



SCHOOL OF COMPUTATION, INFORMATION
AND TECHNOLOGY - INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics: Data Engineering and Analytics

**Distillation of Semantically Similar Context
Windows into Disjoint and Complete Class
Archetype Sets**

Maria Nakhla



SCHOOL OF COMPUTATION, INFORMATION
AND TECHNOLOGY - INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics: Data Engineering and Analytics

**Distillation of Semantically Similar Context
Windows into Disjoint and Complete Class
Archetype Sets**

**Destillation von semantisch ähnlichen
Kontextfenstern in disjunkte und vollständige
Klassenarchetypenmengen**

| | |
|------------------|-----------------------------|
| Author: | Maria Nakhla |
| Supervisor: | Prof. Dr. Florian Matthes |
| Advisor: | Stephen Meisenbacher, M.Sc. |
| Submission Date: | 27/04/2025 |

I confirm that this master's thesis in informatics: data engineering and analytics is my own work and I have documented all sources and material used.

Munich, 27.04.2025

Location, Submission Date

Maria
Vakhta

Author

AI Assistant Usage Disclosure

Introduction

Performing work or conducting research at the Chair of Software Engineering for Business Information Systems (sebis) at TUM often entails dynamic and multi-faceted tasks. At sebis, we promote the responsible use of *AI Assistants* in the effective and efficient completion of such work. However, in the spirit of ethical and transparent research, we require all student researchers working with sebis to disclose their usage of such assistants.

For examples of correct and incorrect AI Assistant usage, please refer to the original, unabridged version of this form, located at this link.

Use of *AI Assistants* for Research Purposes

I have used AI Assistant(s) for the purposes of my research as part of this thesis.

X Yes No

Explanation:

- **Language Support and Enhancing Writing Assistance:** AI-powered language model was used to enhance clarity, grammar, and paraphrasing in the thesis writing. However, any piece of writing was then reviewed and adapted by the author.

I confirm in signing below, that I have reported all usage of AI Assistants for my research, and that the report is truthful and complete.

Munich, 27.04.2025

Location, Date



Author

Acknowledgments

First and foremost, I thank God for His endless guidance, wisdom, and strength that illuminated my path throughout this journey. In every challenge and moment of uncertainty, His presence gave me clarity, perseverance, and hope.

I would like to express my deepest gratitude to Prof. Dr. Florian Matthes for his insightful mentorship and inspiring leadership throughout my academic development. His guidance shaped not only the direction of this thesis but also the way I approach research and knowledge.

I am especially grateful to my advisor, Stephen Meisenbacher, for his continuous support, valuable feedback, and unwavering encouragement. His patience and dedication helped me refine my ideas and push the boundaries of my understanding.

To my beloved parents and brother, thank you for your unconditional love and support. Your belief in me has been a cornerstone of my motivation, and your sacrifices never go unnoticed. You have been my anchor through all highs and lows.

I am also thankful to my friends, whose companionship, encouragement, and shared moments of joy and stress made this journey more meaningful and memorable.

This thesis stands as a reflection of all the love, guidance, and support I have received. To all who contributed to this milestone, I am deeply thankful.

Abstract

The proliferation of unstructured text data has created an urgent need for efficient and effective methods to transform these data into structured and annotated datasets for AI applications. This general objective involves a data distillation step, which, in this study, focuses on the convergence of semantically relevant context windows into coherent archetypes. The generated archetypes, representing a domain or subdomain, are subsequently used to classify text contexts instead of relying on millions of general domain contexts, which may include irrelevant information or duplications.

The proposed approach involves collecting datasets from crucial domains, including Technology, Business, Sports, Politics, and Entertainment. First, context windows are extracted for each domain using a previously implemented pipeline. Then, a recursive hierarchical clustering approach is employed to group relevant subcontexts within each domain. This clustering step paves the way for experimenting with various large language models (LLMs) to generate an archetype per cluster. Prompt engineering techniques are explored to refine the retrieval of high-quality archetypal LLM outputs, with numerous iterations ensuring that the LLM-generated outputs adhere to the desired format. Edge cases are handled carefully to ensure proper parsing. By the end of this process, each cluster maps to a single archetypal rule, and the combination of all clusters' archetypes results in a comprehensive archetype set for a specific domain.

Various evaluation techniques were utilized, including semantic search, fine-tuning text classifiers, conducting surveys, and re-clustering generated domain archetypes. These methods provide a consistent evaluation framework to assess archetype completeness, disjointness, and insights into training classifiers on full-domain text versus domain archetypes.

The results of this study demonstrate that training classifiers on domain archetypes outperforms training on full text. Furthermore, although Meta-Llama-3-8B-Instruct model is considered to be a light-weight LLM, it surpasses other bigger models in contextual knowledge distillation and archetype generation. There are a few limitations such as length bias in human evaluations, inconsistencies between subjective human ratings and classification metrics, and dataset scope. Longer generated archetypes were often favored by the survey participants, possibly conflating verbosity with quality. Results may not generalize beyond coarse-grained news domains. Future work should target nuanced, multilingual corpora.

Overall, this research contributes to the development and evaluation of an automated, domain-expert-driven approach to archetype creation. This method accurately captures domain-specific knowledge and enhances the quality of subsequent text classifications. The research successfully fulfills its goal of transforming unstructured text data into structured and annotated datasets, ultimately supporting the creation of more effective AI applications.

Kurzfassung

Die Verbreitung unstrukturierter Textdaten hat einen dringenden Bedarf an effizienten und effektiven Methoden zur Umwandlung dieser Daten in strukturierte und annotierte Datensätze für KI-Anwendungen geschaffen. Dieses allgemeine Ziel umfasst einen Schritt der Daten-Destillation, der sich in dieser Studie auf die Konvergenz semantisch relevanter Kontextfenster zu kohärenten Archetypen konzentriert. Die generierten Archetypen, die ein bestimmtes Fachgebiet oder Untergebiet repräsentieren, werden anschließend verwendet, um Textkontexte zu klassifizieren anstelle der Verwendung von Millionen allgemeiner Kontexte, die irrelevante Informationen oder Duplikate enthalten können.

Der vorgeschlagene Ansatz umfasst die Sammlung von Datensätzen aus wichtigen Bereichen wie Technologie, Wirtschaft, Sport, Politik und Unterhaltung. Zunächst werden Kontextfenster für jede Domäne mithilfe einer zuvor implementierten Pipeline extrahiert. Anschließend wird ein rekursiver hierarchischer Clustering-Ansatz verwendet, um relevante Subkontexte innerhalb jeder Domäne zu gruppieren. Dieser Clustering-Schritt ebnet den Weg für Experimente mit verschiedenen großen Sprachmodellen (LLMs), um pro Cluster einen Archetypen zu generieren. Es werden Techniken des Prompt Engineerings erforscht, um die Gewinnung qualitativ hochwertiger Archetypen durch LLMs zu verfeinern. Zahlreiche Iterationen stellen sicher, dass die LLM-Ausgaben dem gewünschten Format entsprechen. Sonderfälle werden sorgfältig behandelt, um eine korrekte Analyse zu gewährleisten. Am Ende dieses Prozesses wird jeder Cluster einer einzelnen archetypischen Regel zugeordnet, und die Kombination aller Archetypen der Cluster ergibt ein umfassendes Archetypen-Set für eine bestimmte Domäne.

Es wurden verschiedene Evaluierungsmethoden eingesetzt, darunter semantische Suche, Feinabstimmung von Textklassifikatoren, Umfragen und erneutes Clustern der generierten Domänen-Archetypen. Diese Methoden bieten einen konsistenten Evaluierungsrahmen zur Beurteilung der Vollständigkeit und Disjunktheit der Archetypen sowie zur Untersuchung des Unterschieds beim Trainieren von Klassifikatoren mit Volltexten gegenüber Archetypen.

Die Ergebnisse dieser Studie zeigen, dass das Trainieren von Klassifikatoren mit Domänen-Archetypen bessere Ergebnisse liefert als das Trainieren mit Volltext. Darüber hinaus übertrifft das Modell Meta-Llama-3-8B-Instruct, obwohl es als leichtgewichtiges LLM gilt, größere Modelle bei der Kontextwissen-Destillation und Archetypenerstellung. Es gibt einige Einschränkungen, wie etwa Längen-Bias bei menschlichen Bewertungen, Inkonsistenzen zwischen subjektiven Einschätzungen und Klassifikationsmetriken sowie die eingeschränkte Reichweite der Datensätze. Längere generierte Archetypen wurden von Umfrageteilnehmenden häufiger bevorzugt, möglicherweise aufgrund der Gleichsetzung von Ausführlichkeit mit Qualität. Die Ergebnisse sind möglicherweise nicht auf feinere Domänen außerhalb von Nachrichten übertragbar. Zukünftige Arbeiten sollten sich auf nuancierte, mehrsprachige

Korpora konzentrieren.

Insgesamt leistet diese Forschung einen Beitrag zur Entwicklung und Bewertung eines automatisierten, domänenexpertengesteuerten Ansatzes zur Archetypen-Erstellung. Diese Methode erfasst domänenspezifisches Wissen präzise und verbessert die Qualität nachfolgender Textklassifikationen. Die Studie erfüllt erfolgreich ihr Ziel, unstrukturierte Textdaten in strukturierte und annotierte Datensätze zu überführen und so die Entwicklung leistungsfähigerer KI-Anwendungen zu unterstützen.

Contents

| | |
|--|-----------|
| Acknowledgments | iv |
| Abstract | v |
| Kurzfassung | vi |
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Table of Important Definitions | 2 |
| 1.3 Thesis Focus: Contextual Knowledge Distillation | 3 |
| 1.4 Research Objectives & Questions | 3 |
| 2 Foundations | 5 |
| 2.1 AI Advances Driving Natural Language Processing Innovation | 5 |
| 2.2 Text Classification Across Domains | 6 |
| 2.3 Large Language Models | 7 |
| 2.3.1 Introduction to LLMs | 7 |
| 2.3.2 LLMs Quantization | 9 |
| 2.3.3 Impact on Text Classification | 11 |
| 2.4 Prompt Engineering Strategies for Optimal LLM Responses | 13 |
| 2.4.1 Zero-Shot and Few-Shot Prompting | 14 |
| 2.4.2 Cloze-Based Prompting | 15 |
| 2.4.3 Chain-of-Thought Prompting | 16 |
| 2.5 Semantic Compression and Distillation | 17 |
| 2.5.1 Extractive Summarization | 17 |
| 2.5.2 Abstractive Summarization | 17 |
| 2.5.3 Contextual Distillation | 18 |
| 2.6 Clustering Techniques | 19 |
| 2.6.1 K-Means Clustering | 19 |
| 2.6.2 Hierarchical Clustering | 20 |
| 2.6.3 Density Peaks Clustering | 20 |
| 2.7 Semantic Knowledge Representations | 21 |
| 2.7.1 Embeddings and Vector Representations | 21 |
| 2.8 AI-Generated Text Evaluation Methods | 22 |
| 2.8.1 Human-Centric Evaluation | 23 |
| 2.8.2 Automatic Metrics | 23 |
| 2.8.3 Machine-Learned Metrics | 23 |

| | | |
|----------|--|-----------|
| 3 | Related Work | 25 |
| 3.1 | Context Windowing for Semantic Extraction | 25 |
| 3.1.1 | Importance and Utility of Context Windows | 25 |
| 3.1.2 | Context Window Extraction Approaches | 26 |
| 3.1.3 | Temporal Dynamics in Context Windowing | 27 |
| 3.1.4 | Applications in Information Extraction | 27 |
| 3.2 | Semantic Grouping via Clustering | 27 |
| 3.2.1 | Clustering with Contextual Embeddings | 28 |
| 3.2.2 | Traditional and Semantic Clustering Approaches | 28 |
| 3.3 | Leveraging LLMs for Text Summarization | 29 |
| 3.3.1 | Key Advantages of LLMs in ATS | 29 |
| 3.3.2 | Categories of LLM-Based Summarization Techniques | 30 |
| 3.3.3 | Taxonomy and Trends in LLM-based Summarization Research | 30 |
| 3.3.4 | Uncertainty-Aware Summarization with LLMs | 30 |
| 3.3.5 | Aspect-Based Summarization via Fine-tuned LLMs | 30 |
| 3.3.6 | Multi-LLM Summarization Frameworks | 31 |
| 3.3.7 | Topic-Driven Summarization (TDS) | 31 |
| 3.3.8 | Evaluation and Limitations | 31 |
| 3.4 | Text Classification Applications & Hurdles | 31 |
| 3.4.1 | Applications of Text Classification | 31 |
| 3.4.2 | Challenges and Future Directions in Text Classification | 32 |
| | | |
| 4 | Methodology | 34 |
| 4.1 | Pipeline Overview | 34 |
| 4.2 | Datasets | 36 |
| 4.2.1 | OnlySports Dataset | 36 |
| 4.2.2 | AG News | 37 |
| 4.2.3 | BBC News | 37 |
| 4.2.4 | 20 Newsgroups | 37 |
| 4.2.5 | HuffPost News Category Dataset (Short Descriptions) | 37 |
| 4.3 | Text Embedding Models | 37 |
| 4.3.1 | Jina AI Embeddings v3 | 38 |
| 4.3.2 | MPNet Base v2 | 38 |
| 4.4 | Domain Context Windows Clustering | 38 |
| 4.4.1 | Recursive Hierarchical Clustering: Full vs. Selected Windows | 38 |
| 4.4.2 | Density Peaks Clustering | 39 |
| 4.5 | Generation of Domain-Specific Archetypes | 40 |
| 4.5.1 | Prompt Design | 40 |
| 4.5.2 | Utilized LLMs & Quantization | 41 |
| 4.5.3 | Batch-Wise Cluster Grouping for Token Efficiency | 43 |
| 4.5.4 | Output Parsing and Validation | 43 |
| 4.6 | Evaluation Techniques | 45 |
| 4.6.1 | Semantic Search | 46 |

| | | |
|----------|---|-----------|
| 4.6.2 | Fine-Tuning Text Classifiers on Archetypes vs. Full Text | 47 |
| 4.6.3 | Human Survey & Feedback | 49 |
| 4.6.4 | Participant Recruitment via Prolific | 49 |
| 4.6.5 | Clustering of Generated Archetypes vs. Full Text | 52 |
| 5 | Results | 56 |
| 5.1 | Semantic Search Evaluation Results | 56 |
| 5.1.1 | BBC Dataset Generated Archetypes vs. Full-Text | 57 |
| 5.1.2 | AG Dataset Generated Archetypes vs. Full-Text | 57 |
| 5.2 | Initial Text Classifiers Comparison | 58 |
| 5.3 | BBC Dataset Fine-Tuning: Roberta-Base Text Classifier | 59 |
| 5.3.1 | AG Dataset | 59 |
| 5.3.2 | Only Sport Dataset | 60 |
| 5.3.3 | Short descriptions Dataset | 61 |
| 5.3.4 | BBC Dataset | 62 |
| 5.3.5 | 20 News Groups Dataset | 63 |
| 5.4 | AG Dataset Fine-Tuning: Roberta-Base Text Classifier | 64 |
| 5.4.1 | AG Dataset | 64 |
| 5.4.2 | Only Sport Dataset | 65 |
| 5.4.3 | Short descriptions Dataset | 66 |
| 5.4.4 | BBC Dataset | 67 |
| 5.4.5 | 20 News Groups Dataset | 68 |
| 5.5 | Computational Efficiency of Fine-Tuning: Full-Text vs. Archetypes | 69 |
| 5.6 | Survey Analysis | 70 |
| 5.6.1 | Overall LLMs' Archetypes Preferences | 70 |
| 5.6.2 | Averaged Archetypes' Interpretability, Relevance, and Completeness | 70 |
| 5.6.3 | Participants Feedback Summary | 70 |
| 5.7 | Domains' Disjointness Comparison: Archetypes vs. Full-Text Clustering | 70 |
| 6 | Discussion | 78 |
| 6.1 | Study Implications | 78 |
| 6.2 | Results Interpretations | 79 |
| 6.2.1 | Insights from LLMs Performance | 79 |
| 6.2.2 | Classification Performance: Domain Full Text vs. Generated Archetypes | 80 |
| 6.2.3 | Archetypes Interpretability, Domain Relevance, and Completeness | 81 |
| 6.2.4 | Analysis of Domain Full Text vs. Archetypes Disjointness | 82 |
| 6.3 | Limitations | 83 |
| 6.4 | Future Work | 84 |
| 7 | Conclusion | 85 |
| | List of Figures | 87 |
| | List of Tables | 89 |

Bibliography

91

1 Introduction

1.1 Motivation

The ever-growing volume of unstructured digital textual data content across domains has created an urgent need for systems that can extract and structure semantic knowledge in an interpretable manner. However, the unlimited masses and redundancy of available digital text make manual analysis infeasible. Moreover, semantic boundaries within domains are rarely cleanly defined concepts. They often blend, overlap, and evolve, making it difficult to design consistent class definitions or representative summaries without deep domain expertise.

Large Language Models (LLMs) offer promising capabilities for semantic understanding and text generation, but they present their own set of limitations. When exposed to large and unstructured inputs, they may overfit to dominant themes and overlook less frequent but semantically important ideas. This leads to incomplete or biased representations, specifically difficult to organize into well-defined conceptual categories or domains. In addition, prompting LLMs effectively to produce useful and representative abstractions remains a non-trivial task specially when dealing with domain-specific context that is dense or nuanced.

Another significant challenge lies in evaluating the outcomes of such semantic processing. Unlike numeric classification or retrieval tasks, the quality of a generated set of text-based representations is hard to quantify. There is no clear ground truth, and common evaluation metrics often fall short of capturing whether a generated representation is both semantically complete and disjoint from others within the same domain or other domains.

This thesis is driven by the urge to bridge these gaps by organizing semantically similar slices of text into meaningful and structured representations that can act as archetypes of a domain's conceptual space. The approach aims to reduce noise, capture the full diversity of sub-ideas within a domain, and enable effective interaction with downstream tasks such as semantic search and classification.

The ultimate objective of this work is to develop a framework that enables the automated understanding of domain-specific knowledge by leveraging both the semantic structure of text and the generative capabilities of LLMs. The motivation lies in advancing the boundaries of domain text overlapping semantic contexts into concise, discriminative, and comprehensive archetypes that are interpretable by both humans and machines. Central to this effort is the challenge of distilling semantically similar context windows, each encapsulating specific ideas or concepts into a set of disjoint and complete class archetypes. These archetypes serve as compact, representative abstractions of the underlying domain concepts, offering structured, domain-aligned knowledge units. The proposed work does not only support deeper semantic understanding but also contributes to the broader goal of structured knowledge extraction and representation from unstructured text.

In doing so, the framework addresses several interconnected challenges:

- Reducing the redundancy and semantic noise inherent in large corpora
- Structuring raw LLM outputs into clearer and more interpretable forms
- Enabling classification and understanding without requiring predefined expert-driven taxonomies

It is also worth-mentioning that CreateData4AI (CD4AI) ¹ is one of the projects that motivated the carrying out of this research, as it represents an innovative solution focusing on transforming unstructured text into structured and annotated datasets that are systematically classified according to specific features. The process encompasses several crucial steps, including **Context Rule Creation**. After acquiring several context windows for each predefined class, the domain expert assesses and identifies which windows aptly capture the essence of the predefined classes. However, the best solution is to avoid manual work and automate the process. That is why no human intervention is preferred, and, at the same time, the underlying domain experts' knowledge could be acquired via well-trained LLMs. These archetypical rules set the foundation for automated data creation.

Through this approach, the thesis proposes a pathway toward scalable and human-aligned semantic understanding that combines the expressive power of LLMs with principled abstraction mechanisms to make sense of unstructured data.

1.2 Table of Important Definitions

| Term | Definition |
|------------------------------|--|
| Context Window | A bounded segment of text that surrounds a specific keyword or topic, capturing semantic relevance to support downstream tasks such as clustering and classification. |
| Class Archetype | A compact and representative abstraction of semantically similar context windows that encapsulates a subdomain or subclass concept. Archetypes serve as distilled knowledge units used for classification. |
| Context Rule Creation | A three-phase process that includes semantic clustering of context windows, generation of archetypes using LLMs, and evaluation of the generated archetypes to ensure disjointness and completeness. |

¹CD4AI

| Term | Definition |
|--------------------------------|---|
| Semantic Completeness | The extent to which generated archetypes comprehensively cover all relevant sub-domains or sub-ideas in the context windows of a domain. |
| Semantic Disjointness | A measure of how well archetypes are non-overlapping, ensuring that each set of domain's archetypes represent a domain with minimal redundancy with respect to other domains. |
| Contextual Distillation | A process of transforming semantically similar context windows into structured representations (archetypes), emphasizing non-redundant, domain-specific knowledge. |
| Semantic Search | An evaluation method that checks the semantic similarity between user queries and corpus |

1.3 Thesis Focus: Contextual Knowledge Distillation

This thesis focuses on the **Context Rule Creation** step, which covers three main sub-steps. The first phase is to group the huge number of domain context windows which have been extracted and are already present per class. The grouping would be in terms of semantic grouping of the context windows so that, each group would represent a subdomain, subclass, or sub-idea for the main domain. This would be achieved using clustering approaches, and subsequently this would prepare the context windows for the next step.

