. "Text extraction using OCR: A Systematic Review". In: *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*. 2020, pp. 357–362. DOI: 10.1109/ICIRCA48905.2020.9183326.

[35] Noman Islam, Zeeshan Islam, and Nazia Noor. "A Survey on Optical Character Recognition System". In: *ITB Journal of Information and Communication Technology* (Dec. 2016). DOI: 10.48550/arXiv.1710.05703.

[36] Md. Zahangir Alom, Tarek M. Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C. Van Essen, Abdul A. S. Awwal, and Vijayan K. Asari. "The history began from AlexNet: A comprehensive survey on deep learning approaches". In: *arXiv* (2018). DOI: 10.48550/arXiv.1803.01164.

[37] Michał M. Mirończuk and Jakub Protasiewicz. "A recent overview of the state-of-the-art elements of text classification". In: *Expert Systems with Applications* 106 (2018), pp. 36–54. DOI: 10.1016/j.eswa.2018.03.058.

[38] Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space". In: *Proceedings of Workshop at ICLR* 2013 (Jan. 2013). DOI: 10.48550/arXiv.1301.3781.

[39] Djoerd Hiemstra. "N-Gram Models". In: *Encyclopedia of Database Systems*. Ed. by LING LIU and M. TAMER ÖZSU. Boston, MA: Springer US, 2009, pp. 1910–1910. ISBN: 978-0-387-39940-9. DOI: 10.1007/978-0-387-39940-9_935.

[40] Usman Naseem, Imran Razzak, Shah Khalid Khan, and Mukesh Prasad. "A Comprehensive Survey on Word Representation Models: From Classical to State-Of-The-Art Word Representation Language Models." In: *ACM Symposium on Neural Gaze Detection* (June 2018), p. 46. DOI: 10.1145/1122445.1122456.

[41] Quoc V. Le and Tomás Mikolov. "Distributed Representations of Sentences and Documents". In: *CoRR* abs/1405.4053 (2014). DOI: 10.48550/arXiv.1405.4053.

[42] Hugo Touvron et al. "Llama: Open and Efficient Foundation Language Models". In: (2023). DOI: 10.48550/arXiv.2302.13971.

[43] Anthropic. *Introducing Claude*. https://www.anthropic.com/index/introducing-claude. Accessed: 2025-08-13. 2023.

[44] Alec Radford and Karthik Narasimhan. "Improving Language Understanding by Generative Pre-Training". In: 2018. URL: https://api.semanticscholar.org/CorpusID:49313245.

[45] Mohaimenul Azam Khan Raiaan, Md. Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. "A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges". In: *IEEE Access* 12 (2024), pp. 26839–26874. DOI: 10.1109/ACCESS.2024.3365742.

[46] Albert Badalyan. *Number Of ChatGPT Users In 2025: Stats, Usage & Impact*. Digital Trends (Digital Silk). Accessed: 16 August 2025. 2025. URL: https://www.digitalsilk.com/digital-trends/number-of-chatgpt-users/.

[47] AllAboutAI Research Team. *How Many People Use ChatGPT Daily? (2025 Statistics)*. AllAboutAI. https://www.allaboutai.com/resources/how-many-people-use-chatgpt-daily/, accessed 16 August 2025. 2025.

[48] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. *A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications*. 2025. DOI: 10.48550/arXiv.2402.07927.

[49] Qianou Ma, Weirui Peng, Chenyang Yang, Hua Shen, Kenneth Koedinger, and Tongshuang Wu. "What Should We Engineer in Prompts? Training Humans in Requirement-Driven LLM Use". In: *ACM Transactions on Computer-Human Interaction* (Apr. 2025). ISSN: 1557-7325. DOI: 10.1145/3731756.

[50] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. *Large Language Models are Zero-Shot Reasoners*. 2023. DOI: 10.48550/arXiv.2205.11916.

[51] Murray Shanahan. "Talking about Large Language Models". In: *Commun. ACM* 67.2 (Jan. 2024), pp. 68–79. ISSN: 0001-0782. DOI: 10.1145/3624724.