The second phase is about how the clustered context windows would be utilized and formatted as the class archetypes. The leading solution is make use of generative AI, design the optimal prompt, then feed it to various LLMs specifically the light-weight ones to generate archetypes not per the whole domain's context windows, but per cluster of context windows generated in the former step, representing a sub-domain. That approach guarantees that each domain sub-idea is addressed, and never overlooked as much as possible by the LLM, instead of passing the whole domain context windows at a one time to the LLM. Therefore, ensuring higher percentage of semantic completeness of the generated class archetypes.

The third phase concerns testing and evaluating of the generated domain archetypes, which are the outcomes of the second phase. A challenge occurs because the text outcome is not measurable. What is crucial to be tested is whether each class has a set of archetypes, which completely and disjointly represent the corresponding domain. Evaluation would include downstream tasks, text semantic search, fine-tuning text classifiers, and conducting a survey.

1.4 Research Objectives & Questions

This thesis looks into the subsequent research questions:

1. What are the most effective methods to accurately capture semantics from related text chunks and distill them into one coherent text?

2. What are the current prompt engineering techniques crucial for generating well-defined, coherent class archetypes that effectively incorporate domain expertise?
3. How can the quality and consistency of the generated class archetypes be evaluated and ensured?
4. Does the downstream utility of generated archetypes achieve similar results to manually annotated datasets in domain-specific tasks?

Diving deep into these questions leads to achieving the main goal of this thesis: supporting domain experts in the definition of classes, particularly creating archetypes for all predefined classes to be used for classifying any text paragraph. Moreover, it aids the CD4AI project in achieving its goal of forming structured datasets suitable as input to machine learning models from unstructured data.

2 Foundations

This chapter outlines the grounds behind the current research methodology. It delves into the principal concepts and techniques, which are crucial for the three aforementioned phases. This chapter covers several key points starting with recent AI advancements and contributions in natural language processing (NLP) field. Moreover, the paramount importance of the text classification, and its vital contributions and effectiveness in various real-life applications. Additionally, this section covers the different types of text summarization and semantic distillation. It also introduces LLMs with a discussion about the latest advanced prompt engineering techniques. Furthermore, the section covers a few top clustering techniques along with some representations for text semantics. Since the evaluation of LLMs generated text is not that easy, some existing methods would be introduced.

2.1 AI Advances Driving Natural Language Processing Innovation

Recent advances in artificial intelligence (AI) are fundamentally transforming the field of NLP, reshaping how machines interpret, generate, and interact with human language. As illustrated in Figure 2.1, the NLP market is projected to experience substantial revenue growth over time, reflecting the increasing integration of AI-powered technologies. In parallel growth to this transformation is the emergence of LLMs, which have proven pivotal due to their scalability, accessibility, and efficiency.

Tools such as ChatGPT exemplify the wide-ranging impact of these models, garnering significant academic and public interest for their ability to generate natural, creative, and context-aware responses. Their applications span diverse domains, including education, research, healthcare, marketing, and customer service [1]. These AI-driven advancements are lowering the technical barriers to working with complex textual data, empowering both individuals and organizations regardless of technical expertise to automate tasks such as writing, translation, summarization, information retrieval, and decision-making. As a result, productivity and innovation are being significantly enhanced across sectors.

Moreover, the integration of AI into NLP has catalyzed the disclosure of new sub-fields and the refinement of existing ones. Areas such as explainable AI for language models, controllable text generation, and ethical NLP focusing on challenges like bias, misinformation, and fairness have gained increasing prominence. These shifts underscore AI's role not merely as a tool but as a transformative force driving both conceptual and methodological evolution in NLP [2]. In addition to expanding the scope of NLP, AI has markedly improved the accuracy and efficiency of core linguistic tasks. Modern neural architectures now underpin critical processes such as part-of-speech tagging, syntactic parsing, and named entity recognition, leading

to more adaptable and context-aware systems. Transformer-based models, which leverage attention mechanisms and large-scale pre-training, have set new performance benchmarks by learning deep, bidirectional representations of language [3].

The generative capabilities of these models have further enabled breakthroughs in real-time dialogue systems, personalized content creation, and specialized applications in areas like healthcare diagnostics and financial analytics [4]. By processing sequential and contextual data with unprecedented accuracy, transformer models have significantly enhanced machines' ability to understand linguistic nuances, idiomatic expressions, sarcasm, and even cross-lingual semantics [5]. These innovations have made NLP systems more intelligent, adaptive, and capable of engaging with human language in a profoundly contextualized and meaningful way.

Ultimately, the ongoing AI-driven progress in NLP is not only reshaping the technical landscape but also amplifying its societal impact. The increasing indispensability of AI in language technologies highlights its foundational role in shaping the present and future of human-machine communication.

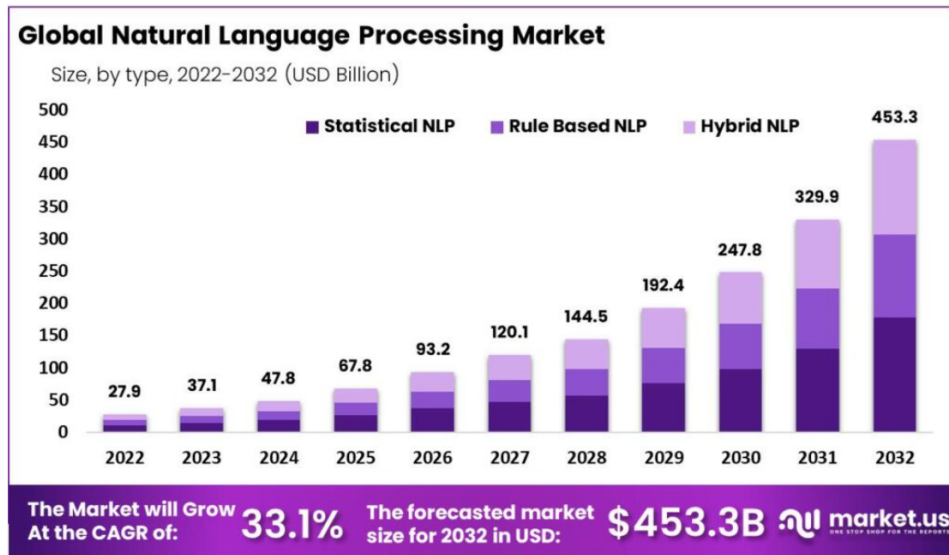


Figure 2.1: NLP Market Forecasted Revenue [6]

2.2 Text Classification Across Domains

Text classification is a foundational task in NLP, underpinning a wide array of real-world applications by enabling the automated assignment of predefined categories to textual data. This automation is crucial for managing and extracting value from the growing volume of unstructured information generated across domains [7]. In the financial sector, for instance, institutions routinely process large quantities of documents such as prospectuses, statements, and regulatory filings which must be accurately classified to ensure compliance

and operational efficiency. However, manual classification remains common in industrial workflows, often leading to costly errors. For example, JP Morgan reported that 80% of its loan servicing errors were caused by manual contract misclassification [8].

The practical relevance of text classification spans various use cases:

- **Sentiment Analysis** is widely applied in business and social media to assess public opinion on products, services, or events.
- **Topic Labeling** is essential for content categorization in news articles, academic literature, and document organization.
- **Question Answering and Dialog Act Classification** are pivotal in customer support systems, virtual assistants, and interactive applications.

Manual classification of large-scale text data is not only labor-intensive and time-consuming but also susceptible to inconsistency due to human fatigue and bias. Automating this process using machine learning models enables scalable, objective, and consistent classification. Moreover, effective text classification enhances information retrieval, reduces information overload, and supports downstream tasks such as content filtering, spam detection, knowledge management, and personalized recommendations [7].

Advancements in classification algorithms especially with the advent of deep learning and attention-based architectures have significantly improved accuracy across domains with diverse data structures and complexities. The availability of hand-annotated datasets further accelerates research in this field, enabling the development of scalable, cross-domain solutions for content analysis [9]. Among the various subfields, cross-domain text classification has emerged as a critical area, particularly in sentiment analysis. This is due to both its broad applicability and the increasing availability of benchmark datasets. Nonetheless, despite the impressive capabilities of large pretrained language models, their performance often degrades in unfamiliar domains unless fine-tuned with costly, domain-specific labeled data [10].

2.3 Large Language Models

Language is a fundamental tool of human communication, enabling the expression of ideas, emotions, and information. However, machines lack the innate capacity to understand or generate human language, necessitating the development of advanced AI techniques. Within the field of NLP, LMs have been developed over years as in Figure 2.2 as essential technologies for capturing linguistic patterns and generating coherent, context-aware text [11].

2.3.1 Introduction to LLMs

LLMs represent a transformative leap in language modeling, building on earlier paradigms such as Statistical Language Models (SLMs), which employed probabilistic methods to predict word sequences, and Neural Language Models (NLMs), which utilized neural networks to

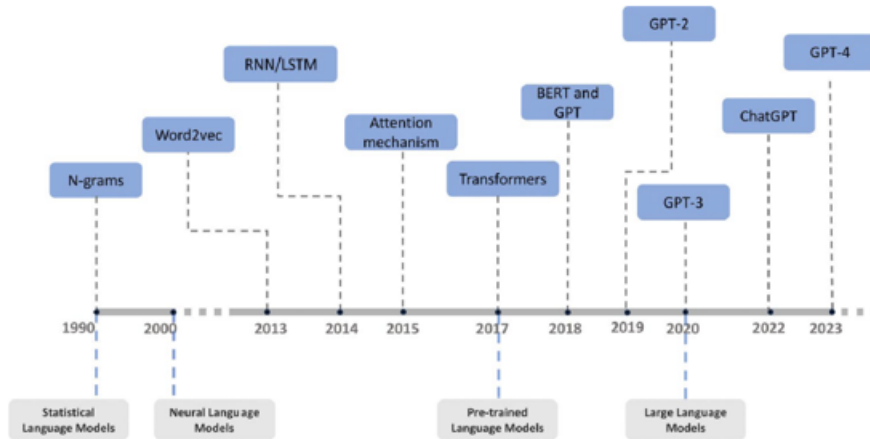


Figure 2.2: LMs Development Over Past Years

learn semantic representations. These were followed by Pre-trained Language Models (PLMs), which leveraged large-scale corpora and self-supervised learning to achieve generalized language understanding. LLMs extend PLMs by incorporating significantly larger datasets, vast computational resources, and advanced training algorithms, resulting in models that can perform complex language tasks with human-like fluency and precision [11].

The architecture of LLMs typically involves two main training phases: a large-scale pre-training stage using general text corpora, followed by a fine-tuning or alignment phase using human feedback or task-specific data. These models often exceed hundreds of billions of parameters and are trained on datasets spanning hundreds of gigabytes to terabytes. Their versatility is reflected in a wide range of applications, including translation, summarization, code generation, and question answering. The development of LLMs has been further propelled by innovations in transformer architectures, which enable effective modeling of long-range dependencies in text [11]. Recent advancements exemplified by the Generative Pre-trained Transformer (GPT) series have demonstrated LLMs' capacity to understand, process, and generate language across diverse domains. Despite their growing presence in both industry and academia, many users still lack a foundational understanding of how these models function, highlighting the need for accessible overviews of their historical development, core principles, and capabilities [11].

A capable LLM should demonstrate four foundational features [12]:

1. Deep comprehension of natural language context,
2. Human-like text generation,
3. Contextual awareness, especially in knowledge-intensive tasks,
4. Robust instruction-following abilities for problem-solving and decision-making.

Several prominent LLMs have been released in recent years, including OpenAI’s ChatGPT, Meta AI’s Llama, and Databricks’ Dolly 2.0. These models have seen rapid adoption across domains such as customer support, education, translation, finance, healthcare, and software development. For example, ChatGPT had surpassed 180 million users by late 2023 [12]. Beyond mainstream NLP applications, LLMs have sparked interest in fields like security and privacy due to their capacity to reason over complex inputs. Their use cases now extend far beyond text generation, underlining their adaptability and the growing importance of understanding their foundational mechanics, broader implications, and associated risks [12].

2.3.2 LLMs Quantization

As the scale and complexity of LLMs continue to increase, deploying them efficiently on resource-constrained environments such as edge devices and personal hardware has become an interesting challenge. Quantization emerges as a critical technique for mitigating this challenge by reducing the bit precision of model parameters, thereby significantly compressing the model size and accelerating inference, while aiming to maintain the original model’s performance.

Modern LLMs consist of hundreds of billions of parameters, often requiring upwards of hundreds of gigabytes of memory even in reduced FP16 formats. Such memory demands are incompatible with the capabilities of most edge devices, thereby inhibiting on-device deployment. Quantization addresses this constraint by representing model weights using lower-precision integers (e.g., 8-bit or 4-bit), substantially reducing both memory footprint and data movement overhead. This not only speeds up inference but also brings deployment within the realm of feasibility for a broader range of hardware.

On-device deployment via quantized models yields several benefits: reduced inference latency, enhanced user privacy through local processing, and reduced reliance on centralized cloud infrastructure translating to lower operational costs and improved scalability. Yet, these benefits hinge on the quantization technique’s ability to preserve the generalization and robustness of the original LLM across diverse tasks and domains, while maintaining computational efficiency.

Quantization methods are broadly categorized into two families as outlined in [13]:

- **Quantization-Aware Training (QAT):** This approach integrates quantization into the training pipeline itself, using backpropagation to simulate quantization effects. Although QAT achieves high post-quantization accuracy, especially in low-bit regimes, it demands extensive computations and is often impractical for massive LLMs due to their training cost and memory requirements.
- **Post-Training Quantization (PTQ):** PTQ applies quantization to pre-trained models without retraining. While significantly more scalable and attractive for real-world deployment, especially by end-users, PTQ often suffers from performance degradation particularly in aggressive quantization settings such as 3-bit or 4-bit precision.

A spectrum of quantization schemes has been proposed, each offering trade-offs between computational efficiency and representational fidelity [14]:

- **Uniform Scalar Quantization:** This basic approach maps each parameter to a fixed grid defined by a global scale factor and offset. While highly efficient, its inability to model the often non-uniform distribution of LLM weights leads to quantization noise and accuracy loss.
- **Non-Uniform Scalar Quantization:** This method improves flexibility by using a learned codebook, where each weight is replaced by its nearest centroid. The quantization grid is thus better adapted to the underlying data distribution, however at the cost of additional storage for the codebook and increased computational complexity.
- **Vector Quantization (VQ):** VQ generalizes scalar quantization by encoding weight vectors (e.g., 2D or 4D) using multi-dimensional centroids. This approach captures inter-weight correlations, leading to denser and more expressive quantization with superior signal-to-quantization-noise ratio (SQNR). Studies demonstrate that higher-dimensional VQ significantly enhances performance under low-bit constraints.

Quantization strategies can further be divided based on whether they require calibration [15]:

- **Zero-Shot Quantization:** Techniques such as LLM.int8(), NF4, and FP4 fall under this category. They perform quantization without task-specific calibration data or fine-tuning, relying instead on simple, efficient schemes like block-wise scaling and rounding to predefined alphabets (e.g., mapping to values in $[-1, 1]$). Their minimal computational overhead and ease of use have facilitated wide adoption, as seen in tools like Hugging Face Transformers.
- **Optimization-Based Quantization:** These methods involve learning quantization parameters such as scale factors, offsets, and codebooks by minimizing quantization error on calibration datasets. Though computationally expensive, this yields higher quality quantized models with better retention of downstream task performance. Such techniques are often used by model developers prior to releasing publicly available quantized variants.

The widespread use of quantization has played a pivotal role in enabling LLM inference on hardware and edge devices without major trade-offs in benchmark performance. Nonetheless, a critical area remains underexplored: *the security and robustness implications* of quantization. Most existing research has primarily focused on utility and accuracy, leaving potential vulnerabilities such as susceptibility to adversarial perturbations or quantization-induced bias relatively neglected [15].

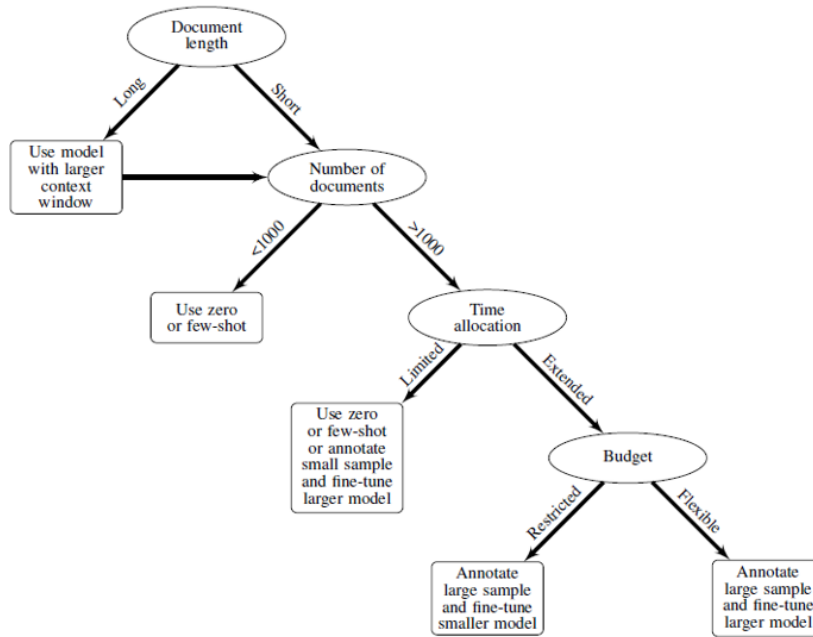


Figure 2.3: LLMs Approach for Text Classification

2.3.3 Impact on Text Classification

LLMs, particularly those based on transformer architectures, have significantly reshaped the field of text classification in recent years. Initially exemplified by models such as Bidirectional Encoder Representations from Transformers (BERT) and GPT, and more recently advanced by GPT-4, Llama, and BloombergGPT, LLMs have exhibited exceptional abilities in capturing contextual and semantic nuances, making them powerful tools for diverse classification tasks [16] as shown in Figure 2.3 [17]. This evolution marks a decisive shift away from traditional rule-based and dictionary-based approaches toward sophisticated, data-driven methods. Pre-trained LLMs like BERT, RoBERTa (A Robustly Optimized BERT Pretraining Approach), and DeBERTa leverage deep learning to model language with high contextual sensitivity, outperforming earlier techniques such as bag-of-words or word embeddings [18]. These encoder-based models have proven particularly effective when fine-tuned on task-specific data, leading to substantial performance gains in complex text classification scenarios.

A comprehensive study by Bucher et al. [18] compared the performance of fine-tuned small-scale LLMs (e.g., RoBERTa, DeBERTa, XLNet) with larger generative models (e.g., ChatGPT-3.5/4, Claude Opus) across various classification tasks and text genres. Results show that fine-tuned models consistently outperform generative counterparts, especially in specialized applications such as stance detection and emotion classification in political discourse. While generative LLMs offer ease of use via prompt-based interactions, they lack the task-specific optimization that supervised fine-tuning provides. The study also highlights that even modest amounts of labeled data (around 200–500 samples) are sufficient for small LLMs to surpass zero-shot generative models in metrics like accuracy and F1-score.

These advantages are particularly pronounced in imbalanced or domain-specific datasets. An open-source toolkit developed by the authors simplifies the training and evaluation process using Hugging Face libraries, making fine-tuning more accessible.

Despite the growing capabilities of generative LLMs, fine-tuned encoder-based models remain the preferred choice for many classification tasks due to their performance, efficiency, and controllability. Smaller models further offer benefits in terms of data privacy, transparency, and on-device deployment—key considerations in sensitive or regulated environments [18]. While generative models may close the gap through advances in few-shot learning and prompt engineering, fine-tuned LLMs currently deliver superior domain-specific performance.

Chae et al. [17] reinforce these findings by demonstrating that LLMs:

- Outperform traditional machine learning models (e.g., SVMs using BoW or embeddings) across various classification benchmarks.
- Excel in both single- and multi-target tasks, modeling nuanced linguistic patterns.
- Deliver strong zero-shot performance (e.g., GPT-3 Davinci) when effectively prompted, though results vary with prompt structure.
- Achieve notable accuracy improvements with fine-tuning, especially smaller models like BERT or GPT-3 Ada, after training on moderate-sized datasets (1000–2000 examples).
- Are sensitive to training data composition and prompt design, which can introduce systematic biases particularly in socio-political contexts.
- Enable low-cost, scalable text classification through APIs, though this raises concerns about reproducibility and transparency when using proprietary models.

These capabilities offer researchers a flexible, high-performance toolkit for domain-specific classification with limited supervision, underscoring the value of LLMs as both predictive and exploratory tools [17].

While general-purpose LLMs like GPT-4, Llama 2, and ChatGLM 2 have shown remarkable capabilities through instruction tuning and in-context learning, they often lag behind specialized PLMs such as RoBERTa and DeBERTa in classification benchmarks [19]. This discrepancy suggests that direct use of generative models for classification may not be optimal without additional adaptation. To address this, Zhang et al. [19] proposed RGPT, a boosting-based framework that iteratively fine-tunes and ensembles LLMs to enhance classification performance. Unlike traditional prompt engineering, RGPT dynamically adjusts sample distributions and incorporates historical prediction errors, resulting in more robust, task-adapted models. Experimental results show that RGPT outperforms both state-of-the-art PLMs and LLMs across four benchmark datasets, with gains of up to 1.88% over the previous best models.

Human evaluations further reveal that RGPT achieves classification accuracy exceeding the average human annotator while requiring significantly less time. Ablation studies confirm

that boosting and ensembling strategies unlock additional performance benefits beyond what is achievable through individual fine-tuning [19].

In the context of hierarchical text classification, LLMs face additional challenges due to the complexity of structured label taxonomies. Models like GPT-4 and Claude, while effective in flat classification tasks, struggle when handling extensive class hierarchies in zero-shot settings due to prompt length limitations and increased inference cost. To overcome these limitations, TELEClass integrates LLMs with corpus-specific knowledge, offering a structure-aware approach for low-supervision classification [20]. TELEClass enhances label understanding via:

- **Taxonomy Enrichment:** LLMs generate class-indicative keywords (e.g., distinguishing "conditioner" with terms like "moisture" or "soft hair" from "shampoo").
- **Core Class Annotation & Data Augmentation:** LLMs assist in annotating unlabeled documents and generating synthetic examples for underrepresented classes, reducing annotation burdens and improving generalization.

TELEClass achieves results on par with or better than GPT-4 across datasets like Amazon-531 and DBPedia-298 while drastically reducing inference cost. The study concludes that targeted integration of LLMs into hierarchical classification frameworks can unlock significant gains over zero-shot prompting and traditional weakly supervised techniques [20].

2.4 Prompt Engineering Strategies for Optimal LLM Responses

Prompt engineering involves designing and refining input queries to optimize interactions with LLMs. This approach enhances the model's ability to generate accurate, contextually relevant, and meaningful responses, significantly influencing its effectiveness in various applications. Practical prompt engineering hinges on clarity, precision, and contextual relevance. Some foundational principles include:

- **Define Objectives Clearly:** Articulate the purpose of the prompt to guide the model's response effectively. For instance, specifying the desired output format or tone can improve the result's alignment with user expectations [21].
- **Use Iterative Refinement:** Iterating on prompts to refine outputs is essential. Rephrasing, adding context, and providing examples help achieve optimal results [22].
- **Incorporate Context:** Contextualizing prompts ensures that the model understands the task better. For example, setting a role or scenario allows the AI to generate responses tailored to specific needs [23].
- **Adopt Frameworks:** Frameworks like the Query Transformation Module (QTM) break down input sentences into objectives and key points, improving comprehension and output quality. Techniques such as zero-shot, few-shot, and cloze-based prompting adapt the model's capabilities to specific tasks [24].

Beyond basic principles, advanced techniques have emerged to enhance the depth and precision of LLM outputs:

- **Chain-of-Thought Prompting:** Encourages the model to process complex tasks step-by-step, improving reasoning and accuracy [22].
- **Role-Based Prompts:** Assigning specific roles to the AI (e.g., “Act as a historian”) helps tailor responses to the task’s nuances [21].
- **Feedback Loops:** Engaging the model iteratively with feedback improves current outputs and informs future interactions [23].

Numerous challenges can arise when formulating prompts, such as preventing model *hallucinations* (i.e., generating inaccurate or fabricated data) and addressing response biases. Ethical practices including transparency in AI interactions and adherence to privacy guidelines are paramount [21].

Prompt engineering has demonstrated utility across various domains, including education, where it enables the co-creation of instructional content, and conversational AI, where it optimizes chatbot interactions. Techniques like retrieval-augmented generation and purpose-specific queries have expanded its potential to generate domain-specific outputs with minimal data [24].

2.4.1 Zero-Shot and Few-Shot Prompting

Zero-shot and few-shot learning are essential techniques in prompt engineering, enabling LLMs to perform tasks with minimal or no task-specific training. These approaches enhance the versatility of generative AI by leveraging the model’s pre-trained knowledge and adaptability.

Zero-shot learning allows models to perform tasks without specific examples or prior task-specific training. Instead, the model relies solely on its pre-trained understanding to generate responses based on the prompt. This method is beneficial for general tasks or scenarios where providing examples is impractical. Zero-shot learning demonstrates the model’s ability to generalize across tasks using only its foundational training [24]. For example, when provided with a query like, “Summarize the main points of this article”, the model generates a response without additional context or examples. The effectiveness of the technique depends heavily on the clarity and specificity of the prompt, highlighting the need for precise prompt engineering [23].

Few-shot learning takes this a step further by incorporating several examples to guide the model’s behavior. These examples provide contextual cues that help the model better understand the task and produce more accurate output. Few-shot learning bridges the gap between zero-shot learning and extensive task-specific training, making it a flexible and resource-efficient approach [24]. An example of a few-shot prompting might involve asking the model to translate a sentence into a target language, accompanied by a few example

translations. This setup allows the model to infer the desired format and output style, improving the quality of its response[24].

Although both techniques work well, their application depends on the complexity of the task. Zero-shot learning is ideal for general or straightforward queries, while few-shot learning excels in scenarios requiring a more nuanced understanding or adherence to specific formats[23]. In educational contexts, for example, zero-shot learning can generate lesson summaries, while few-shot learning can refine those outputs by providing examples of preferred styles or structures. These approaches are instrumental in developing conversational models, summarization tools, and content generation systems[24].

Both zero-shot and few-shot learning rely heavily on the quality of the prompt. Poorly constructed prompts can lead to irrelevant or nonsensical outputs. Moreover, ethical considerations, such as ensuring accurate and unbiased information, remain critical[23]. Zero-shot and Few-shot learning represent foundational techniques in prompt engineering, offering robust solutions to leverage LLMs in diverse applications. Their effectiveness underscores the importance of carefully designed prompts and iterative refinement processes.

2.4.2 Cloze-Based Prompting

Cloze-based prompting is a specific technique in prompt engineering that involves creating fill-in-the-blank style prompts to guide LLMs toward generating precise and relevant outputs. This approach is convenient in tasks requiring contextual comprehension and structured responses[24]. In cloze-based prompting, a sentence or query is presented with certain parts omitted, prompting the model to complete the missing sections. This technique enables the model to focus on the specific information required to fill the gaps, leveraging its pre-trained knowledge and context understanding [21].

For example:

Query: "The Statue of Liberty is located in ____."

Model Output: "New York."

This method structures the model's task in a way that narrows its focus, improving accuracy and relevance, especially in tasks involving factual information, language learning, or knowledge retrieval [21] [24].

Cloze-based prompting is widely applicable across various domains:

Language Learning: Facilitates vocabulary building and grammar exercises by asking learners to fill in blanks, thereby reinforcing learning.

Content Creation: Aids in generating structured content, such as filling in missing data points in a predefined format or creating summaries with guided input.

Knowledge Retrieval: Supports the extraction of specific pieces of information from a dataset or knowledge base, making it useful for question-answering systems and academic research.

Despite cloze-based prompting's advantages, it requires carefully designed prompts to ensure clarity and avoid ambiguity. Poorly structured prompts may lead to irrelevant or nonsensical

outputs. Additionally, the technique relies heavily on the model's ability to infer context accurately, which may vary depending on the model's pre-training and fine-tuning[21]. As generative AI continues to evolve, cloze-based prompting techniques are expected to integrate multimodal elements (e.g., combining text with images or data tables) to address more complex tasks. Additionally, adaptive cloze-based techniques that dynamically adjust prompts based on user feedback or context could further enhance their applicability [24]. In summary, cloze-based prompting represents a versatile and effective method in prompt engineering, enabling precise and contextually relevant outputs across diverse applications. Its structured approach makes it a valuable tool for optimizing LLMs' performance in educational and professional settings.

2.4.3 Chain-of-Thought Prompting

Chain-of-thought (CoT) prompting is an advanced prompt engineering technique that enhances reasoning capabilities in LLMs. By encouraging models to state their reasoning processes step-by-step, CoT prompting improves the accuracy and depth of responses in complex tasks [22]. Chain-of-thought prompting replicates human reasoning by breaking down problems into smaller, logical steps. This structured approach enables LLMs to tackle multi-step problems that require sequential thinking. For instance, when asked to solve a math problem or generate an argument, the model is prompted to consider each part of the problem sequentially, leading to more accurate and coherent responses[23].

CoT prompting has found applications across various domains, including education, content generation, and problem-solving in technical fields. Examples include:

Mathematics and Logical Reasoning: Breaking down calculations into smaller steps to improve accuracy and transparency in problem-solving [22].

Content Structuring: Assisting in drafting essays or technical documents by generating outlines and expanding each section step-by-step[23].

Teaching and Training: Enhancing AI-based educational tools to provide step-by-step explanations can help learners grasp complex concepts more effectively [22].

By guiding the model through incremental steps, CoT prompting significantly reduces the likelihood of generating irrelevant or incorrect outputs, particularly in tasks that demand logical reasoning or contextual understanding [22]. However, CoT prompting requires well-structured prompts and careful design to ensure effectiveness. Poorly designed CoT prompts may lead to verbose or redundant outputs. Additionally, ethical considerations are vital to ensure the transparency and reliability of the reasoning process [23].

As models become more sophisticated, CoT prompting techniques are likely to evolve, incorporating multimodal reasoning (e.g., combining text, images, or numbers) and enabling deeper contextual understanding. This evolution will further expand the potential applications of CoT prompting in educational and professional settings. Chain-of-thought prompting exemplifies the intersection of structured reasoning and prompt engineering, unlocking new possibilities for complex problem-solving and content generation. By adopting CoT techniques, practitioners can harness the full potential of LLMs to address complex challenges [22].

Prompt engineering bridges the gap between human intent and AI capability, embodying both an art and a science. Its development as a disciplined practice ensures that LLMs deliver on their promise of meaningful and effective human-machine interactions [22].

2.5 Semantic Compression and Distillation

Semantic summarization or distillation is a key task in NLP, designed to condense significant texts into brief, coherent summaries while retaining the main ideas. It can be broadly divided into extractive and abstractive methods, each addressing distinct challenges and applications. Extractive summarization involves selecting key sentences directly from the source text, ensuring relevance and semantic accuracy. In contrast, abstractive summarization generates new sentences that capture the essence of the content, requiring advanced generative models for fluency and coherence. Both approaches complement each other however, there is also contextual distillation that mainly focuses on synthesizing non-redundant knowledge.

2.5.1 Extractive Summarization

Extractive summarization focuses on selecting sentences directly from the source text to create concise summaries. As discussed in [25], this approach retains the original wording, preserving semantic integrity and context. Techniques like graph-based models (e.g., LexRank, TextRank) have been widely used, leveraging sentence similarity graphs to rank sentences by centrality. These methods benefit domain-independent and large-scale applications, as they do not require labeled data. However, challenges such as maintaining coherence and reducing redundancy persist.

Recent advancements in extractive summarization have integrated knowledge distillation techniques, as highlighted in [26]. Knowledge distillation leverages soft probability targets to train smaller models, improving their generalization while capturing complex inter-sentence relationships. This approach addresses traditional methods' limitations, such as their inability to model nuanced sentence-level features.

Ensuring summary coherence and developing standardized evaluation metrics is challenging however, some efforts to tackle these issues include concept-based methods, as mentioned in [25], and innovative frameworks like the DPC, which exemplify how clustering techniques can improve summarization efficiency and quality.

2.5.2 Abstractive Summarization

Abstractive summarization aims to generate summaries that are not merely extracted fragments of the source text but reformulated sentences that encapsulate the core ideas. This method relies heavily on advanced neural architectures and generative models. As described in [27], abstractive summarization techniques increasingly utilize knowledge distillation to train smaller student models, which mimic larger teacher models through pseudo-labels and soft outputs. These methods enhance model efficiency and adaptability, even in low-resource scenarios. The task of query-focused summarization (QFS), highlighted in the same study,

exemplifies the potential of abstractive techniques. QFS models tailor summaries to specific user queries, addressing challenges like domain adaptation through PEGASUS’s q-GSG (query-aware Gap Sentence Generation) approaches. Additionally, synthetic datasets like Query-CNNM have been pivotal in mitigating data scarcity for abstractive models, enabling their application in zero-shot and few-shot settings.

In the shell of multi-document summarization (MDS), [28] emphasizes the use of sequence-to-sequence transformer models, such as BART and PEGASUS. These models excel in handling long input sequences, although challenges like truncation and information loss persist. Two-phase "extract-then-abstract" approaches have emerged as a solution, where extractive methods reduce input size before generative summarization. This strategy enhances the relevance and coherence of summaries.

Abstractive summarizaion has also facilitated advancements in long-text understanding, as explored in [29]. Using gist detection, student models distilled from abstractive summarization frameworks can effectively condense lengthy texts while maintaining semantic richness. Techniques like pseudo-labeling and the shrink-and-fine-tune (SFT) method provide computationally efficient pathways for training lightweight yet performant summarization models.

Redundancy, coherence, and domain adaptation remain central obstacles to the field. Innovations like heterogeneous graph neural networks (GNNs) and sequence-level knowledge distillation continue refining abstractive summarization, ensuring scalability and contextual accuracy in generic and specialized summarization tasks.

2.5.3 Contextual Distillation

Contextual Distillation refers to the transfer of higher-order structural and semantic relationships that go beyond direct outputs or raw features. Unlike classical knowledge distillation techniques, which typically rely on mimicking final logits or intermediate representations, contextual distillation emphasizes the relationships among features, instances, or classes such as similarities between feature vectors or inter-class structures [30].

A significant strand of this approach lies in symbolic contextual distillation, which seeks to extract the embedded contextual understanding from LLMs and represent it in interpretable forms. Rather than relying on response-based, feature-based, or relation-based methods, this form of distillation captures latent, structured knowledge and expresses it symbolically through logical rules, semantic frames, or knowledge graphs [31].

[27] introduces contextual distillation as a way that combines both abstractive and extractive summarization, specifically tailored for low-resource settings. Its method leverages a teacher-student framework that generates query-specific abstractive summaries without requiring direct supervision. It achieves this by distilling knowledge through context-aware pseudo summaries going beyond surface-level extraction to target semantic relevance, domain alignment, and summary quality via synthetic supervision.

Another important dimension of contextual distillation, especially in the shadow of LLMs, involves black-box techniques. These do not require access to internal model parameters or activations. Instead, knowledge is transferred through crafted natural language contexts such

as examples, explanations, or instructions produced by teacher models like GPT-4 or PaLM. This is particularly useful when the teacher models are closed-source or API-accessible only.

Three major paradigms exemplify this form of contextual distillation [32]:

- **In-Context Learning (ICL):** Knowledge transfer occurs through demonstrations embedded in prompts, allowing student models to mimic output behaviors from input-output pairs.
- **Chain-of-Thought (CoT):** Extends ICL by including intermediate reasoning steps or rationales, enabling students to replicate both outputs and the underlying reasoning, thus enhancing generalization.
- **Instruction Following:** Involves training student models on large-scale instruction datasets generated by teacher models. This enables the students to align with task-oriented prompts and achieve stronger zero-shot and generalization performance.

2.6 Clustering Techniques

Clustering is involved in a vast number of real-life applications as an unsupervised learning technique that enables the discovery of latent structures and patterns within unlabelled textual data. By grouping semantically or syntactically similar items such as words, sentences, documents, or embeddings, clustering facilitates more efficient organization, understanding, and processing of natural language at scale. Its strength lies in its ability to reveal hidden relationships without the need for extensive labeled datasets, making it particularly valuable in the early stages of data exploration or when working with large corpora.

A few applications utilizing clustering techniques in NLP would include topic modeling, where documents are grouped by underlying themes; document de-duplication, which helps identify similar or identical content across datasets; and word sense disambiguation, which clusters words based on contextual meaning. Additionally, clustering is foundational in building taxonomies, summarizing information, and enhancing information retrieval systems by organizing search results into coherent groups such as retrieving user-preference in recommendation systems such as in Figure 2.4 [33]. As the demand for scalable and intelligent language technologies grows, clustering remains an essential tool for uncovering structure and meaning within complex language data. Next are some crucial clustering techniques that have consistently proven their value over time.

2.6.1 K-Means Clustering

K-Means Clustering is a commonly used method for dividing a dataset into K distinct, non-overlapping subgroups or clusters. K-means clustering is well-known for its computational efficiency and ease of interpretation. However, it assumes spherical clusters and it is sensitive to the initial placement of centroids[34], which can sometimes lead to suboptimal clustering solutions. As a result, it causes a bit of a struggle to decide the estimated number of clusters.

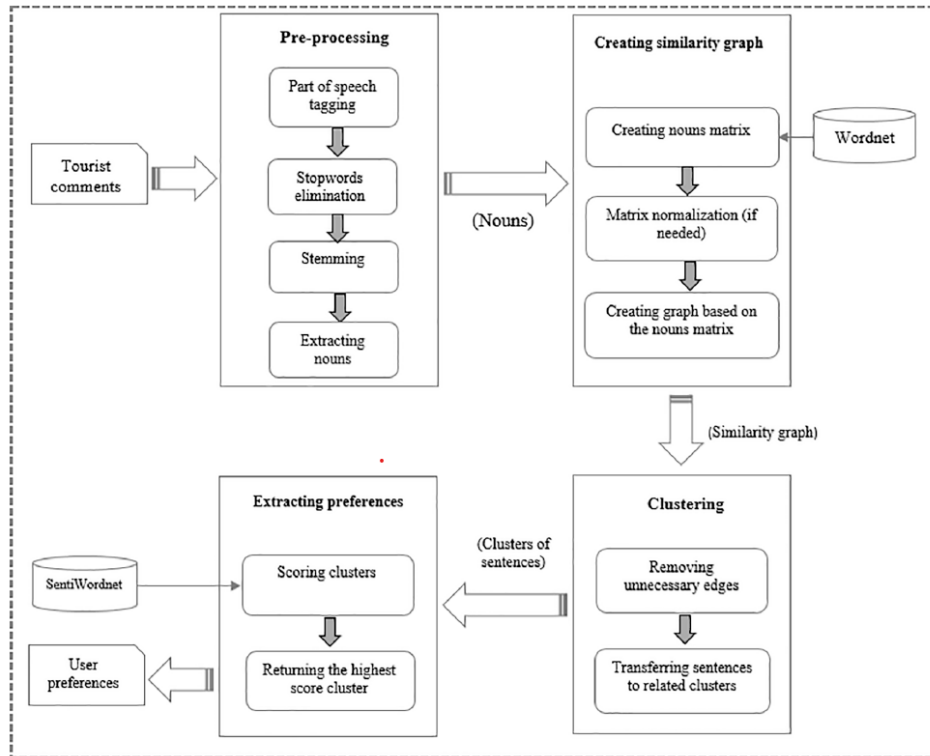


Figure 2.4: Example Utilization of Clustering for User Preferences

2.6.2 Hierarchical Clustering

Hierarchical Clustering is a method of cluster analysis that seeks to build a hierarchy of clusters. This approach is distinct in its use of an agglomerative algorithm, which starts with each data point as a single cluster and then successively merges clusters until one single cluster remains or a certain criterion is met. Ward's method is particularly efficient in minimizing the variance within each cluster[35]. Thus, it is advantageous for identifying the precise number of clusters, which would usually vary from one domain to another.

2.6.3 Density Peaks Clustering

The **Density Peaks Clustering** (DPC) summarization framework is specifically designed to jointly account for *relevance* and *redundancy* in extractive summarization. Unlike traditional methods that handle redundancy as a post-processing step, DPC integrates both aspects within a unified, one-pass selection process. The core idea introduced in [36] is that a concise summary should consist of sentences that act as *cluster centers*, i.e., sentences with both high *density* and high *divergence* relative to others. This stems from the observation that sentences in a document can be grouped into latent subtopics, and identifying these central sentences allows for broad yet focused coverage of the document content.

2.7 Semantic Knowledge Representations

There are various ways to represent knowledge or data. Those representations include complete sentences, context windows, tables, embeddings, triplets, graphs, and more. This section will explore the most common used embedding models, including the most recent ones and some worth mentioning earlier models.

2.7.1 Embeddings and Vector Representations

NLP has witnessed a significant transformation with the advent of embeddings and vector representations to keep up with its non-ending use-cases [37]. These techniques have become fundamental in enabling computers to process and interpret human language. Embedding models convert words, sentences, and entire documents into dense, low-dimensional vector spaces without altering the syntactic and semantic information. The focus has shifted from static representations, such as Word2Vec and GloVe, to contextual embeddings that adapt based on linguistic context. The most advanced models, often leveraging transformer architectures, have demonstrated remarkable capabilities across diverse languages, including English, German, and multilingual settings.

Contextual embeddings represent the forefront of NLP innovation. Unlike static embeddings, which assign a single vector to a word regardless of usage, contextual embeddings dynamically adjust based on the surrounding text. Among the pioneers of this paradigm was Embeddings from Language Models (ELMo), introduced by Peters et al. ELMo generates embeddings through bidirectional LSTMs, considering both the left and right context of a word[37]. However, transformer models have outperformed the architecture of ELMo, which provides more efficient parallel computation and a deeper understanding of context.[38].