[52] Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. *The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities*. 2024. DOI: 10.48550/arXiv.2408.13296.

[53] Mubashar Raza, Zarmina Jahangir, Muhammad Bilal Riaz, Muhammad Jasim Saeed, and Muhammad Awais Sattar. "Industrial applications of large language models". In: *Scientific Reports* 15.1 (2025), p. 13755. ISSN: 2045-2322. DOI: 10.1038/s41598-025-98483-1.

[54] Dominik Trautmann. "Large language model prompt chaining for long legal document classification". In: *SwissText 2023 Late Breaking Work (Generative AI & LLM)*. 2023. DOI: 10.48550/arXiv.2308.04138.

[55] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019. DOI: 10.48550/arXiv.1908.10084.

[56] Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. "When Does Pretraining Help? Assessing Self-Supervised Learning for Law and the CaseHOLD Dataset". In: *Proceedings of the 18th International Conference on Artificial Intelligence and Law*. Association for Computing Machinery, 2021. DOI: 10.48550/arXiv.2104.08671.

[57] N. Prasad, M. Boughanem, and T. Dkaki. "Exploring Large Language Models and Hierarchical Frameworks for Classification of Large Unstructured Legal Documents". In: *Advances in Information Retrieval: Proceedings of the 46th European Conference on Information Retrieval (ECIR 2024), Part II*. Springer, 2024, pp. 221–237. DOI: 10.1007/978-3-031-56060-6_15.

[58] Shashank Menon and Carl Vondrick. "Visual classification via description from large language models". In: *arXiv* (2022). DOI: 10.48550/arXiv.2210.07183.

[59] Lefteris Loukas, Ilias Stogiannidis, Odysseas Diamantopoulos, Prodromos Malakasiotis, and Stavros Vassos. "Making LLMs Worth Every Penny: Resource-Limited Text Classification in Banking". In: *Proceedings of the Fourth ACM International Conference on AI in Finance*. ICAIF '23. Brooklyn, NY, USA: Association for Computing Machinery, 2023, pp. 392–400. ISBN: 9798400702402. DOI: 10.1145/3604237.3626891.

[60] Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. *Text Classification via Large Language Models*. 2023. DOI: 10.48550/arXiv.2305.08377.

[61] Konstantinos I. Roumeliotis, Nikolaos D. Tselikas, and Dimitrios K. Nasiopoulos. "Leveraging Large Language Models in Tourism: A Comparative Study of the Latest GPT Omni Models and BERT NLP for Customer Review Classification and Sentiment Analysis". In: *Information* 15.12 (2024). ISSN: 2078-2489. DOI: 10.3390/info15120792.

[62] Fusheng Wei, Robert Keeling, Nathaniel Huber-Fliflet, Jianping Zhang, Adam Dabrowski, Jingchao Yang, Qiang Mao, and Han Qin. "Empirical Study of LLM Fine-Tuning for Text Classification in Legal Document Review". In: *2023 IEEE International Conference on Big Data (BigData)*. 2023, pp. 2786–2792. DOI: 10.1109/BigData59044.2023.10386911.

[63] Artifex Software Inc. and contributors. *PyMuPDF: Python bindings for MuPDF*. Version 1.24.14. 2024. URL: https://pymupdf.readthedocs.io/.

[64] D. O'Riordan Matthew. *Python-tesseract is an optical character recognition (OCR) tool for python*. Accessed: November 2024. 2024. URL: https://github.com/madmaze/pytesseract.

[65] Ray Smith. *Tesseract: An Open-Source OCR Engine*. Accessed: November 2024. 2007. URL: https://github.com/tesseract-ocr/tesseract.

[66] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. "Retrieval-Augmented Generation for Large Language Models: A Survey". In: *arXiv preprint arXiv:2312.10997* (2023). Ongoing Work. DOI: 10.48550/arXiv.2312.10997.