BERT, introduced by Devlin et al., marked a paradigm shift in NLP by employing a fully bidirectional transformer architecture. Unlike earlier models that process text in a single direction, BERT utilizes masked language modeling (MLM) to predict randomly masked words within a sentence. This design enables BERT to capture bidirectional dependencies and perform state-of-the-art tasks like question answering, named entity recognition, and sentiment analysis. Building on BERT, RoBERTa enhanced the pretraining process by eliminating the Next Sentence Prediction (NSP) objective and training on significantly larger datasets. RoBERTa's optimization improved performance across multiple benchmark tasks without introducing additional architectural complexity[39].

The growing need for multilingual NLP has led to the development of embedding models capable of operating across diverse languages. Embedding models have revolutionized monolingual tasks in English and German. For English, BERT and its variants dominate benchmarks in tasks ranging from text classification to machine translation[39]. Similarly, German NLP has benefited from multilingual models like mBERT and XLM-R (Cross-lingual Language Model-Robust), alongside dedicated German models such as GottBERT, which is pre-trained exclusively on German corpora. These advancements have improved tasks like sentiment analysis, document classification, and syntactic parsing[37]. mBERT, although lacking explicit cross-lingual training objectives, demonstrated impressive zero-shot transfer

capabilities by leveraging shared subword tokenization and multilingual corpora. It laid the groundwork for subsequent models like RemBERT (Retrieval-Enhanced BERT), which further optimizes multilingual embeddings by integrating retrieval-based techniques to enhance semantic alignment[37].

XLM and its successor XLM-R represent significant milestones. XLM combines masked language modeling with translation language modeling, using parallel corpora to align representations across languages[38]. XLM-R extends this approach by training on CommonCrawl data for over 100 languages, providing robust performance on multilingual tasks such as cross-lingual natural language inference and unsupervised machine translation[39]. Another noteworthy evolution is T5 (Text-to-Text Transfer Transformer), which reframes all NLP tasks into a text-to-text format. This approach unifies text generation, summarization, translation, and classification under a single framework, making it versatile for both English and multilingual tasks. By employing a text-infilling objective and pretraining on the Colossal Clean Crawled Corpus (C4), T5 achieved state-of-the-art results in various domains[37].

2.8 AI-Generated Text Evaluation Methods

Evaluating generated text is critical in developing and assessing Natural Language Generation (NLG) systems. Effective evaluation methods ensure these systems produce text that meets the desired quality, coherence, and usability standards. Over time, evaluation methodologies have evolved to balance human judgment and automated metrics, each addressing unique aspects of text quality.

Text evaluation methods can be widely categorized into human-centric evaluations, automatic metrics, and machine-learned metrics, as detailed in [40]. Each method serves distinct purposes:

Human-Centric Evaluations: These involve direct human judgments and are often considered the gold standard for assessing fluency, coherence, and human likeness. However, they are resource-intensive and can be inconsistent across evaluators.

Automatic Metrics: Metrics like BLEU, ROUGE, and METEOR rely on n-gram overlaps to measure text similarity. These are computationally efficient but often fail to capture semantic nuances.

Machine-Learned Metrics: These newer approaches leverage neural networks to evaluate generated texts based on embeddings and semantic similarity, addressing some limitations of traditional metrics.

Evaluating text generated by NLG systems is a nuanced task requiring multiple methodologies to assess different quality aspects. These methods balance human judgment and automated systems, each addressing specific facets of evaluation. The paper [41] introduces a unified toolkit for evaluating sentence representations across tasks like natural language inference and semantic similarity, emphasizing the importance of standardized and centralized evaluation pipelines to improve reproducibility and fairness. According to [32], effective evaluation hinges on capturing fluency, coherence, and task-specific relevance. It suggests using diverse metrics tailored to the complexity of modern models like LLMs, as the

evaluation process requires a combination of quantitative and qualitative approaches.

2.8.1 Human-Centric Evaluation

Human evaluation remains critical in capturing nuanced aspects like fluency, coherence, and informativeness. However, as described in [42], the lack of standardized practices often leads to inconsistencies. This study emphasizes using clearly defined criteria, such as fluency and meaning preservation and recommends employing multiple annotators and reporting inter-annotator agreement for more reliable results. However, [43] critiques these methods as resource-intensive and inconsistent. Instead, it advocates for combining automatic and human evaluations to triangulate results. It has also been noted in [44] that human evaluations typically score text on grammaticality, coherence, and informativeness. Despite their reliability, the inconsistency and cost of human evaluations remain significant drawbacks.

2.8.2 Automatic Metrics

Metrics like BLEU and ROUGE are extensively used for tasks such as summarization and translation. However, their limitations are increasingly evident in open-ended tasks. For instance, BLEU fails to capture semantic similarity, often producing high scores for text with minimal informational content. While more recall-focused, ROUGE struggles with long text generation where narrative and factual consistency are critical [40]. [45] also adds METEOR to the precedent methods and highlights the limitations of such metrics, especially for tasks requiring semantic understanding. It introduces word deletion-based evaluation as an alternative, emphasizing its utility in local fidelity assessments. The limitations of such metrics are also discussed in [41], which suggests evaluating representations on downstream tasks to ensure broader applicability and reliability.

2.8.3 Machine-Learned Metrics

Recent advancements integrate machine-learned metrics to address the semantic gaps left by traditional methods. [43] introduces mutual information-based objectives, leveraging contextual embeddings to evaluate saliency and faithfulness without requiring human references. This approach showcases promise for unsupervised settings. Recent advances have introduced machine-learned metrics that employ contextual embeddings from models like BERT to assess text quality. These models are adept at capturing deeper semantic relationships. For instance, Sentence Mover's Distance (SMD) evaluates text coherence by comparing sentence embeddings and has demonstrated an improved correlation with human judgments in summarization tasks [40].

Combining human judgment with automated metrics is emerging as a best practice for robust evaluations. According to [42], hybrid methods that integrate task-specific metrics with qualitative human assessments offer comprehensive insights into generated text quality. Effective evaluation involves a mix of intrinsic and extrinsic methods. Standardized toolkits like SentEval provide robust pipelines for evaluating sentence representations, while best

practices for human evaluation ensure consistency and reproducibility. Together, these approaches support the development of accurate and contextually appropriate NLG systems. Integrating human feedback with advanced automated metrics can provide comprehensive insights. [40] also supports blending intrinsic and extrinsic evaluations to better understand surface-level and task-specific text quality.

3 Related Work

This chapter gathers a comprehensive overview of existing research to identify previously proposed solutions, highlight their limitations, and outline promising directions for further investigation aligned with the objectives of this thesis. In particular, it explores context windowing techniques for semantic extraction, examines clustering-based approaches to semantic grouping, investigates the potential of LLMs for text summarization, and addresses prevailing challenges in text classification. By collecting these contributions and identifying gaps in the previously carried-out related research topics, the chapter establishes the core upon which the subsequent chapters build, highlighting the focal points of this thesis.

3.1 Context Windowing for Semantic Extraction

Context windowing is a foundational technique in semantic extraction tasks. It involves segmenting text into manageable spans termed *context windows* to preserve semantic coherence while enabling downstream tasks such as clustering, frame induction, and retrieval augmentation. By maintaining the integrity of localized semantics, context windows facilitate efficient and meaningful processing of linguistic data.

3.1.1 Importance and Utility of Context Windows

Context windows are essential in processing long text sequences, especially for models that need to retain semantic coherence across input spans [46]. For instance, the *Parallel Context Windows* (PCW) approach segments text into discrete chunks and restricts attention mechanisms within each window using positional embeddings as illustrated in Figure 3.1 [47]. This technique not only enhances computational efficiency but also improves performance in multi-hop question answering and retrieval-augmented generation.

In semantic frame induction, contextualized embeddings such as BERT’s are crucial for distinguishing verb senses and their associated predicate-argument structures. These embeddings rely on context windows to cluster instances into coherent semantic frames, thereby enabling accurate disambiguation of polysemous verbs [48].

In the domain of information retrieval, context windows have been shown to enrich queries via semantic augmentation [46]. Window sizes directly affect clustering quality and alignment with user intent. Shorter windows (e.g., 10 minutes) better capture immediate context, improving search precision, especially in cold-start scenarios [47]. Models which utilize advanced semantic embedding and consistent context semantics (SECS) leverage multi-view consistency from context windows for more coherent clustering of high-dimensional sparse text data [49].

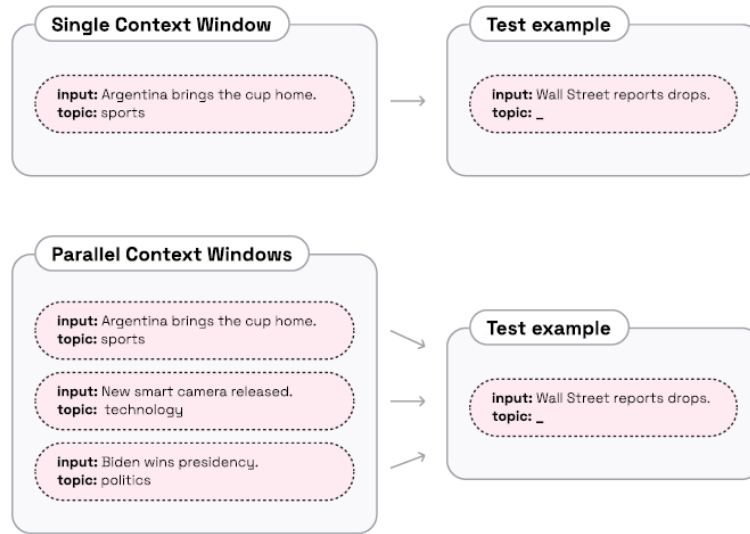


Figure 3.1: Illustration of LLM Training on PCW vs. Single Context Window

3.1.2 Context Window Extraction Approaches

This thesis builds on the methods proposed by Seibicke [50] for extracting and evaluating context windows around class-specific keywords. The motivation stems from the inherent ambiguity in natural language, where words like *bank* can denote different meanings depending on context.

Manual Approaches

- **Basic:** Fixed number of tokens before and after the keyword.
- **Naive:** Includes syntactic dependencies using tools like spaCy.
- **Dependency-Based:** Dynamically expands the window via syntactic relations.

Machine Learning Approaches

Supervised models (e.g., Random Forest, XGBoost, Gaussian classifiers) label tokens as inside/outside the window based on features like POS tags, dependency labels, and distance to keyword. Post-processing ensures inclusion of semantically meaningful tokens and the keyword itself [50].

Evaluation and Integration

Precision, recall, and F1 scores quantify model performance. Expert evaluations favored Gaussian models for producing longer, semantically rich spans. Extracted windows also support clustering (e.g., K-means) to classify contexts into semantic classes. The approach was validated on domain-specific corpora such as the German Business Registry [50].

3.1.3 Temporal Dynamics in Context Windowing

Vuong et al. [46] examined how different temporal windows (10 minutes, 1 hour, 1 day) affect semantic extraction. Short windows outperformed longer ones in cold-start scenarios by leveraging recent non-search digital behavior. Conversely, long-term contexts provided benefits for informational queries.

They employed Dirichlet–Hawkes Processes (DHP) for topic modeling, demonstrating that recency has a significant semantic weight. These findings emphasize the need for intent-aligned window sizing in semantic distillation pipelines [46].

3.1.4 Applications in Information Extraction

In the Semantic Web context, context windows facilitate entity extraction and linking by aligning mention surroundings with candidate entity representations [51].

- **Keyword-Based Context:** Local token analysis for disambiguation.
- **Variable Window Sizes:** Adjusted to balance performance and accuracy depending on the document type.
- **Collective Disambiguation:** Joint consideration of multiple entity mentions.

Contextual features exploited include strings, co-occurring terms, entity graphs, ontological categories, and syntactic structures. Context windows also support joint inference models (e.g., JERL, Babelify) and aid in detecting previously unseen entities [51].

Context windowing is widely considered as a critical enabler for semantic extraction in both traditional NLP pipelines and modern LLM-based architectures. From verb sense disambiguation to long-context retrieval, adaptive context window strategies ensure the preservation of semantic coherence and scalability. As methods for extending context horizons become more robust and data-efficient, the integration of context-aware modeling promises even greater precision and generalization in knowledge distillation systems.

3.2 Semantic Grouping via Clustering

Semantic grouping through clustering integrates clustering techniques with context windows to enable effective semantic knowledge extraction. Context windows in NLP highlight

semantic and sequential relational dependencies, producing refined representations for downstream tasks. Clustering captures patterns and organizes semantically similar contexts to enhance knowledge generalization.

3.2.1 Clustering with Contextual Embeddings

Clustering is widely used in applications that involve organizing semantically similar elements into coherent structures. It supports improved structure, retrieval, and representation of textual knowledge. Recent advancements use contextual embeddings and multi-view representations to tackle traditional limitations such as sparsity and high dimensionality. By incorporating multiple perspectives such as syntactic, semantic, and metadata multi-view clustering aligns diverse views to generate consistent and interpretable clusters [49].

Transformer-based embeddings (e.g., BERT) have revolutionized clustering by capturing fine-grained semantic relationships. These contextualized embeddings allow clustering models to handle nuanced meanings. For example, clustering verbs and arguments into semantic frames enhances representation and disambiguation [48]. In retrieval systems, clustering helps segment user queries and behaviors into semantically meaningful units, aligning outcomes more closely with user intent [47].

3.2.2 Traditional and Semantic Clustering Approaches

Semantic clustering differs from traditional clustering by emphasizing conceptual rather than lexical similarity. While Bag-of-Words (BoW) and vector space models often fail due to synonymy and polysemy, semantic clustering incorporates ontologies and contextual analysis to group documents more meaningfully [52].

Techniques in Semantic Clustering [52]:

- **Ontology-based clustering:** Uses domain knowledge bases (e.g., WordNet) to identify semantic relations.
- **Latent Semantic Analysis (LSA):** Projects documents into a latent concept space.
- **Word Sense Disambiguation (WSD):** Assigns contextually correct senses to words.
- **Concept weighting:** Enhances TF-IDF with semantic context.

Benefits:

- Improves accuracy through conceptual similarity.
- Resolves lexical ambiguity.
- Supports synonym-aware and cross-lingual clustering.
- Enhances evaluation metrics such as precision, recall, and F-measure.

Architectures and Algorithms:

- Preprocessing (tokenization, stop-word removal, stemming)
- Semantic enrichment via ontology mapping
- Algorithms:
 - *Bisecting K-Means*: Efficient for large-scale datasets.
 - *Hierarchical Agglomerative Clustering (HAC)*: Builds a cluster hierarchy.
 - *Self-Organizing Maps (SOM)*: Visualizes high-dimensional data.

Similarity Measures: Cosine similarity is commonly used post-transformation into concept space.

Challenges:

- Scarcity of high-quality ontologies.
- WSD dependency.
- Difficulties in benchmarking and evaluation.

3.3 Leveraging LLMs for Text Summarization

The advent of LLMs, such as GPT-3, GPT-4, and Llama, has significantly reshaped the field of Automatic Text Summarization (ATS), transitioning it from rigid, paradigm-specific techniques to more flexible, generative approaches. Unlike earlier models limited to either extractive or abstractive summarization, LLMs enable paradigm-agnostic summarization, seamlessly blending both approaches within a single framework in the form of contextual knowledge distillation [53].

3.3.1 Key Advantages of LLMs in ATS

LLMs offer several compelling advantages over traditional models in the summarization domain as highlighted in [53]:

- **In-context and Few-shot Learning:** They exhibit strong performance in zero-shot and few-shot settings, reducing the dependence on task-specific training data.
- **Generative Flexibility:** LLMs can easily switch between extractive, abstractive, and contextual knowledge distillation (hybrid summarization) via prompt modifications, without altering the underlying architecture.
- **Superior Output Quality:** Their training on massive and diverse corpora results in summaries that are fluent, coherent, and semantically rich.

3.3.2 Categories of LLM-Based Summarization Techniques

LLM-based summarization have been categorized according to [53] into:

- **Prompt Engineering:** Custom prompts guide LLMs in summary generation. Techniques include template-based prompts, Chain-of-Thought (CoT) prompting, agent-based pipelines, and Retrieval-Augmented Generation (RAG).
- **Fine-tuning:** Domain-specific fine-tuning using either internal parameter adjustments or adapter modules enhances summarization capabilities.
- **Knowledge Distillation:** LLMs act as teachers to train smaller models, enabling efficient summarization in low-resource settings.

3.3.3 Taxonomy and Trends in LLM-based Summarization Research

LLM-based summarization research has evolved into three major directions [54]:

- **Benchmarking Studies:** Evaluation across datasets like CNN/DailyMail and XSum has revealed strengths in coherence and fluency, but challenges remain with hallucinations, lead bias, and factual accuracy.
- **Modeling Studies:** Innovations include structured prompt design (e.g., PromptSum, SumCoT), multi-agent frameworks (e.g., SummIt, ImpressionGPT), distillation pipelines (e.g., InheritSumm), and chain-of-thought summarization.
- **Evaluation Studies:** Traditional metrics (ROUGE, BLEU) are increasingly supplemented with LLM-based evaluators (e.g., GPTScore, G-Eval), offering improved correlation with human judgment.

3.3.4 Uncertainty-Aware Summarization with LLMs

Preserving semantic uncertainty in summaries is an emerging focus. Recent work explores LLMs' ability to identify and retain lexical and semantic uncertainty expressions during summarization [55]. Using GPT-4 with XML-guided annotation, the study shows improved alignment and iterative feedback learning, achieving high precision and recall in preserving uncertainty in generated summaries.

3.3.5 Aspect-Based Summarization via Fine-tuned LLMs

Fine-tuning open-source LLMs such as Llama2, Mistral, Gemma, and Aya using the OASUM dataset has shown significant improvements in aspect-specific summarization [56]. Techniques like QLoRA and PEFT enhance performance across standard metrics and GPT-4-based evaluations, with Llama2-13b-FT leading in quality and generalization across domains.

3.3.6 Multi-LLM Summarization Frameworks

To mitigate challenges in long-document summarization, multi-LLM frameworks have been proposed [57]. These include centralized models for best-summary selection and decentralized architectures that use consensus mechanisms for generation and evaluation. This approach demonstrates significant gains (up to 3×) over single-LLM baselines in terms of ROUGE, BLEU, and factuality.

3.3.7 Topic-Driven Summarization (TDS)

LLMs also show promise in Topic-Driven Summarization (TDS), where structured outputs are guided by document objectives or user-defined topics [58]. Using GPT-4o’s extended context window (up to 128k tokens), models can generate table-of-contents, guiding questions, and context-aware summaries. This method surpasses traditional prompting strategies and facilitates both summarization and classification tasks, particularly for unstructured documents.

3.3.8 Evaluation and Limitations

Despite their strengths, LLM-based summarization methods face several challenges [53]:

- **Factual Hallucinations:** Tendency to generate plausible yet incorrect information.
- **Prompt Sensitivity:** Outputs vary significantly with minor prompt changes.
- **Computational Overhead:** High resource requirements for inference and fine-tuning.

Nevertheless, LLMs continue to redefine the state-of-the-art in ATS due to their generalization ability and human-like summary generation.

3.4 Text Classification Applications & Hurdles

Text classification is a foundational task in NLP with a wide array of real-world applications, including sentiment analysis, spam detection, product reviews, news categorization, email sorting, question answering, customer support automation, and consumer complaint handling. As the demand for intelligent document organization and real-time information retrieval increases, both multi-class and multi-label classification techniques have gained prominence [59].

3.4.1 Applications of Text Classification

Multi-label classification is particularly crucial when a document may belong to multiple categories simultaneously, which is an increasingly common requirement in domains like finance and consumer protection. For example, a single consumer complaint may include elements of “Credit Cards” and “Debt Collection,” or an email may reference invoice numbers,

payment confirmations, and dispute-related content, necessitating labels like “Request for Invoice” and “Dispute” [59].

Traditional supervised text classification methods rely heavily on large, annotated datasets. Deep learning models such as CNNs and RNNs have achieved strong performance in this space but require tens of thousands of labeled documents per task [60]. In response to the high labeling cost, researchers have explored semi-supervised, weakly-supervised, and zero-shot learning methods as scalable alternatives.

Semi-supervised approaches reduce annotation needs by incorporating unlabeled data through augmentation and graph-based learning techniques. Zero-shot classification attempts to infer labels for previously unseen categories using semantic representations but still depends on supervised training for seen classes. Weakly-supervised methods like LOTClass go a step further by using only label names without document-level annotations to achieve competitive results, making them attractive for low-resource domains [60].

3.4.2 Challenges and Future Directions in Text Classification

Despite advances in modeling techniques, several key challenges persist in building effective text classification systems which were made clear in both research papers [59] & [60]:

- **Contextual Complexity:** Traditional models such as Word2Vec and Bag-of-Words fail to capture deep contextual relationships, especially in multi-label classification tasks.
- **Entity Ambiguity:** Extracting entities such as dates, invoice numbers, and payment details for accurate labeling is difficult without robust Named Entity Recognition (NER) systems.
- **Text Variance:** Emails and complaint texts vary significantly in length and structure, making it hard to design models that generalize across diverse formats.
- **Performance Bottlenecks:** Transformer-based models like BERT achieve state-of-the-art results but are computationally expensive, posing limitations for real-time and industrial-scale deployments.
- **Imbalanced Datasets:** Some categories are underrepresented (e.g., “Virtual Currency” vs. “Credit Cards”), leading to biased models and poor performance on minority classes.
- **Overlapping and Hierarchical Labels:** Classes often have hierarchical relationships (e.g., “Prepaid Cards” under “Credit Cards”), complicating label definitions and evaluation metrics.
- **Label Name Ambiguity:** Single-word labels like “business” can be vague. While methods like LOTClass expand these into richer category vocabularies, semantic ambiguity remains a challenge.

- **Contextual Disambiguation:** Words may represent different categories depending on context. For instance, “sports” can refer to both activities and product types.
- **Sparse Supervision:** When using minimal supervision (e.g., label names only), models struggle to generalize without enhancements like masked category prediction and self-training.
- **Implicit Sentiment or Topic Representation:** Some documents encode meaning through subtle cues like sarcasm or comparison, requiring deep contextual models to uncover true intent.
- **General vs. Specific Labels:** Highly general labels lack precision, whereas overly specific ones demand domain expertise for effective construction.
- **Error Propagation:** In self-training pipelines, incorrect pseudo-labels can reinforce errors across iterations.

Due to the previously mentioned limitations, most of the current text classification-based researches follow the hybrid models direction that blend contextual understanding with structured entity extraction to handle the complexities of modern text classification [59]. Thus, this thesis topic handles and supports lots of those major limitations. First, it tackles **Contextual Complexity** and **Contextual Disambiguation** by leveraging LLMs equipped with windowed semantic extraction, enabling the capture of nuanced relationships beyond surface-level word co-occurrence. Through **Semantic Grouping via Clustering**, the work reduces **Label Name Ambiguity** and handles **Overlapping and Hierarchical Labels** by distilling contextually grounded archetypes that better reflect latent label structures. Moreover, the proposed framework improves robustness under **Sparse Supervision** by using label-driven prompt engineering combined with unsupervised clustering approaches, reducing dependence on large annotated datasets. Finally, by constructing domain-specific archetypes and focusing on semantically cohesive contexts, the approach narrows the gap between **General vs. Specific Labels** and enhances model interpretability across highly variable and imbalanced textual data distributions.

4 Methodology

This chapter outlines the methodology employed in this study, presenting a complete overview of the entire pipeline. It begins with a high-level description of the workflow, followed by detailed sections covering the datasets utilized, the text embedding models applied, and the clustering of domain-specific context windows. It then delves into the generation of domain-specific archetypes, concluding with the evaluation techniques used to assess the effectiveness and validity of the proposed approach.

4.1 Pipeline Overview

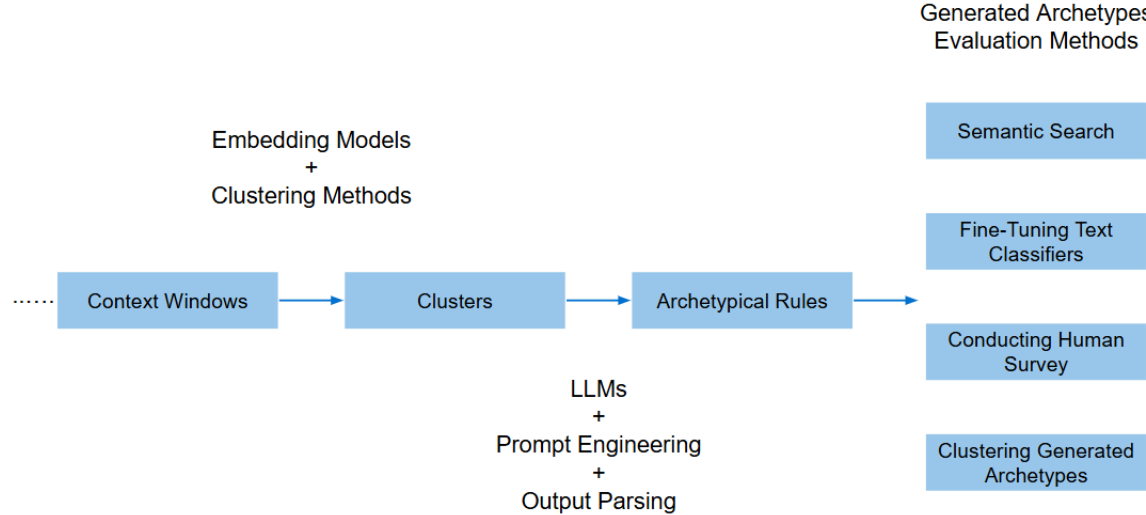


Figure 4.1: Thesis Work Pipeline Illustration

In this section, a pipeline overview of the thesis work is traversed to get a general idea of the flow as shown in Figure 4.1. Starting with the already implemented context window extraction pipeline. A brief summary of the previously implemented pipeline includes the following steps:

1. **Defining Domain Seed Keywords:** A foundational set of seed keywords was carefully curated through expert elicitation, targeted domain literature review, and established taxonomies related to the intended domain. These seed terms served as high-precision anchors, representative of the core semantic dimensions of each target class. Drawing

from the methodology proposed by [61], seed keywords were intentionally limited to unigrams to maintain precision in downstream keyword evaluation and to align with the lexical patterns typical to the domain corpus.

- 2. Domain Keywords Extraction:** Leveraging the curated seed keywords, an expanded set of class-specific domain keywords was generated using an iterative guided keyword extraction strategy. This process built upon a modified version of the KEYBERT framework, in which corpus embeddings were bypassed in favor of seed-centric similarity scoring. Each batch of the corpus was processed using cosine similarity between candidate and seed embeddings, scored via a hybrid averaging and max-scoring scheme, and incrementally enriched with high-ranking candidates mirroring the two-stage scoring and dynamic seed augmentation proposed in the referenced method. This allowed the pipeline to progressively refine and densify the keyword representation space for each semantic class over multiple iterations [61].
- 3. Filtering Extracted Domain Keywords:** To ensure the semantic integrity of the domain-specific keyword sets, a comprehensive hybrid filtering strategy by [62] was used. This included an initial clustering phase, primarily using recursive hierarchical clustering methods optimized by adaptive distance thresholds and followed by corrective refinement via geometric techniques such as convex hull and circle-based enclosures. These techniques isolate and exclude keywords lying outside clusters tightly associated with class seed keywords. Subsequently, advanced outlier detection algorithms like Isolation Forest and Local Outlier Factor were applied, leveraging contamination metrics aligned with clustering-derived outlier ratios. Finally, a semantic validation step was conducted using cosine similarity scores enhanced by lexical and conceptual expansions from WordNet and ConceptNet, ensuring that only contextually relevant and semantically coherent keywords were retained.
- 4. Extracting Domain Context Windows:** For each filtered domain keyword, multiple context windows were collected from the corpus. These context windows were defined using the techniques introduced by [50] and also referred to in the Subsection 3.1.2, capturing the surrounding semantic and syntactic environment. Examples from resulting context windows related to sports domain include: [*the big ten coaches were*, *five players*, *is vc bet 39 s heaviest liability ahead of this week 39 s australian pga championship*, *eye for sports a week later the tape of the fight between the indiana pacers and the detroit fans sticks*, *controversy over the bowl championship series*, *by bullying tactics from their opponents*, *look now but there s a bit of drama building in the quot other quot league championship series*, *take on mcgrady yao debut since rotations are usually shorter during preseason games players have*, *fans players brawl indiana s ron artest*, *39 union deadlocked the national hockey league and its players ended*, *trafficking and its long running guerrilla war than baseball players*, *england players sepp blatter the president of football 39 s world governing body fifa will tell*, *reveal their hands this week as olympic chiefs scrutinize their plans for holding the world 39 s greatest sporting extravaganza*, *do or die for braves astros cbc sports online the situation is*, *cup over finland cbc sports online canada*]

Utilizing the previously illustrated pipeline, context windows are extracted for each domain based on the datasets described in the following section. These context windows are then clustered using various clustering algorithms alongside embedding models to produce semantically coherent clusters. Each cluster is expected to represent a sub-domain or sub-topic within the main domain. This clustering step is critical, as it facilitates the next phase by structuring the input LLMs to generate domain archetypes more effectively.

In the next phase, multiple LLMs are experimented. Tailored prompts are designed and fed into the LLMs with the clustered context windows from the previous step to generate domain archetypes. The output is then cleaned and re-parsed to ensure it conforms to the desired format and structure.

The final phase is evaluation, which consists of four components:

1. **Semantic Search:** Comparing the similarity between two datasets' full-texts versus the similarity between one dataset's full-texts and the other's generated archetypes.
2. **Classifier Fine-tuning:** Fine-tuning text classifiers on the original full-text data and comparing their performance when fine-tuned on the generated archetypes.
3. **Human Evaluation:** Conducting a survey in which participants compare archetypes generated by different LLMs for the same domain and cluster, to determine human preference.
4. **Cluster Uniqueness Assessment:** Comparing the clustering of archetypes from mixed domains to clustering of full-text documents from mixed domains, to assess whether the archetype-based clusters exhibit greater domain separation and uniqueness.

4.2 Datasets

To evaluate the effectiveness of the proposed approach for semantic extraction and class archetype distillation, five diverse textual datasets are involved in this thesis work. These datasets span multiple domains such as sports journalism, general news classification, and topic-based news grouping. This selection ensures a wide-scope evaluation across various content genres, linguistic styles, and label granularities.

4.2.1 OnlySports Dataset

The OnlySports dataset is a curated sports-focused news dataset that contains over 864M samples for only sports category. It includes articles with rich semantic variation, as it comprise a diverse range of content including not only news articles, but also blogs, match reports, interviews, and tutorials, making it ideal for testing contextual windowing in narrowly-defined domains [63].

4.2.2 AG News

AG News is a well-known benchmark for text classification. It comprises 120,000 training and 7,600 test samples categorized into four high-level news topics: World, Sports, Business, and Sci/Tech. Its simplicity and balance make it an excellent dataset for baseline performance evaluation [64].

4.2.3 BBC News

This dataset features 2,225 news articles from the BBC, distributed across five topics: business, entertainment, politics, sport, and tech. Despite its smaller size, it is useful for early-stage validation and qualitative analysis [65].

4.2.4 20 Newsgroups

A classic dataset used in topic modeling and clustering, the 20 Newsgroups dataset contains nearly 19,000 forum posts across 20 discussion categories. Its high class count and informal language pose a suitable challenge for fine-grained semantic grouping [66].

4.2.5 HuffPost News Category Dataset (Short Descriptions)

The HuffPost dataset includes over 200,000 entries labeled across 42 fine-grained news categories. While each entry links to a full article, the actual article content is not explicitly provided. Instead, the short descriptions accompanying each entry were utilized as the primary textual input. These concise short descriptions add a challenge for semantic analysis and enable robust evaluation of large-scale clustering and archetype discovery tasks [67].

Table 4.1 summarizes the key characteristics of each dataset, including their domain, number of classes, total samples, and respective sources.

| Dataset Name | Domain | Classes | Samples | Source |
|---------------|-----------------|---------|---------------|-------------|
| OnlySports | Sports | 1 | 1,727,979,830 | HuggingFace |
| AG News | News Articles | 4 | 127,600 | HuggingFace |
| BBC News | News Categories | 5 | 2,225 | Kaggle |
| 20 Newsgroups | Forum Posts | 20 | 18,846 | Kaggle |
| HuffPost News | Online News | 42 | 209,527 | Kaggle |

Table 4.1: Overview of Employed Datasets

4.3 Text Embedding Models

To enable effective clustering of context windows and news article representations, dense vector embeddings were generated using two state-of-the-art transformer-based models:

`jinaai/jina-embeddings-v3` and `sentence-transformers/all-mpnet-base-v2`. Both models transform text into semantically meaningful embeddings, making them well-suited for unsupervised clustering tasks.

4.3.1 Jina AI Embeddings v3

The `jinaai/jina-embeddings-v3` model [68] is a multilingual, instruction-tuned transformer-based encoder optimized for semantic similarity tasks across diverse domains. It produces 768-dimensional embeddings and supports input texts in over 100 languages. Developed by Jina AI, this model emphasizes dense clustering performance and is tuned for real-world retrieval and ranking tasks.

4.3.2 MPNet Base v2

The `sentence-transformers/all-mpnet-base-v2` model, based on Microsoft’s MPNet architecture, is a widely used benchmark for sentence-level embeddings. It offers strong general-purpose performance across a variety of NLP tasks such as semantic search, clustering, and sentence similarity. This model also outputs 768-dimensional vectors and is trained on large-scale datasets with contrastive learning objectives.

4.4 Domain Context Windows Clustering

In this section, the focus is on clustering the context windows extracted from the domain-specific data. The goal is to group windows that exhibit similar patterns, thereby uncovering cohesive substructures that can aid in subsequent phase. Two main approaches are examined:

1. **Recursive Hierarchical Clustering** on both the full set of windows and on a reduced subset of selected windows.
2. **Density Peaks Clustering** to discover representative clusters based on local density and divergence.

The overall workflow involves generating the context windows, preparing their feature embeddings as discussed in last section, and then feeding these vectors into the respective clustering algorithms.

4.4.1 Recursive Hierarchical Clustering: Full vs. Selected Windows

Recursive Hierarchical Clustering (RHC) leverages the iterative, top-down process of classical agglomerative clustering to refine the cluster structure. Initially, all context windows (the “full” set) are clustered, then recursively focus on each major cluster to explore deeper partitions if needed. The main steps are:

1. **Feature Extraction:** Represent each context window with a feature-embedding.

2. **Initial Clustering:** Apply hierarchical agglomerative clustering to the entire set of windows using ward linkage method.
3. **Recursive Partitioning:** For each identified subcluster, check for internal heterogeneity. If the cluster is sufficiently large or diverse via density measure threshold, apply hierarchical clustering again, thus “splitting” it further.
4. **Comparison of Full vs. Selected Windows:**
 - **Full Windows:** Use all available windows to capture a broad representation of the domain. This can reveal coarse-grained groupings and may expose interesting high-level patterns but could introduce noise.
 - **Selected Windows (Trigram Filtering):** Instead of including all windows, a domain-driven tri-gram based approach is borrowed to filter out overlapping or redundant context windows. This strategy could result in more cohesive groupings and reduces overall complexity, although, it risks discarding some nuanced variations.

A cluster example using full-windows iterative hierarchical clustering and jinaai/jina-embeddings-v3 in domain sport about evaluations and reactions to a sports game sub-domain is: *Cluster 83: [‘m satisfied the game was handled’, ‘been very interesting to watch the game’, ‘the game arguably the highlight of the draw is’, ‘are pleased but move on to the next game’, ‘had a chance to win the game’, ‘felt we did enough to win the game’, ‘was a very intense occasion and a very destructive game’, ‘thought it was a horrible game in the first half and it was not much better in the second’, ‘everything we could have done to win the game’, ‘was a hell of a tough game’, ‘are all disappointed we lost the french game’, ‘controlled the game in the first half but we knew that they would come out and try everything after half’, ‘think we played quite well and it was a very good game’, ‘had three chances to win the game’, ‘knew all along that we would be a huge threat particularly the first game’, ‘chance we had before this game’, ‘had a superb game’, ‘was amazing to watch but never did i think the french could lose that game’, ‘was a great game’, ‘had an awesome game’, ‘are playing a great team game’, ‘fantastic with the players and the coaches’, ‘played a very good game’, ‘are still very disappointed with our last game’, ‘missed another chance to seal the game’, ‘enjoyed the game’]*

4.4.2 Density Peaks Clustering

Density Peaks Clustering (DPC) is a distinct approach that identifies cluster centers by looking for points in the feature space characterized by high local density and large distance from points with higher density. Once these cluster centers are identified, other points are assigned to the nearest cluster center. The key steps in DPC procedure as introduced in [36] are:

1. **Distance Computation:** Compute pairwise Euclidean distances among the context windows.
2. **Local Density Estimation:** For each window, estimate its density by counting the number of points within a specified distance cutoff or by summing distance-based weights.

3. **Distance to Higher-Density Points:** For each window, measure how far it is from the nearest point with a higher density.
4. **Cluster Center Identification:** Points that combine (i) high density and (ii) large distance to any higher-density neighbor are flagged as *cluster centers*.
5. **Assignment of Remaining Points:** Non-center windows are assigned to the nearest cluster center. This typically yields a user-defined number of clusters based on density thresholds.

This method complements hierarchical clustering by giving the chance to uncover potential clusters without strictly imposing a global linkage criterion. In many cases, DPC can detect small but dense clusters of windows that might remain undetected by traditional hierarchical methods.

4.5 Generation of Domain-Specific Archetypes

The core step in the methodology involves the generation of domain-specific archetypes that summarize the thematic essence of each cluster of semantically similar context windows. These archetypes serve as a backbone for downstream tasks such as domain-specific classification, reasoning, and pattern interpretation. This step was realized through an iterative process involving prompt engineering, lightweight LLM experimentation, batch-wise cluster grouping, and robust output parsing and validation mechanisms.

4.5.1 Prompt Design

A range of prompt structures were experimented with to guide the LLMs toward producing high-quality, interpretable archetypes. The early approaches included minimal prompts with and without contextual cluster examples, but they often yielded generic or repetitive outputs. To overcome this, **Chain-of-Thought** prompting was adopted to encourage deeper reasoning over the thematic structure of the clusters. In scenarios where certain clusters consistently failed to generate meaningful archetypes, **cloze-style prompting** was tested, framing the task as if prompting the LLM, if you cannot generate an archetype for any of the clusters, just place an empty string, so that would challenge the LLM to generate an archetype.

Consequently, the final prompt was explicitly designed to instruct the model to output an empty string for both the `rule` and `examples` fields in cases where a coherent archetype could not be generated. This also ensured structural consistency in the output JSON while transparently signaling gaps in coverage.

After iterative refinement, the prompt evolved into a structured and explicit system message, which clearly delineated the role of the model, the format of the expected output, and post-generation constraints. Notably, the prompt enforced:

- Domain-specific language.

- One archetype per cluster.
- A strict output format in JSON, including fallback behavior for low-signal clusters.

The prompt also embedded a final consistency check directive that validated the one-to-one mapping between clusters and archetypes, helping reduce hallucinations and misalignment in the generated output.

The final prompt structure used is shown in Figure 4.2. This prompt structure guaranteed consistent, machine-parseable output and prevented extraneous commentary or reasoning from polluting the final archetypes.

4.5.2 Utilized LLMs & Quantization

Numerous language models were evaluated for their ability to generate high-fidelity archetypes across multiple domains. The focus was on **lightweight, instruction-tuned models** to ensure fast inference and ease of deployment. The models experimented with included:

- Qwen2.5-1.5B-Instruct [69]
- Gemma-2B
- Mistral-7B-v0.3
- Phi-3.5-Mini-Instruct
- OpenBMB-MiniCPM-4B [70]
- Phi-4-14B
- Meta-Llama-3-8B [71]
- DeepSeek-R1-Distill-Qwen-14B [72]
- DeepSeek-R1-Distill-Llama-8B [72]

Where applicable, quantized variants of the larger models were used to balance memory efficiency and generation quality. Specifically:

- Phi-4 and DeepSeek-R1-Distill-Qwen-14B were used in 4-bit quantized versions.
- DeepSeek-R1-Distill-Llama-8B and Llama-3 8B were used in 8-bit quantized form.

These smaller or quantized models were found to be sufficiently capable of handling structured prompt-following and domain archetypes generation, especially when paired with robust prompting strategies.