[67] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Accepted at NeurIPS 2020. 2020. DOI: 10.48550/arXiv.2005.11401.

[68] Kim Martineau. *What is retrieval-augmented generation?* https://research.ibm.com/blog/retrieval-augmented-generation-RAG. Accessed: 2024-12-26. Aug. 2023.

[69] OpenAI. *Embeddings - OpenAI Documentation*. Accessed: 2024-12-23. 2024. URL: https://platform.openai.com/docs/guides/embeddings/.

[70] LangChain. *InMemoryVectorStore - LangChain Documentation*. 2024. URL: https://python.langchain.com/api_reference/core/vectorstores/langchain_core.vectorstores.in_memory.InMemoryVectorStore.html.

[71] Sanjay Kukreja, Tarun Kumar, Vishal Bharate, Amit Purohit, Abhijit Dasgupta, and Debashis Guha. "Vector Databases and Vector Embeddings-Review". In: *2023 International Workshop on Artificial Intelligence and Image Processing (IWAIIP)*. 2023, pp. 231–236. DOI: 10.1109/IWAIIP58158.2023.10462847.

[72] Yikun Han, Chunjiang Liu, and Pengfei Wang. "A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge". In: *arXiv preprint arXiv:2310.11703* (2023). DOI: 10.48550/arXiv.2310.11703.

[73] Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. "Does Prompt Formatting Have Any Impact on LLM Performance?" In: *arXiv preprint arXiv:2411.10541* (2024). Submitted to NAACL 2025. DOI: 10.48550/arXiv.2411.10541.

[74] OpenAI. *OpenAI Guide to Prompt Engineering*. 2024. URL: https://platform.openai.com/docs/guides/prompt-engineering.

[75] Jiyeon Park and Sam Choo. "Generative AI Prompt Engineering for Educators: Practical Strategies". In: *Journal of Special Education Technology* 0.0 (0), p. 01626434241298954. DOI: 10.1177/01626434241298954.

[76] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. "Chain of Thought Prompting Elicits Reasoning in Large Language Models". In: *CoRR* (2022). DOI: 10.48550/arXiv.2201.11903.

[77] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *CoRR* (2019). DOI: 10.48550/arXiv.1910.13461.

[78] Wenpeng Yin, Jamaal Hay, and Dan Roth. "Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP) and IJCNLP*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 3914–3923. DOI: 10.18653/v1/D19-1404.

[79] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. "MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020. URL: https://www.microsoft.com/en-us/research/publication/minilm-deep-self-attention-distillation-for-task-agnostic-compression-of-pre-trained-transformers/.

[80] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. "MiniLMv2: Multi-Head Self-Attention Relation Distillation for Compressing Pretrained Transformers". In: *arXiv preprint arXiv:2012.15828* (2021). DOI: 10.48550/arXiv.2012.15828.

[81] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. "LEGAL-BERT: The muppets straight out of law school". In: (2020). DOI: 10.48550/arXiv.2010.02559.

[82] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[83] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. "Domain-specific language model pretraining for biomedical natural language processing". In: *ACM Transactions on Computing for Healthcare* 3.1 (2022), pp. 1–23. DOI: 10.1145/3458754.

[84] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bryan Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. "Parameter-efficient transfer learning for NLP". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2790–2799. DOI: 10.48550/arXiv.1902.00751.

[85] Edward J Hu, Yelong Shen, Phil Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. "LoRA: Low-rank adaptation of large language models". In: *International Conference on Learning Representations (ICLR)*. 2022. DOI: 10.48550/arXiv.2106.09685.

[86] Xiang Lisa Li and Percy Liang. "Prefix-tuning: Optimizing continuous prompts for generation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2021. DOI: 10.48550/arXiv.2101.00190.

[87] OpenAI. *GPT-4 Technical Report*. Tech. rep. Retrieved from https://openai.com/research/gpt-4. OpenAI, 2023. URL: https://openai.com/research/gpt-4.