4.5. GENERATION OF DOMAIN-SPECIFIC ARCHETYPES

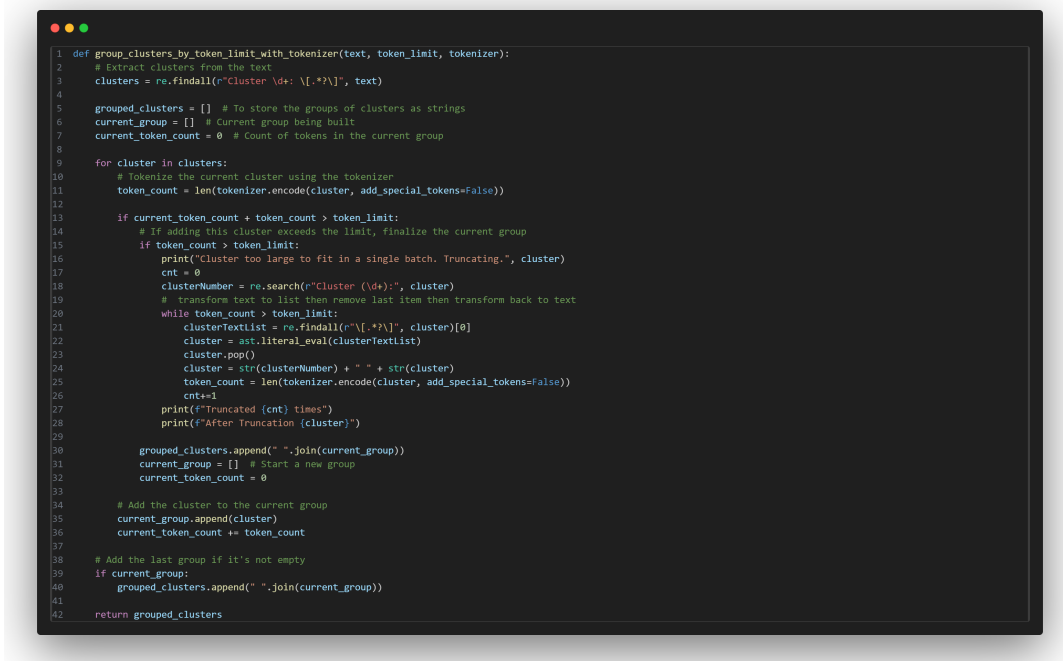
```
1 self.system_prompt = f"""
2 You are a {domain_name}
3 domain expert and a classification rule generator. Your task is to create a set of coherent, complete, and domain-specific archetypal rules—one rule per cluster—for multiple clusters of context windows. Each cluster groups similar context windows that share thematic or conceptual similarities, and the rules you generate will later be used to classify new text in the given domain.
4
5 Input:
6 - The user will provide the domain and a set of clusters.
7 - Each cluster is a collection of context windows that share a common theme or pattern.
8 - The number of clusters is explicitly provided; you must produce exactly one rule per cluster.
9
10 Instructions:
11 1. Read and understand the domain and the clusters:
12 - Use domain-specific terminology and concepts consistently.
13 - Treat each cluster as representing a unique theme or pattern within the domain.
14
15 2. For each cluster:
16 - Analyze the context windows to identify the core theme or pattern.
17 - Create one concise rule that captures the essence of that cluster without starting with phrases like "it focuses on...", "describes...", "highlights...", "concerns about...", etc.
18 - Ensure the rule is abstract yet precise enough to classify similar texts in the domain.
19 - Follow the rule with several illustrative sentences that embody the same theme, each separated by a period.
20
21 3. Output:
22 - Produce a JSON array where each element is a JSON object with exactly two keys:
23 - "cluster_number": a Number representing the cluster number.
24 - "rule": a String containing the generated rule.
25 - "examples": a String containing the illustrative sentences for the rule.
26 - If you are unable to generate a rule for a particular cluster, set the values of both "rule" and "examples" to "" as an empty string for that cluster.
27 - The order of the objects in the JSON array should exactly match the order of the clusters provided.
28
29 4. Final Check:
30 - Ensure that the number of objects in the JSON array exactly matches the number of clusters provided.
31 - Ensure that no internal reasoning, summaries, or commentary appears in the final output.
32 - Each JSON object must strictly follow this shape: {{ "cluster_number": Number, "rule": String, "examples": String }}.
33
34 Example (for illustration only, not actual content):
35 If the domain is "Food & Drink" and clusters provided are:
36 - Cluster 1: ['fast food company was accused', 'fast food company was accused in federal court of serving adulterated food', 'food trucks are', 'takeout and delivery food items have become', 'food and beverage industry often hidden in', 'restaurant food can be made', 'of fast food breakfast wars', 'realize how difficult it is to find a food', 'free food', 'know to start a food fight', 'eat a street cart food dog it s', 'about the safety of your takeout and delivery food', 'firm against the fast food brand', 'fast food restaurants keep', 'fewer in flight meals more food available in the airports and closer in flight quarters has', 'containers to store all your food', 'way institutions like schools prisons and senior centers think about food', 'know about the food court', 'ponders how we can fight food insecurity']
37 - Cluster 2: ['been your most loved foods', 'rounded up 10 of the most essential foods', 'comfort foods provide', 'believed that good food is good food', 'russian food is', 're always on the go but don't want to compromise good food', 'accompanied with great food', 'had the opportunity to eat some really good food', 'offer quality food', 'of my top picks for foods', 'believed that good food is', 'types of fish and other foods', 'love these foods', 'food and durable goods are', 'is my favorite food', 'foods of the world', 'about the enjoyment of quality genuine food', 'enjoy the mightiest of all foods', 'is a hybrid food']
38 - Cluster 3: ['political correspondent james hardy said', 'by ex itn political editor john sergeant', 'political editor andrew marr said', 'political correspondent andrew marr said', 'political correspondent mark mardell said', 'political editor andrew marr suggested', 'political correspondent glenn campbell said', 'political editor andrew marr mr brown said', 'political correspondent vicky young said', 'political correspondent carole walker said', 'according to political brief', 'political correspondent shaun ley says']
39
40 A correct output might look like:
41 [
42 {{
43 "cluster_number": 1,
44 "rule": "Food consumption, preparation, safety, and societal impacts.",
45 "examples": "Fast food companies accused of unsafe practices. Food insecurity in schools and prisons. Takeout and delivery food safety."
46 }},
47 {{
48 "cluster_number": 2,
49 "rule": "Enjoyment, quality, and preference of food.",
50 "examples": "Good food experiences, comfort foods, and top food picks. Diverse types of food including cultural and hybrid variants."
51 }},
52 {{
53 "cluster_number": 3,
54 "rule": "",
55 "examples": ""
56 }}
57 ]
58
59 Remember: Output only the final JSON array with one object per cluster this object must contain the cluster_number, rule and examples as keys. Do not include any additional text.
60 """
61
```

Figure 4.2: Final LLM Prompt for Clusters' Archetypes Generation

4.5.3 Batch-Wise Cluster Grouping for Token Efficiency

Given LLMs' token context window limitations, clusters were first grouped into batches that respected each model's maximum token limit. To accomplish this, a dynamic tokenizer-based grouping function was developed. It tokenized each cluster and accumulated them until just under the token threshold, ensuring that no batch exceeded the model's capacity.

In cases where an individual cluster was too large to fit even by itself, it was truncated incrementally by removing context windows until it met the token requirement as shown in Figure 4.3. This enabled robust inference across varying domain cluster sizes while minimizing the risk of context overflow.



```

1 def group_clusters_by_token_limit_with_tokenizer(text, token_limit, tokenizer):
2     # Extract clusters from the text
3     clusters = re.findall("Cluster \d+: [^\s]*", text)
4
5     grouped_clusters = [] # To store the groups of clusters as strings
6     current_group = [] # Current group being built
7     current_token_count = 0 # Count of tokens in the current group
8
9     for cluster in clusters:
10        # Tokenize the current cluster using the tokenizer
11        token_count = len(tokenizer.encode(cluster, add_special_tokens=False))
12
13        if current_token_count + token_count > token_limit:
14            # If adding this cluster exceeds the limit, finalize the current group
15            if token_count > token_limit:
16                print("cluster too large to fit in a single batch. Truncating.", cluster)
17                cnt = 0
18                clusterNumber = re.search("Cluster (\d+):", cluster)
19                # transform text to list then remove last item then transform back to text
20                while token_count > token_limit:
21                    clusterTextList = re.findall("[^\s]*", cluster)[0]
22                    cluster = ast.literal_eval(clusterTextList)
23                    cluster.pop()
24                    cluster = str(clusterNumber) + " " + str(cluster)
25                    token_count = len(tokenizer.encode(cluster, add_special_tokens=False))
26                    cnt+=1
27                print(f"Truncated {cnt} times")
28                print(f"After Truncation {cluster}")
29
30                grouped_clusters.append(" ".join(current_group))
31                current_group = [] # Start a new group
32                current_token_count = 0
33
34            # Add the cluster to the current group
35            current_group.append(cluster)
36            current_token_count += token_count
37
38            # Add the last group if it's not empty
39            if current_group:
40                grouped_clusters.append(" ".join(current_group))
41
42        return grouped_clusters

```

Figure 4.3: Grouping Clusters per each LLM Call According to Model's Tokenizer limit

4.5.4 Output Parsing and Validation

Given the occasional instability of LLM outputs especially with longer responses or edge cases, a **post-generation validation and extraction step** was implemented. Initially, the prompt itself attempted to enforce a JSON-formatted output matching the number of input clusters. However, occasional model hallucinations, formatting errors, or additional comments could break the structure.

To address this, a Python-based cleaning and extraction utility was developed. This parser:

1. Extracts all valid { ... } JSON-like objects from the raw response.
2. Cleans anomalies such as:

- Trailing commas
 - Extra quotes
 - Non-JSON comments
3. Validates structural integrity of each object.
 4. Filters objects with missing or empty `rule` and `examples` fields.
 5. Tracks and reports cluster indices with missing archetypes.

Furthermore, each archetype corresponding to a cluster would be a concatenation of the `rule` field, followed by `examples` field from the LLM generated JSON object. This ensured structural consistency across the final archetype set and made the pipeline resilient to malformed outputs, ensuring every cluster had a corresponding entry, even if that entry was marked empty. In cases where valid JSON extraction failed, diagnostic printouts were used for manual debugging and batch-specific refinement.

Illustrations for LLMs archetypes outputs generation for the same sports domain cluster given in Subsection 4.4.1 after concatenating `rule` and `examples` fields are shown in Table 4.2

| LLM | Generated Archetype |
|--------------------|--|
| OpenBMB-MiniCPM-4B | Intensity, opportunities, and emotional responses during and after games. Highly competitive games, chances created, intense occurrences, feelings of victory or defeat, overall game evaluations, reactions to close calls or missed opportunities. |
| Phi-4 | Post-match analyses reflecting satisfaction, intensity, and opportunities missed in sports engagements. Contentment with handling of event proceedings. Observational interest in high-profile matches. Highlight status attributed to pivotal fixtures. Forward-looking mindset post-victory/defeat. Missed scoring opportunities lamented. Intensity characterized by aggressive tactics. Critique of poor initial-half performance. Comprehensive effort acknowledgment despite loss. Tactical dominance discussed alongside expected counter-strategies. Recognition of strong collective performance. Multiple near-success instances recounted. Anticipatory threats identified pre-contest. Reflective consideration of preceding challenges. Commendation of exceptional individual displays. Surprise reactions to unexpected defeats. Overall positive assessment of engaging contests. Team cohesion praised amid adversity. Collaborative spirit between athletes and mentors emphasized. Disappointment voiced concerning recent losses. Regrettable unseized decisive moments. |
| Meta-Llama-3-8B | Player performances, emotions, and reactions post-game. Solid game, disappointing loss, feeling good throughout the game. |
| Mistral-7B-v0.3 | Post-match analysis and sentiments expressed by teams and individuals. Satisfaction with the game. Intense occasion. Destructive game. Disappointment over losing the French game. Controlling the game. Great game, fantastic game. |

Table 4.2: Generated Archetypes by LLMs for Game Evaluation Cluster Example

4.6 Evaluation Techniques

This chapter presents the chosen evaluation approaches to evaluate the effectiveness and validity of the proposed approach. It covers a range of techniques, including semantic search to assess similarity comparisons between the domain’s full-text, and generated archetypes, fine-tuning of text classifiers using both full-text and distilled archetypes to compare classification accuracy, and a human evaluation survey designed to gather qualitative feedback

and validate the semantic coherence, and multiple LLMs outputs variations of generated archetypes. Additionally, clustering experiments are conducted on both the archetypal rules representations and the original full-text data to analyze their structural consistency, semantic grouping capabilities, and domains disjointness.

4.6.1 Semantic Search

As an initial step to evaluate the semantic coherence and label alignment of the generated archetypes, two types of semantic search experiments are conducted. These experiments assess how well archetypes capture the underlying semantics of domain-specific texts and test the generated archetypes' scope coverage in comparison with the corresponding original full-text.

Semantic Search with Archetypes from Multiple LLMs

In the first evaluation, archetypes generated by several distinct LLMs were used, including **Mistral-7B**, **Phi-4**, **MiniCPM-3B**, and **Meta-Llama-3-8B**, to form an archetypal corpus for semantic search. Each archetype was associated with a label derived from its corresponding cluster's dataset domain. A pre-classified dataset was used as the query set, with each query labeled according to its ground truth category.

Both the query texts and the model-generated archetypes were embedded using the **jinaai/jina-embeddings-v3** model via the **SentenceTransformer** framework. For each query, the top 3 most semantically similar archetypes from each model's corpus were retrieved using cosine similarity. Then, the predicted label (derived from the top match and from the most frequently occurring label among the top 3) was compared to the query's actual label.

Performance was evaluated using two metrics:

- **Top-1 Accuracy:** The percentage of queries for which the top retrieved archetype's label matched the ground truth label.
- **Top-3 Majority Accuracy:** The percentage of queries where the most frequent label among the top 3 matches corresponded to the true label.

Direct Semantic Search Between Texts of Similar Domains

In the second evaluation, the inherent semantic similarity between full-text samples is examined across datasets belonging to the same or similar domains. Thus, instead of using archetypes, raw texts from one dataset (used as the corpus) are directly compared against another dataset (used as the query set), ensuring both originated from the same general domain.

Using the same embedding model, both the corpus and the queries were encoded and performed top-3 nearest neighbor search using semantic search. As with the previous evaluation, both top-1 and top-3 accuracy were computed by comparing the predicted labels to the ground truth labels of the query texts.

This setup serves as a strong baseline for testing the representational effectiveness of full-text semantic embeddings. It also provides a comparison point for evaluating the utility of archetypes in capturing semantic distinctions and highlights the overlap between datasets from similar domains. It is expected that the generated archetypes would cover the same scope as the original datasets' full-text.

4.6.2 Fine-Tuning Text Classifiers on Archetypes vs. Full Text

In order to evaluate the generalization of archetype-based representations in supervised classification tasks, a series of fine-tuning experiments were conducted using three transformer-based models: `roberta-base`, `deberta-v3-base`, and `bert-base-cased`. First, experiments were conducted with 3 epochs for each classifier trained on same data, to point out the best one. Then, the top model was fine-tuned on full-text dataset at a time and on archetypes that had been generated by various LLMs at another time. Experiments were carried out over 1, 3, and 5 training epochs to observe performance trends under varying training durations.

Dataset Preparation

For archetype-based training, archetypes were first extracted and grouped based on their source LLM. Each archetype was associated with a semantic label, and the resulting datasets were structured into tabular form. Labels were mapped to integer identifiers to be used as class indices.

For full-text fine-tuning, existing labeled datasets were employed. Label-to-integer mappings were also generated and saved for consistency during inference and evaluation.

Training Procedure

All models were fine-tuned using the HuggingFace Trainer API. The following steps were followed during training:

- **Tokenization:** Text inputs were tokenized using each model's respective tokenizer, with truncation and padding applied to a maximum sequence length of 512 tokens.
- **Splitting:** Each dataset was divided into training and validation subsets using an 80/20 split.
- **Training Configuration:** A learning rate of 2×10^{-5} , weight decay of 0.01, and batch size of 8 were applied. Models were evaluated and checkpoints saved at the end of each epoch.
- **Epoch Variants:** Each configuration was trained separately for 1, 3, and 5 epochs to assess training efficiency and generalization performance.

Evaluation Methodology

Upon completion of training, the models were evaluated on other test datasets from the same domain as the training data. The evaluation was performed in inference mode, with computations carried out on GPU. For each text input, the predicted class label was obtained.

The following metrics were used to assess performance:

- **Accuracy** measures the proportion of correctly predicted samples and is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP , TN , FP , and FN denote true positives, true negatives, false positives, and false negatives, respectively.

- **Precision** indicates the proportion of positive identifications that were actually correct:

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

- **Recall** represents the proportion of actual positives that were correctly identified:

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

- **F1-score** is the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

These metrics were computed and averaged for each test dataset classification results.

Comparison Criteria

It is anticipated that generated archetypes would generalize more than datasets' full-text in text-classification downstream tasks. The results obtained from archetype-based fine-tuning were compared against those derived from full-text fine-tuning using the same model architectures and training configurations. This comparison was used to assess the degree to which distilled archetypes preserved classification-relevant information. It also served to highlight the potential of LLM-generated archetypes as efficient and compact representations, particularly in scenarios with limited computational resources or when training time is constrained.

4.6.3 Human Survey & Feedback

To evaluate the alignment between AI-generated summaries and human interpretations, a structured survey titled “*A Survey on Text Generalization for Classification*” was designed and conducted. The goal of the survey was to analyze non-domain experts preferences for the various LLMs’ generated archetypes and check if those preferences match previous evaluations’ results. The survey consisted of grouped text excerpts from the business domain for classification purposes from clusters generated from both AG News and BBC News datasets.

Survey Design and Purpose

The survey was structured into four distinct clusters of text excerpts, labeled A, B, C, and D. Each cluster was composed of short snippets derived from business-related news articles. These clusters were curated to cover a diverse range of themes within the business domain, such as corporate governance, financial regulation, reporting delays, and economic developments as shown in Figure 4.4.

For each cluster, four AI-generated summaries were provided including **Mistral-7B**, **Phi-4**, **MiniCPM-3B**, and **Meta-Llama-3-8B**, each produced by a different model. Participants were instructed to evaluate each summary in the context of its corresponding text cluster. The evaluation was conducted to determine the summary’s effectiveness in terms of content coverage, interpretability, and relevance.

The following questions were addressed through the survey for each cluster (A, B, C, D) and for each resulting summary (1, 2, 3, 4):

- How relevant and meaningful is Summary A-1 according to the provided clustered business domain text in Cluster A?
- Does Summary A-1 cover the complete variety of the content provided by the clustered business domain text in Cluster A?
- Is Summary A-1 easy for non-experts in the business domain to comprehend?

4.6.4 Participant Recruitment via Prolific

The survey was distributed through the Prolific platform¹, a widely used participant recruitment service for academic and industrial research. To ensure high-quality responses, a pre-screened group labeled *Qualified AI Taskers* was selected.

This group consisted of individuals who had successfully passed Prolific’s internal AI Task Assessment, which evaluates reasoning, fact-checking, and writing skills. As shown in Figure 4.5, this cohort comprised 1,211 verified participants at the time of survey deployment.

By targeting this specific group, a higher degree of reliability, comprehension, and engagement with the task was ensured, which is particularly important for evaluating complex tasks involving summarization and content generalization.

¹<https://www.prolific.com>

4.6. EVALUATION TECHNIQUES

Group 1

Please read the following cluster of text excerpts (bulleted list), followed by their corresponding summaries. Then, for each summary, please answer the three provided questions.

Cluster A:

- "us american companies were granted a six month delay wednesday on the deadline for reporting the value of stock options"
- "delay filing restated financial results shaking investor confidence in the leading telecom equipment maker and sending its stock"
- "unveil details on getting free credit reports the nation 39 s three biggest credit reporting companies have announced"
- "way of corporate control"
- "federal appeals court tuesday to block temporary rules by the u s federal communications commission forcing the companies to"
- "national accounting firms are eagerly filling"
- "had two significant excuses for a protest vote as news corporation sought"
- "stock options must be treated as expenses in a final statement on a stock options rule the financial accounting"
- "said wednesday that it might strengthen corporate governance provisions"
- "findings from a six month investigation of the nation 39 s big four accounting firms"
- "received a first notice for 157 million in back taxes on wednesday sending its stock"
- "of audits of big corporations"
- "shrouds us e vote machines the three companies that certify the us 39 voting technologies operate"
- "due to problems at its electronic voting machine business"
- "concessions on corporate governance"
- "decided to move forward the release date of its fiscal second quarter earnings to monday dec 13 before markets open"
- "corporate governance of"
- "a temporary restraining order by the us district court for the western district of washington has been placed on two companies"
- "counted as expenses the nation 39 s accounting rule maker decided yesterday that companies will have"
- "says it has approved stricter corporate governance provisions"
- "at the securities and exchange commission and the public company accounting oversight board accounting firms"
- "rules for corporate reporting"
- "inspections of the big four accounting firms had found"
- "companies unveil details on getting free credit reports the nation 39 s three"
- "corporate governance score is quot pretty darn awful quot google has"
- "manipulated its earnings to artificially inflate its stock price"
- "fined and censured by nasd new york october 25 newratings com citigroup global markets inc has been censured"
- "loom an oil giant with a corporate governance structure"
- "provisions of australian corporate law"

Summary A-1: *Corporate governance, accounting standards, and regulatory compliance. Delay in reporting stock option values, changes in corporate governance provisions, investigations into accounting firms' practices.*

Summary A-2: *Corporate governance, finance, and regulatory compliance. US companies' delayed stock option reporting, audit findings, and concessions on corporate governance. Back taxes owed, and fines imposed on accounting firms.*

Summary A-3: *Corporate finance, governance, and regulatory compliance. Delay in financial reporting, tax fines, investigations into accounting firms, changes in corporate governance provisions, and SEC regulations.*

Summary A-4: *Corporate governance challenges and regulatory compliance in U.S. businesses. U.S. American companies delayed reporting stock option valuations affecting investor trust. Federal courts intervened over FCC regulations impacting telecommunications. Accounting scandals led to investigations into major auditing firms. Companies faced penalties for manipulating earnings and inadequate corporate governance structures.*

Figure 4.4: Example of a survey section showing the presentation of a text cluster with multiple AI-generated summaries

Choose the specific group you would like to use

Qualified AI taskers

Qualified AI taskers

Description
Participants who have passed our AI Task Assessment, demonstrating advanced reasoning, fact-checking, and writing skills.

Qualified AI taskers Count
1,211

Figure 4.5: Qualified AI Taskers group on Prolific

Evaluation Criteria

Each summary was evaluated using three Likert-scale questions as shown in Figure 4.9, with ratings provided on a scale from 1 to 5:

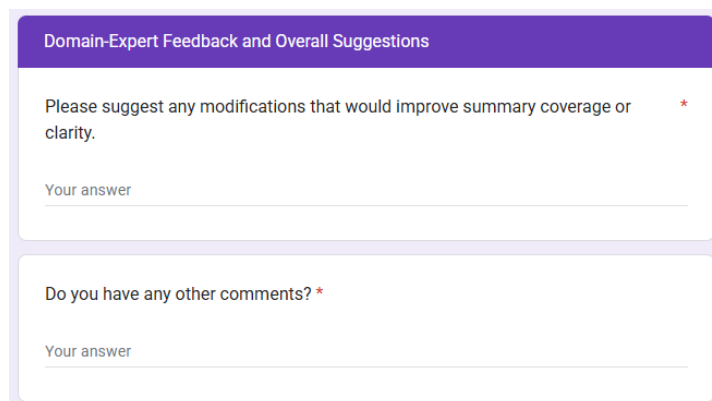
1. **Coverage of Content:** Ratings ranged from “Very little coverage” (1) to “Complete coverage” (5), and were used to measure how comprehensively the summary reflected the original cluster.
2. **Ease of Comprehension:** Ratings ranged from “Impossible to comprehend” (1) to “Very easy to comprehend” (5), and were used to assess the clarity of the summary for non-experts.
3. **Relevance:** Ratings ranged from “Irrelevant” (1) to “Highly relevant” (5), and were used to judge how closely the summary aligned with the main themes of the original text.

To ensure attentiveness, two attention-check questions were embedded randomly throughout the survey’s sections, prompting participants to deliberately select a specified option as shown in Figure 4.8. Additionally, text boxes were included to allow for open-ended qualitative feedback as illustrated in Figure 4.6.

Participant Demographics

Demographic information was collected to contextualize responses based on participants’ backgrounds as shown in Figure 4.7. The following information was gathered:

- Age range
- Highest completed level of education
- Current occupation and role
- Domain of expertise (e.g., AI, Finance, Education)



Domain-Expert Feedback and Overall Suggestions

Please suggest any modifications that would improve summary coverage or clarity. *

Your answer _____

Do you have any other comments? *

Your answer _____

Figure 4.6: Open-ended feedback section where participants provided suggestions for improving summary clarity and coverage

- Self-assessed technical proficiency (scale of 1–5)

The survey results would provide insights about users preferences whether for example, longer summaries or archetypes are preferred than shorter ones, which LLM generate simpler, clearer, and complete archetypes. The survey is also expected to inform future improvements which could be taken into consideration in generalization models and evaluation frameworks, particularly in domains where precision, clarity, and semantic completeness are essential.

4.6.5 Clustering of Generated Archetypes vs. Full Text

To evaluate the semantic disjointness of the domains' generated archetypes, a comparative clustering analysis was conducted between the original full-text segments and their corresponding archetypes. The focus of this evaluation was to determine whether the distilled archetypes, produced via prompt-based generation by multiple LLMs, yield more coherent and domain-specific clusters compared to the original texts.

A mixed-domain corpus was first clustered using both k -means and recursive hierarchical clustering algorithms. Subsequently, archetypes corresponding to each original segment generated by multiple LLMs were subjected to the same clustering procedures. All textual inputs were embedded into high-dimensional semantic space using `jinaai/jina-embeddings-v3` to ensure consistency across representations.

The clustering outputs were then analyzed for domains' disjointness by inspecting the distribution of domain labels within each cluster. By comparing the clustering results of the full text and generated archetypes across different LLMs and clustering methods, insights into the effectiveness of archetypal distillation for semantic abstraction and domain disambiguation were obtained.

4.6. EVALUATION TECHNIQUES

The figure shows a vertical questionnaire form with five sections, each containing radio button options and an 'Other' field with a text input line.

- Age Range ***
 - 18-24
 - 25-34
 - 35-44
 - 45-54
 - 55-64
 - 65+
- Highest Level of Education Completed ***
 - High school diploma
 - Associate degree
 - Bachelor's degree
 - Master's degree
 - PhD or equivalent
 - Other: _____
- Current Occupation/Role ***
 - Student (undergraduate / graduate)
 - Researcher / Academic
 - Data Scientist / ML Engineer
 - Industry Professional (other technical role)
 - Industry Professional (non-technical)
 - Other: _____
- Field/Industry ***
 - Computer Science / AI
 - Healthcare
 - Finance
 - Education
 - Marketing
 - Other: _____
- Self-Assessment of Technical Skills ***

1 2 3 4 5

Very basic computer skills (email, web browsing) Specialized (active NLP researcher or ML engineer)

Figure 4.7: Demographic questionnaire used to collect information on participants' age, education, profession, domain, and technical skills

Does Summary B-2 cover the complete variety of the content provided by the clustered business domain text in Cluster B? *

1 2 3 4 5

Very little coverage for the provided content Complete coverage for the provided content

Is Summary B-2 easy for non-experts in the business domain to comprehend? *

1 2 3 4 5

Impossible to comprehend Very easy to comprehend

The third/middle choice is the correct answer. Please select it! *

1 2 3 4 5

Irrelevant Highly relevant

For the next 3 questions, **Summary B-3** will be needed so it is provided again for easier access.

Summary B-3: *Statements regarding business progress, growth, and profitability. Confidence in business strategies leading to growth. Improvements in business performance.*

Figure 4.8: Embedded attention-check question instructing participants to select the middle answer to ensure attentiveness

For the next 3 questions, **Summary A-1** will be needed so it is provided again for easier access.

Summary A-1: *Corporate governance, accounting standards, and regulatory compliance. Delay in reporting stock option values, changes in corporate governance provisions, investigations into accounting firms' practices.*

How relevant and meaningful is Summary A-1 according to the provided clustered * business domain text in Cluster A?

1 2 3 4 5

Irrelevant Highly relevant

Does Summary A-1 cover the complete variety of the content provided by the clustered business domain text in Cluster A? *

1 2 3 4 5

Very little coverage for the provided content Complete coverage for the provided content

Is Summary A-1 easy for non-experts in the business domain to comprehend? *

1 2 3 4 5

Impossible to comprehend Very easy to comprehend

Figure 4.9: Evaluation questions used to assess summary quality in terms of coverage, clarity, and relevance

5 Results

This chapter presents the outcomes of the evaluation methods introduced in the methodology chapter. It begins with a comparative analysis of semantic search performance when using the full-text dataset versus the generated archetypes. This is followed by an evaluation of multiple text classifiers, identifying the most effective one for fine-tuning. Fine-tuning is conducted in two configurations: once using the original full-text and once using archetypes generated by various LLMs. This comparison aims to assess whether the archetypes generalize better than raw text data.

In addition to classification performance, computational aspects such as training time and efficiency are reported. The chapter also includes a detailed analysis of the survey responses, highlighting participant preferences for archetypes based on interpretability, relevance, and completeness. Lastly, the degree of domain disjointness exhibited by the full-text data compared to the generated archetypes is evaluated, offering insights into their ability to separate and encapsulate semantic domains.

Table 5.1 provides an overview of the dataset sizes and the domains involved in all evaluation experiments whose results are detailed in the following sections. The most common domains among all datasets were chosen which are sport, business, and tech. All experiments except the survey and initial text classifiers comparison are carried out twice, once with BBC dataset behaving as the source knowledge base, and another time with AG News dataset.

| Dataset | Size | Domains |
|--------------------|--------|-----------------------|
| AG News | 10,000 | Sport, Business, Tech |
| BBC | 1,423 | Sport, Business, Tech |
| Only Sport | 6,660 | Sport |
| Short Descriptions | 9,540 | Business, Sport |
| 20 News Groups | 1,947 | Sport |

Table 5.1: Datasets Details used for Evaluation and Training

5.1 Semantic Search Evaluation Results

Initially, the focus is drawn towards the idea if the LLMs' generated archetypes cover the same scope as the original full-text data. Thus, semantic search is utilized and results are shown in the next two subsections. The evaluation metric used is the accuracy.

5.1.1 BBC Dataset Generated Archetypes vs. Full-Text

Table 5.2 shows that when BBC dataset acts as the base knowledge, mistral-7B model achieves the highest results compared to the other models except for short descriptions dataset which meta-llama-3-8B outperforms in this case. However, as shown in Table 5.3, archetypes' results exceed BBC full-text except for AG dataset.

| Model | AG | Only Sport | 20 News Groups | Short Descriptions |
|--------------------------------|---------------|---------------|----------------|--------------------|
| mistral-7B | 0.7450 | 0.8715 | 0.9004 | 0.6313 |
| mistral-7B (Top-3) | 0.7664 | 0.9038 | 0.9178 | 0.6552 |
| microsoft/phi-4 | 0.7446 | 0.8749 | 0.9070 | 0.6173 |
| microsoft/phi-4 (Top-3) | 0.7620 | 0.8927 | 0.9024 | 0.6329 |
| miniCPM3-4B | 0.7390 | 0.8326 | 0.8629 | 0.6201 |
| miniCPM3-4B (Top-3) | 0.7532 | 0.8581 | 0.8798 | 0.6312 |
| meta-llama-3-8B | 0.7246 | 0.8736 | 0.8870 | 0.6610 |
| meta-llama-3-8B (Top-3) | 0.7469 | 0.8925 | 0.9070 | 0.6741 |

Table 5.2: BBC archetypes semantic search accuracy results across various LLMs and datasets. Top-3 indicates averaged score from top-3 retrieved results. Bold values indicate the highest score in each column.

| Dataset | Accuracy@1 | Accuracy@3 |
|--------------------|------------|------------|
| AG | 0.8303 | 0.8439 |
| Only Sport | 0.8539 | 0.8877 |
| 20 News Groups | 0.8038 | 0.8731 |
| Short descriptions | 0.6589 | 0.6541 |

Table 5.3: BBC full-text semantic search accuracy across four datasets. Accuracy@1 is for the top retrieved result, while Accuracy@3 is averaged over the top-3 retrieved results.

5.1.2 AG Dataset Generated Archetypes vs. Full-Text

| Dataset | Accuracy@1 | Accuracy@3 |
|--------------------|------------|------------|
| Short Descriptions | 0.6878 | 0.7055 |
| 20 News Groups | 0.9111 | 0.9384 |
| BBC | 0.9416 | 0.9684 |
| Only Sport | 0.9351 | 0.9514 |

Table 5.4: AG full-text semantic search accuracy across four datasets. Accuracy@1 is for the top retrieved result, while Accuracy@3 is averaged over the top-3 retrieved results.

5.2. INITIAL TEXT CLASSIFIERS COMPARISON

| Model | Short Descriptions | 20 News Groups | BBC | Only Sport |
|--------------------------------|--------------------|----------------|---------------|---------------|
| mistral-7B | 0.7531 | 0.9553 | 0.9198 | 0.9668 |
| mistral-7B (Top-3) | 0.7755 | 0.9569 | 0.9346 | 0.9736 |
| microsoft/phi-4 | 0.7427 | 0.9471 | 0.9233 | 0.9686 |
| microsoft/phi-4 (Top-3) | 0.7642 | 0.9563 | 0.9381 | 0.9773 |
| miniCPM3-4B | 0.7501 | 0.9605 | 0.9058 | 0.9721 |
| miniCPM3-4B (Top-3) | 0.7642 | 0.9671 | 0.9219 | 0.9794 |
| meta-llama-3-8B | 0.7563 | 0.9471 | 0.9163 | 0.9650 |
| meta-llama-3-8B (Top-3) | 0.7769 | 0.9527 | 0.9346 | 0.9734 |

Table 5.5: AG archetypes semantic search accuracy results across various LLMs and datasets. Top-3 indicates averaged score from top-3 retrieved results. Bold values indicate the highest score in each column.

As illustrated in both Tables 5.4 & 5.5, LLMs’ archetypes’ semantic search accuracies across all datasets is more than full-text’s accuracies except for BBC dataset, when AG dataset acts as the knowledge base.

5.2 Initial Text Classifiers Comparison

The following step is to examine whether the LLM-generated archetypes generalize better than the full-text when fine-tuned on different text classifiers. To this end, three top-performing classifiers were initially fine-tuned on the BBC full-text dataset for three epochs, then evaluated on the BBC, 20 news groups, and short descriptions datasets. As shown in Table 5.6, both deberta-v3-base and roberta-base performed well. However, due to time constraints, the roberta-base model was selected to proceed with the full-text versus archetypes comparison, as it demonstrated a significant performance advantage over deberta-v3-base on the short descriptions dataset.

| Model | BBC | 20 News Groups | Short Descriptions |
|------------------------|---------------|----------------|--------------------|
| bert-base-cased | 0.9937 | 0.4429 | 0.2462 |
| deberta-v3-base | 0.9906 | 0.7116 | 0.3090 |
| roberta-base | 0.9942 | 0.6530 | 0.6963 |

Table 5.6: Accuracy of text classifiers across different datasets. Bold values indicate the best accuracy per dataset.

5.3 BBC Dataset Fine-Tuning: Roberta-Base Text Classifier

The main and one of the most crucial evaluation steps was to compare fine-tuned roberta-base model with full-text versus archetypes as mentioned before, to check if LLMs’ generated archetypes would generalize on other datasets more than full-text. All of the subsequent fine-tuning experiments were carried out with 1, 3, and 5 epochs. The evaluation metrics used are accuracy, precision, recall, and F1-score. In this section, BBC dataset behaves as the the knowledge base, so all archetypes and full-text data results are based on fine-tuning BBC data. The next section will cover the results when AG dataset behaves as the knowledge base.

5.3.1 AG Dataset

| Model | Accuracy | Precision | Recall | F1-Score |
|-----------------------------|----------------------|--------------------|--------------------|--------------------|
| <i>1 Epoch</i> | | | | |
| microsoft/phi-4 | 0.0000 | 0.00 | 0.00 | 0.00 |
| meta-llama-3-8B | 0.0000 | 0.00 | 0.00 | 0.00 |
| miniCPM3-4B | 0.4449 | 0.43 | 0.44 | 0.36 |
| mistral-7B | 0.0165 | 0.64 | 0.02 | 0.03 |
| <u>BBC Full-Text</u> | <u>0.8894</u> | <u>0.89</u> | <u>0.88</u> | <u>0.88</u> |
| <i>3 Epochs</i> | | | | |
| microsoft/phi-4 | 0.8169 | 0.91 | 0.82 | 0.86 |
| meta-llama-3-8B | 0.8257 | 0.88 | 0.83 | 0.84 |
| <u>miniCPM3-4B</u> | 0.8459 | <u>0.91</u> | <u>0.85</u> | <u>0.87</u> |
| mistral-7B | 0.8403 | 0.90 | 0.84 | 0.86 |
| BBC Full-Text | 0.8507 | 0.88 | 0.85 | 0.85 |
| <i>5 Epochs</i> | | | | |
| microsoft/phi-4 | 0.8349 | 0.90 | 0.83 | 0.86 |
| meta-llama-3-8B | 0.8426 | 0.90 | 0.84 | 0.87 |
| miniCPM3-4B | 0.8151 | 0.89 | 0.82 | 0.84 |
| mistral-7B | 0.8056 | 0.89 | 0.81 | 0.83 |
| BBC Full-Text | 0.8628 | 0.88 | 0.86 | 0.86 |

Table 5.7: Classification performance of BBC dataset’s archetypes vs. full-text on AG dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column.

5.3.2 Only Sport Dataset

| Model | Accuracy | Precision | Recall | F1-Score |
|-------------------------------|----------------------|--------------------|--------------------|--------------------|
| <i>1 Epoch</i> | | | | |
| All Models | 0.0000 | 0.00 | 0.00 | 0.00 |
| BBC Full-Text | 0.8551 | 1.00 | 0.86 | 0.92 |
| <i>3 Epochs</i> | | | | |
| microsoft/phi-4 | 0.9081 | 1.00 | 0.91 | 0.95 |
| <u>meta-llama-3-8B</u> | <u>0.9272</u> | <u>1.00</u> | <u>0.93</u> | <u>0.96</u> |
| miniCPM3-4B | 0.8835 | 1.00 | 0.88 | 0.94 |
| mistral-7B | 0.8778 | 1.00 | 0.88 | 0.93 |
| BBC Full-Text | 0.9013 | 1.00 | 0.90 | 0.95 |
| <i>5 Epochs</i> | | | | |
| microsoft/phi-4 | 0.9029 | 1.00 | 0.90 | 0.95 |
| <u>meta-llama-3-8B</u> | <u>0.9203</u> | <u>1.00</u> | <u>0.92</u> | <u>0.96</u> |
| miniCPM3-4B | 0.8994 | 1.00 | 0.90 | 0.95 |
| mistral-7B | 0.8635 | 1.00 | 0.86 | 0.93 |
| BBC Full-Text | 0.9126 | 1.00 | 0.91 | 0.95 |

Table 5.8: Classification performance of BBC dataset’s archetypes vs. full-text on Only Sport dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column.

5.3.3 Short descriptions Dataset

| Model | Accuracy | Precision | Recall | F1-Score |
|-----------------------------|---------------|-------------|-------------|-------------|
| <i>1 Epoch</i> | | | | |
| microsoft/phi-4 | 0.0000 | 0.00 | 0.00 | 0.00 |
| meta-llama-3-8B | 0.0000 | 0.00 | 0.00 | 0.00 |
| miniCPM3-4B | 0.4882 | 0.29 | 0.49 | 0.37 |
| mistral-7B | 0.0017 | 0.54 | 0.00 | 0.00 |
| BBC Full-Text | 0.6826 | 0.91 | 0.68 | 0.78 |
| <i>3 Epochs</i> | | | | |
| microsoft/phi-4 | 0.5729 | 0.93 | 0.57 | 0.70 |
| meta-llama-3-8B | 0.6582 | 0.94 | 0.66 | 0.77 |
| <u>miniCPM3-4B</u> | 0.6189 | 0.96 | 0.62 | 0.75 |
| mistral-7B | 0.6567 | 0.96 | 0.66 | 0.78 |
| BBC Full-Text | 0.7947 | 0.91 | 0.79 | 0.85 |
| <i>5 Epochs</i> | | | | |
| microsoft/phi-4 | 0.5399 | 0.93 | 0.54 | 0.67 |
| meta-llama-3-8B | 0.6240 | 0.94 | 0.62 | 0.74 |
| <u>miniCPM3-4B</u> | 0.6493 | 0.95 | 0.65 | 0.77 |
| mistral-7B | 0.6537 | 0.94 | 0.65 | 0.77 |
| <u>BBC Full-Text</u> | 0.7967 | 0.89 | 0.80 | 0.84 |

Table 5.9: Classification performance of BBC dataset’s archetypes vs. full-text on short description dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column.

5.3.4 BBC Dataset

| Model | Accuracy | Precision | Recall | F1-Score |
|----------------------|---------------|-------------|-------------|-------------|
| <i>1 Epoch</i> | | | | |
| microsoft/phi-4 | 0.0000 | 0.00 | 0.00 | 0.00 |
| meta-llama-3-8B | 0.0000 | 0.00 | 0.00 | 0.00 |
| miniCPM3-4B | 0.4065 | 0.42 | 0.41 | 0.28 |
| mistral-7B | 0.0000 | 0.00 | 0.00 | 0.00 |
| BBC Full-Text | 0.9944 | 0.99 | 0.99 | 0.99 |
| <i>3 Epochs</i> | | | | |
| microsoft/phi-4 | 0.8805 | 0.97 | 0.88 | 0.92 |
| meta-llama-3-8B | 0.9423 | 0.98 | 0.94 | 0.96 |
| miniCPM3-4B | 0.9374 | 0.98 | 0.94 | 0.96 |
| mistral-7B | 0.9346 | 0.98 | 0.93 | 0.96 |
| BBC Full-Text | 0.9993 | 1.00 | 1.00 | 1.00 |
| <i>5 Epochs</i> | | | | |
| microsoft/phi-4 | 0.9332 | 0.98 | 0.93 | 0.95 |
| meta-llama-3-8B | 0.9459 | 0.98 | 0.95 | 0.96 |
| miniCPM3-4B | 0.9332 | 0.98 | 0.93 | 0.96 |
| mistral-7B | 0.9121 | 0.97 | 0.91 | 0.94 |
| BBC Full-Text | 0.9993 | 1.00 | 1.00 | 1.00 |

Table 5.10: Classification performance of BBC dataset’s archetypes vs. full-text on BBC dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column.

5.3.5 20 News Groups Dataset

| Model | Accuracy | Precision | Recall | F1-Score |
|-------------------------------|----------------------|--------------------|--------------------|--------------------|
| <i>1 Epoch</i> | | | | |
| All Models | 0.0000 | 0.00 | 0.00 | 0.00 |
| BBC Full-Text | 0.7946 | 1.00 | 0.79 | 0.89 |
| <i>3 Epochs</i> | | | | |
| microsoft/phi-4 | 0.9147 | 1.00 | 0.91 | 0.96 |
| meta-llama-3-8B | 0.9240 | 1.00 | 0.92 | 0.96 |
| miniCPM3-4B | 0.8603 | 1.00 | 0.86 | 0.92 |
| mistral-7B | 0.8485 | 1.00 | 0.85 | 0.92 |
| BBC Full-Text | 0.7946 | 1.00 | 0.79 | 0.89 |
| <i>5 Epochs</i> | | | | |
| microsoft/phi-4 | 0.8860 | 1.00 | 0.89 | 0.94 |
| <u>meta-llama-3-8B</u> | <u>0.9373</u> | <u>1.00</u> | <u>0.94</u> | <u>0.97</u> |
| miniCPM3-4B | 0.9183 | 1.00 | 0.92 | 0.96 |
| mistral-7B | 0.8891 | 1.00 | 0.89 | 0.94 |
| BBC Full-Text | 0.9281 | 1.00 | 0.93 | 0.96 |

Table 5.11: Classification performance of BBC dataset’s archetypes vs. full-text on 20 news groups dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column.