[88] Md. Abul Kalam Raiaan et al. "A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges". In: *IEEE Access* 12 (2024), pp. 26839–26874. DOI: 10.1109/ACCESS.2024.3365742.

Appendix A

Framework informations

A.1 Label - Descriptions used

Table A.1: All pairs Label - Description used. All written in Portuguese because the framework was analyzing strictly Portuguese documents.

Label	Description
Balancete	Um balancete é um relatório contábil de estrutura tabular que mostra os saldos das contas de uma empresa em um período específico, geralmente com dados detalhados desse período ou valores acumulados. Normalmente é composto por débitos, créditos e saldos ajudando a visualizar de forma clara o total de débitos e créditos das várias contas.
RelatorioContas	O relatório e contas de uma empresa é um documento que analisa sua performance e situação financeira em determinado período, com base na contabilidade. O relatório inclui dados sobre a atividade da empresa e suas demonstrações financeiras tal como Balanço, Demonstrações de resultados, Demonstração das alterações no capital próprio e Fluxos de caixa. Este relatório pode também incluir informações, certificações ou avaliações detalhadas sobre os seus ativos incluindo imóveis, frotas automóveis entre outros.
CadernetaPredialUrbana	A caderneta predial urbana serve para consultar informações fiscais sobre um imóvel, sendo este um documento com valor legal emitido pelas Finanças. A informação contida no documento inclui a identificação do prédio, a localização, e características físicas.
CertidaoRegistoPredial	A certidão de Registo Predial descreve de uma maneira muito detalhada todo o histórico da habitação no que se refere à sua constituição e localização. Além disso, também serve para confirmar a titularidade dos seus proprietários.
CertificadoEnergetico	Um certificado energético é um documento que avalia um bem imóvel numa escala de performance energética de A+ a F (classe energética) e ainda constitui indicadores de desempenho que determinam a classe energética do edíficio e a eficiência na utilização de energia.

Label	Description
CertidaoRegistoComercial	A certidão de Registo Comercial destina-se a comprovar a existência do registo comercial de uma entidade, contendo, não só informações sobre a sua constituição, como também todas as atualizações de registos que forem ocorrendo. É um documento é atribuído a partir do momento da sua constituição de uma empresa, composto por informações que podem ser consultadas publicamente sobre ela. Contém informações como a natureza jurídica, sede, capital, cae, sócios ou gestores, entre outros.
Escritura	Uma escritura é um documento público, normalmente emitido por um cartório notarial, que formaliza um ato jurídico, como a compra e venda de um imóvel, doações, contratos ou testamentos. Geralmente costuma conter um título, a identificação dos intervenientes (exemplo: primeiros, segundos) a qualificação das partes, a descrição do bem, declarações e condições, a base legal e as assinaturas das partes e do tabelião, garantindo sua validade.
RelatorioAvaliacaoImovel	O Relatório de Avaliação Imobiliária é totalmente relacionado à análise de um imóvel em específico, normalmente estimando o seu valor (resídual, de mercado, entre outros), com base nas suas características (antiguidade, localização, entre outros). Não prevê nada mais que avaliação de um imóvel.
ApoliceSeguroAutomovel	A apólice de seguro automóvel é o documento que formaliza o contrato entre o segurado e a seguradora relativamente a um veículo automóvel. Pode conter informações como coberturas contratadas (responsabilidade civil obrigatória, danos próprios, assistência em viagem, entre outras), limites de capital, exclusões, franquias e o período de validade. Este documento identifica o veículo, o tomador do seguro e as condições em que a seguradora garante a reparação de danos ou indemnização.
ApoliceSeguroSaude	A apólice de seguro de saúde é um documento contratual que estabelece as condições de cobertura de despesas médicas e hospitalares do segurado. Inclui informação sobre os serviços abrangidos, tais como consultas, exames, internamentos, cirurgias, medicamentos, rede de prestadores, limites de capital e períodos de carência. Define ainda exclusões e copagamentos quando aplicável.
ApoliceSeguroTrabalho	A apólice de seguro de acidentes de trabalho é o contrato que garante a proteção dos trabalhadores contra riscos de acidentes ou doenças profissionais decorrentes da sua atividade laboral. O documento especifica as coberturas, capitais seguros, exclusões e direitos do trabalhador em caso de sinistro, incluindo prestações de natureza médica, indemnizações e pensões.
ApoliceSeguroVida	A apólice de seguro de vida é o documento que formaliza a cobertura financeira em caso de morte ou invalidez do segurado. Contém a identificação do tomador e beneficiários, as coberturas contratadas, capitais seguros, exclusões e prazo de validade. O objetivo é assegurar apoio económico aos beneficiários designados, garantindo proteção financeira em situações imprevistas.

Label	Description
ApoliceSeguroMultirriscos	A apólice de seguro multirriscos é o contrato que protege bens imóveis (como habitações, escritórios ou estabelecimentos comerciais) contra diversos riscos, tais como incêndio, inundações, fenómenos naturais, roubo e responsabilidade civil. O documento define as coberturas incluídas, capitais, franquias e exclusões, assegurando uma proteção abrangente do património segurado.
RelatorioRisco	O Relatório de risco ou avaliação de risco é um documento técnico elaborado por uma entidade ou especialista com o objetivo de avaliar o nível de risco associado a uma operação, projeto, investimento, financiamento ou atividade específica. Contém a análise detalhada dos fatores de risco, metodologias de avaliação utilizadas, critérios de classificação, conclusões e recomendações. Este documento é frequentemente utilizado por instituições financeiras, seguradoras ou órgãos reguladores como suporte à tomada de decisão.
ParecerJuridico	O parecer jurídico é um documento elaborado por um advogado ou especialista em direito que tem como objetivo analisar uma situação, contrato, decisão ou problema sob a perspetiva legal. Apresenta a interpretação das normas aplicáveis, identifica riscos e implicações jurídicas, e fornece recomendações fundamentadas para apoiar a tomada de decisão. Este documento pode ser solicitado por empresas, instituições ou particulares e serve como orientação técnica e suporte em processos administrativos, negociais ou judiciais.
ContratoFinanciamento	O contrato de financiamento é um documento jurídico que formaliza o acordo entre uma instituição financeira e um cliente (particular ou empresa) para a disponibilização de crédito em determinadas condições. Estabelece os montantes financiados, prazos de reembolso, taxas de juro, garantias associadas, obrigações das partes, penalizações por incumprimento e demais cláusulas contratuais. Este documento serve como base legal e regulatória da relação entre financiador e financiado, garantindo direitos e deveres de ambas as partes.
ContratoArrendamento	Um contrato de arrendamento é um acordo jurídico pelo qual uma parte, denominada senhorio, concede a outra, denominada arrendatário, o direito de uso e fruição de um imóvel, mediante o pagamento de uma renda e por um prazo determinado, estabelecendo direitos, deveres e garantias para ambas as partes.
AtaSocietaria	Uma ata societária é um documento formal que regista de maneira escrita e fidedigna as deliberações tomadas pelos órgãos de uma sociedade, como assembleia geral ou conselho de administração, assegurando validade legal às decisões e servindo como prova perante sócios, administradores e entidades externas.

A.2 Prompts

Listing A.1: System Prompt

```
You are an expert in classification of documents. It is going to be
   given labels and their descriptions.
It is also going to be provided pieces of text from different chunks.
Your task is to interpret the all the input data and classify into one
   of the given labels.
However, if you don't find any appropriate label, classify as 'Other'.

What do you need to pay attention:
    - The user may provide some text that may not be clear to classify.
    - Simply finding the title or repetition of the label is not a
       sufficient condition to return a label other than 'Other'.
    - All the information of the label description must match the text
       provided by the user. If not, classify as 'Other'.
```

Listing A.2: Input prompt

```
[
    {
        'role': 'system',
        'content': "You are an expert in classification of documents. It is
             going to be given labels and their descriptions. It is also
            going to be provided pieces of text from different chunks. Your
             task is to interpret the all the input data and classify into
            one of the given labels. However, if you don't find any
            appropriate label, classify as 'Other'. What do you need to pay
             attention: The user may provide some text that may not be
            clear to classify. Simply finding the title or repetition of
            the label is not a sufficient condition to return a label other
             than "Other". All the information of the label description
            must match the text provided by the user. If not, classify as '
            Other'."
    },
    {
        'role': 'system',
        'content': """
<label>
   <name> CertidaoRegistoComercial </name>
   <description> A Certidao de Registo Comercial destina-se a comprovar a
       existencia do registo comercial de uma entidade, contendo, nao so
       informacoes sobre a sua constituicao, como tambem todas as
       atualizacoes de registos que forem ocorrendo. E um documento e
       atribuido a partir do momento da sua constituicao de uma empresa,
       composto por informacoes que podem ser consultadas publicamente
       sobre ela. Contem informacoes como a natureza juridica, sede,
       capital, cae, socios ou gestores, entre outros. </description>
</label>
<label>
   <name> ParecerJuridico </name>
   <description> O parecer juridico e um documento elaborado por um
       advogado ou especialista em direito que tem como objetivo analisar
       uma situacao, contrato, decisao ou problema sob a perspetiva legal.
       Apresenta a interpretacao das normas aplicaveis, identifica riscos e
        implicacoess juridicas, e fornece recomendacoes fundamentadas para
       apoiar a tomada de decisao. Este documento pode ser solicitado por
       empresas, instituicoes ou particulares e serve como orientacao
```

```
       tecnica e suporte em processos administrativos, negociais ou
       judiciais. </description>
</label>
...
"""
    },
    {
        'role': 'user',
        'content': """<chunks>
   <start_chunk>
      <number> 1 </number>
      <text> 2/7 RUA RAINHA D. ESTEFANIA, 251 contrato, pretendendo agora o
           municipio utilizar aquelas verbas em diferentes obras embora
          todas da mesma natureza daquelas que haviam sido previamente
          aprovadas. Cumpre, pois, informar: A apreciacao das questoes
          colocadas pressupoe uma incursao, ainda que necessariamente breve
          , pelo regime de credito e de endividamento municipal -
          genericamente regulado na Lei n. 73/2013, de 03 de setembro2 e na
           Lei n. 75/2013, de 12 ... </text>
   </end_chunk>
   <start_chunk>
      <number> 2 </number>
      <text> E tambem sabido que, em obediencia ao principio da tipicidade
          dos emprestimos das autarquias locais, os mesmos apenas podem ser
           contraidos para a prossecuccao de finalidade especificamente
          prevista na lei.7 Em concreto, no que respeita aos emprestimos de
           medio e longo prazo, ... </text>
   </end_chunk>
    ...
 """
    }
]
```

Listing A.3: Framework Response

```
{
    "File Name": "pj-25",
    "Number of Pages": 7,
    "Classification Model": "GPT_4O",
    "Retrieval Mode": "RAG",
    "Results": {
        "Classification": "ParecerJuridico",
        "Rationale": "O texto fornecido e um parecer juridico que analisa a
             situacao de alteracao dos projetos de investimento
            identificados no contrato de emprestimo, sob a perspectiva
            legal. Apresenta a interpretacao das normas aplicaveis,
            identifica riscos e implicancias juridicas, e fornece
            recomendacoes fundamentadas para apoiar a tomada de decisao.
            Portanto, corresponde a descricaoo da label 'ParecerJuridico'."
            ,
        "Time": "6.7s",
        "Number of Tokens": 3880
    }
}
```

A.3 Pricing informations

Model	Price per 1M tokens (USD)
GPT-4o	2.50
GPT-4o-mini	0.15
GPT-3.5-turbo	0.50

Table A.2: Prices per million tokens for different models