5.4 AG Dataset Fine-Tuning: Roberta-Base Text Classifier

The rest of the classifier’s fine-tuning results continues in this section, with AG acting as the knowledge base.

5.4.1 AG Dataset

| Model | Accuracy | Precision | Recall | F1-Score |
|---------------------|---------------|-------------|-------------|-------------|
| <i>1 Epoch</i> | | | | |
| microsoft/phi-4 | 0.9037 | 0.91 | 0.90 | 0.90 |
| meta-llama-3-8B | 0.9059 | 0.91 | 0.91 | 0.91 |
| miniCPM3-4B | 0.8946 | 0.90 | 0.89 | 0.89 |
| mistral-7B | 0.8956 | 0.89 | 0.90 | 0.89 |
| AG Full-Text | 0.9615 | 0.96 | 0.96 | 0.96 |
| <i>3 Epochs</i> | | | | |
| microsoft/phi-4 | 0.9039 | 0.91 | 0.90 | 0.90 |
| meta-llama-3-8B | 0.8838 | 0.89 | 0.88 | 0.88 |
| miniCPM3-4B | 0.8989 | 0.90 | 0.90 | 0.90 |
| mistral-7B | 0.8906 | 0.89 | 0.90 | 0.89 |
| AG Full-Text | 0.9775 | 0.98 | 0.98 | 0.98 |
| <i>5 Epochs</i> | | | | |
| microsoft/phi-4 | 0.9037 | 0.91 | 0.90 | 0.90 |
| meta-llama-3-8B | 0.8926 | 0.89 | 0.89 | 0.89 |
| miniCPM3-4B | 0.9019 | 0.90 | 0.90 | 0.90 |
| mistral-7B | 0.9013 | 0.90 | 0.90 | 0.90 |
| AG Full-Text | 0.9861 | 0.99 | 0.99 | 0.99 |

Table 5.12: Classification performance of AG dataset’s archetypes vs. full-text on AG dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column.

5.4.2 Only Sport Dataset

| Model | Accuracy | Precision | Recall | F1-Score |
|-------------------------------|----------------------|--------------------|--------------------|--------------------|
| <i>1 Epoch</i> | | | | |
| mistral-7B | 0.9677 | 1.00 | 0.97 | 0.98 |
| miniCPM3-4B | 0.9638 | 1.00 | 0.96 | 0.98 |
| meta-llama-3-8B | 0.9488 | 1.00 | 0.95 | 0.97 |
| microsoft/phi-4 | 0.9209 | 1.00 | 0.92 | 0.96 |
| AG Full-Text | 0.9081 | 1.00 | 0.91 | 0.95 |
| <i>3 Epochs</i> | | | | |
| mistral-7B | 0.9523 | 1.00 | 0.95 | 0.98 |
| miniCPM3-4B | 0.9649 | 1.00 | 0.96 | 0.98 |
| <u>meta-llama-3-8B</u> | <u>0.9832</u> | <u>1.00</u> | <u>0.98</u> | <u>0.99</u> |
| microsoft/phi-4 | 0.9308 | 1.00 | 0.93 | 0.96 |
| AG Full-Text | 0.9041 | 1.00 | 0.90 | 0.95 |
| <i>5 Epochs</i> | | | | |
| mistral-7B | 0.9131 | 1.00 | 0.91 | 0.95 |
| miniCPM3-4B | 0.9578 | 1.00 | 0.96 | 0.98 |
| <u>meta-llama-3-8B</u> | <u>0.9730</u> | <u>1.00</u> | <u>0.97</u> | <u>0.99</u> |
| microsoft/phi-4 | 0.9146 | 1.00 | 0.91 | 0.96 |
| AG Full-Text | 0.8996 | 1.00 | 0.90 | 0.95 |

Table 5.13: Classification performance of AG dataset’s archetypes vs. full-text on only sport dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column.

5.4.3 Short descriptions Dataset

| Model | Accuracy | Precision | Recall | F1-Score |
|-------------------------------|----------------------|--------------------|--------------------|--------------------|
| <i>1 Epoch</i> | | | | |
| mistral-7B | 0.7115 | 0.90 | 0.71 | 0.78 |
| miniCPM3-4B | 0.7835 | 0.90 | 0.78 | 0.84 |
| meta-llama-3-8B | 0.7748 | 0.91 | 0.77 | 0.84 |
| microsoft/phi-4 | 0.6211 | 0.94 | 0.62 | 0.74 |
| AG Full-Text | 0.7613 | 0.92 | 0.76 | 0.83 |
| <i>3 Epochs</i> | | | | |
| mistral-7B | 0.5706 | 0.94 | 0.57 | 0.68 |
| <u>miniCPM3-4B</u> | 0.7893 | 0.92 | 0.79 | <u>0.85</u> |
| <u>meta-llama-3-8B</u> | <u>0.7949</u> | 0.89 | <u>0.79</u> | 0.84 |
| <u>microsoft/phi-4</u> | 0.6036 | <u>0.96</u> | 0.60 | 0.73 |
| AG Full-Text | 0.7686 | 0.91 | 0.77 | 0.83 |
| <i>5 Epochs</i> | | | | |
| mistral-7B | 0.6071 | 0.95 | 0.61 | 0.73 |
| miniCPM3-4B | 0.7369 | 0.91 | 0.74 | 0.81 |
| meta-llama-3-8B | 0.7379 | 0.90 | 0.74 | 0.79 |
| microsoft/phi-4 | 0.6074 | 0.96 | 0.61 | 0.74 |
| AG Full-Text | 0.7751 | 0.89 | 0.78 | 0.83 |

Table 5.14: Classification performance of AG dataset’s archetypes vs. full-text on short descriptions dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column.

5.4.4 BBC Dataset

| Model | Accuracy | Precision | Recall | F1-Score |
|-------------------------------|----------------------|--------------------|--------------------|--------------------|
| <i>1 Epoch</i> | | | | |
| mistral-7B | 0.9761 | 0.98 | 0.98 | 0.98 |
| miniCPM3-4B | 0.9796 | 0.98 | 0.98 | 0.98 |
| meta-llama-3-8B | 0.9838 | 0.98 | 0.98 | 0.98 |
| microsoft/phi-4 | 0.9810 | 0.98 | 0.98 | 0.98 |
| AG Full-Text | 0.7518 | 0.86 | 0.75 | 0.74 |
| <i>3 Epochs</i> | | | | |
| mistral-7B | 0.9564 | 0.96 | 0.96 | 0.96 |
| miniCPM3-4B | 0.9712 | 0.97 | 0.97 | 0.97 |
| meta-llama-3-8B | 0.9761 | 0.98 | 0.98 | 0.98 |
| microsoft/phi-4 | 0.9733 | 0.97 | 0.97 | 0.97 |
| AG Full-Text | 0.9114 | 0.93 | 0.91 | 0.91 |
| <i>5 Epochs</i> | | | | |
| mistral-7B | 0.9684 | 0.97 | 0.97 | 0.97 |
| miniCPM3-4B | 0.9768 | 0.98 | 0.98 | 0.98 |
| meta-llama-3-8B | 0.9810 | 0.98 | 0.98 | 0.98 |
| <u>microsoft/phi-4</u> | <u>0.9866</u> | <u>0.99</u> | <u>0.99</u> | <u>0.99</u> |
| AG Full-Text | 0.8481 | 0.90 | 0.85 | 0.85 |

Table 5.15: Classification performance of AG dataset’s archetypes vs. full-text on BBC dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column.

5.4.5 20 News Groups Dataset

| Model | Accuracy | Precision | Recall | F1-Score |
|-------------------------------|----------------------|--------------------|--------------------|--------------------|
| <i>1 Epoch</i> | | | | |
| mistral-7B | 0.9379 | 1.00 | 0.94 | 0.97 |
| miniCPM3-4B | 0.9414 | 1.00 | 0.94 | 0.97 |
| meta-llama-3-8B | 0.9327 | 1.00 | 0.93 | 0.97 |
| microsoft/phi-4 | 0.8844 | 1.00 | 0.88 | 0.94 |
| AG Full-Text | 0.6538 | 1.00 | 0.65 | 0.79 |
| <i>3 Epochs</i> | | | | |
| mistral-7B | 0.9070 | 1.00 | 0.91 | 0.95 |
| miniCPM3-4B | 0.9548 | 1.00 | 0.95 | 0.98 |
| <u>meta-llama-3-8B</u> | <u>0.9697</u> | <u>1.00</u> | <u>0.97</u> | <u>0.98</u> |
| microsoft/phi-4 | 0.9091 | 1.00 | 0.91 | 0.95 |
| AG Full-Text | 0.7365 | 1.00 | 0.74 | 0.85 |
| <i>5 Epochs</i> | | | | |
| mistral-7B | 0.8783 | 1.00 | 0.88 | 0.94 |
| miniCPM3-4B | 0.9111 | 1.00 | 0.91 | 0.95 |
| <u>meta-llama-3-8B</u> | <u>0.9276</u> | <u>1.00</u> | <u>0.93</u> | <u>0.96</u> |
| microsoft/phi-4 | 0.8593 | 1.00 | 0.86 | 0.92 |
| AG Full-Text | 0.6667 | 1.00 | 0.67 | 0.80 |

Table 5.16: Classification performance of AG dataset’s archetypes vs. full-text on 20 news groups dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column.

5.5 Computational Efficiency of Fine-Tuning: Full-Text vs. Archetypes

Tables 5.17 & 5.18 reveal computational efficiency-related results. The tables show the time taken for each of AG and BBC datasets to be fine-tuned by the classifier with 1, 2, and 3 epochs. Table 5.19 shows the GPU configuration used to run all experiments.

| Model | 1 Epoch | 3 Epochs | 5 Epochs |
|------------------------|-----------|-----------|-----------|
| microsoft/phi-4 | 20 | 44 | 67 |
| meta-llama-3-8B | 17 | 40 | 61 |
| miniCPM3-4B | 18 | 41 | 65 |
| mistral-7B | 18 | 42 | 66 |
| BBC Full-Text | 48 | 136 | 229 |

Table 5.17: Training times (in seconds) for fine-tuning classifier on BBC full-text dataset & archetypes. Bold values indicate the lowest training time in each column.

| Model | 1 Epoch | 3 Epochs | 5 Epochs |
|------------------------|-----------|------------|------------|
| mistral-7B | 49 | 129 | 210 |
| miniCPM3-4B | 46 | 126 | 207 |
| meta-llama-3-8B | 44 | 121 | 198 |
| microsoft/phi-4 | 47 | 127 | 208 |
| AG Full-Text | 2676 | 7872 | 13115 |

Table 5.18: Training times (in seconds) for fine-tuning classifier on AG full-text dataset & archetypes. Bold values indicate the lowest training time in each column.

| Component | Specification |
|----------------|-----------------------------|
| GPU Model | NVIDIA Tesla V100-PCIE-16GB |
| CUDA Version | 12.4 |
| Driver Version | 550.54.15 |
| Memory | 16 GB |

Table 5.19: GPU configuration used during experiments

5.6 Survey Analysis

This section contains aggregated results collected from 100 participants about the conducted survey to assess human's archetypes general preferences in addition to which of the models' archetypes exceeds the others in terms of context windows' clusters interpretability, relevance, and completeness for the domain.

5.6.1 Overall LLMs' Archetypes Preferences

Figure 5.1 explicitly shows that phi-4 model's archetypes were preferred among all other models' archetypes by the majority of participants.

5.6.2 Averaged Archetypes' Interpretability, Relevance, and Completeness

Figure 5.2 suggests that phi-4 model's archetypes were favoured among all other models' archetypes in terms of comprehension, relevance, and coverage, whether for all aggregated results, aggregated roles, or participants who work in computer science and AI field. However, there was just one slight difference in archetypes' comprehension by the ones who work in AI and computer science field in which Mistral model exceeded Phi-4.

5.6.3 Participants Feedback Summary

Participant feedback revealed a strong preference for summaries/archetypes that they were way easier to comprehend, particularly when dealing with complex or jargon-heavy content. Some found summaries that attempted to infer causality or meaning not explicitly stated in the original content to be misleading or unhelpful. A recurring issue was the fragmented nature of the cluster source of context windows, which often lacked sufficient context or were contradictory, making summarization inherently difficult. Despite the challenges, numerous participants appreciated the study's depth and intellectual demand, describing it as interesting, well-designed, and even "artful" in some cases.

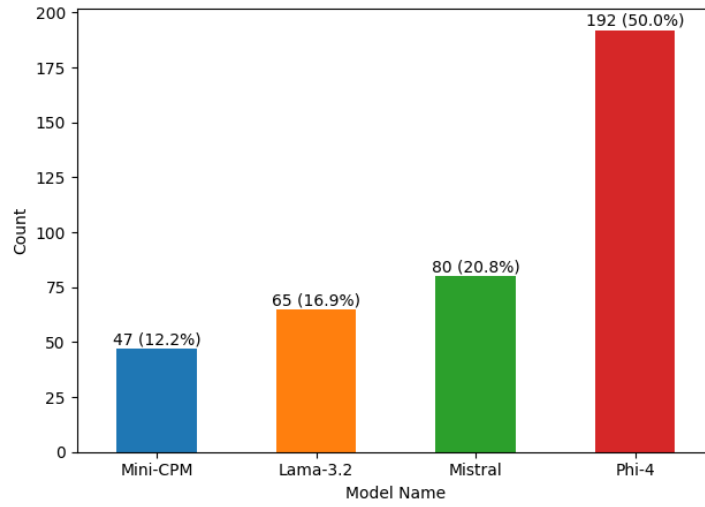
5.7 Domains' Disjointness Comparison: Archetypes vs. Full-Text Clustering

To assess the capability of archetype-based representations in capturing semantically distinct domains, this section presents a comparative analysis between archetypes and full-text inputs using two widely-adopted clustering approaches: K-means and recursive hierarchical clustering. The evaluation is conducted on both the AG News and BBC datasets, with the latter extended to include the additional categories politics and entertainment to test the generalizability of domain separability. As K-means clustering configuration must include a specific number of clusters, it was inputted equals to the number of given domains.

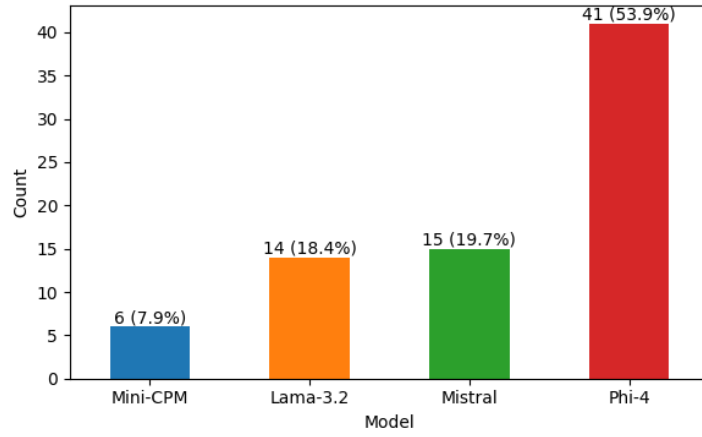
5.7. DOMAINS' DISJOINTNESS COMPARISON: ARCHETYPES VS. FULL-TEXT CLUSTERING

The valid and invalid clusters labels shown in Figures 5.3 & 5.4 & 5.5 & 5.6 refer to whether the generated clusters are composed of one domain (*valid*) or mixed domains (*invalid*). The corresponding results agree that archetypes result in more significantly separable domains than full-text, which signals to more domains' disjointness specifically in recursive hierarchical clustering. This analysis seeks to answer whether archetypes can better encapsulate the structural boundaries between domains compared to their full-text counterparts, thus offering more disjoint and semantically coherent groupings.

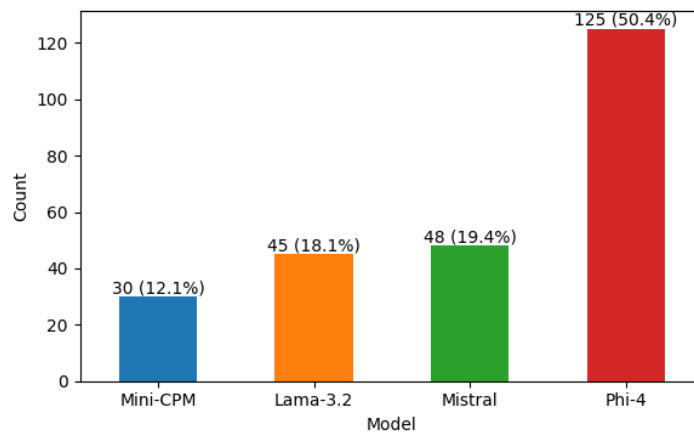
5.7. DOMAINS' DISJOINTNESS COMPARISON: ARCHETYPES VS. FULL-TEXT CLUSTERING



(a) Overall distribution of LLM-preferred archetypes across all participant groups



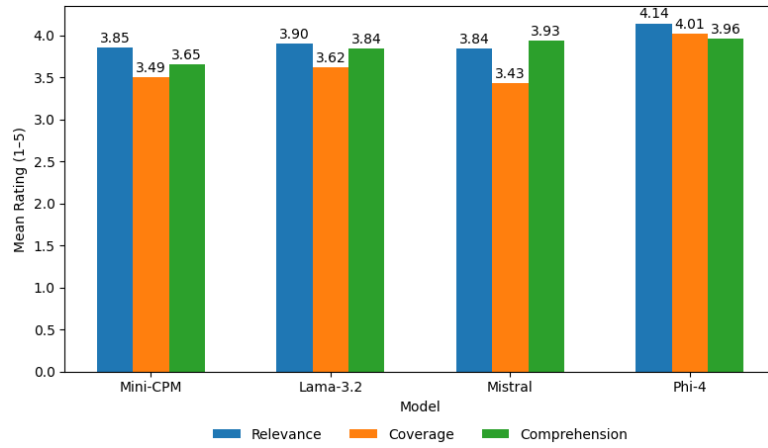
(b) Archetype preferences by participants in the Computer Science and AI fields



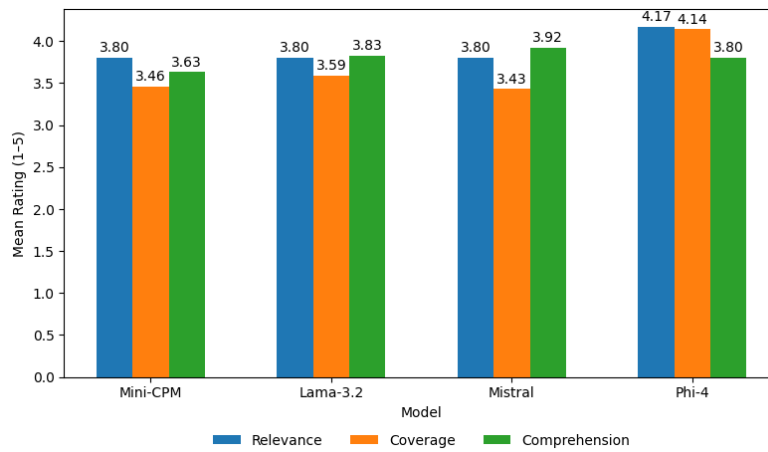
(c) Archetype preferences segmented by role: Researcher, Data Scientist, and Industry Professional

Figure 5.1: LLMs' archetype preferences across overall participants, disciplines, and roles.

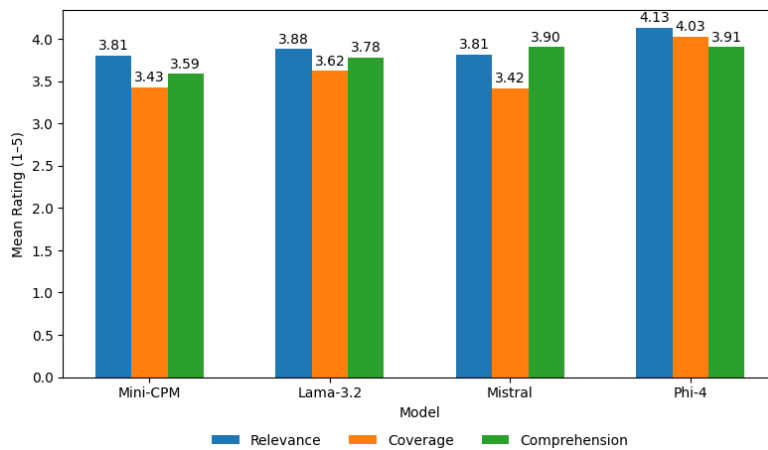
5.7. DOMAINS' DISJOINTNESS COMPARISON: ARCHETYPES VS. FULL-TEXT CLUSTERING



(a) Overall averaged ratings for archetype interpretability, relevance, and completeness



(b) Average ratings by Computer Science and AI participants



(c) Average ratings per role: Researcher, Data Scientist, Industry Professional

Figure 5.2: Averaged ratings for archetype interpretability, relevance, and completeness across different groups.

5.7. DOMAINS' DISJOINTNESS COMPARISON: ARCHETYPES VS. FULL-TEXT CLUSTERING

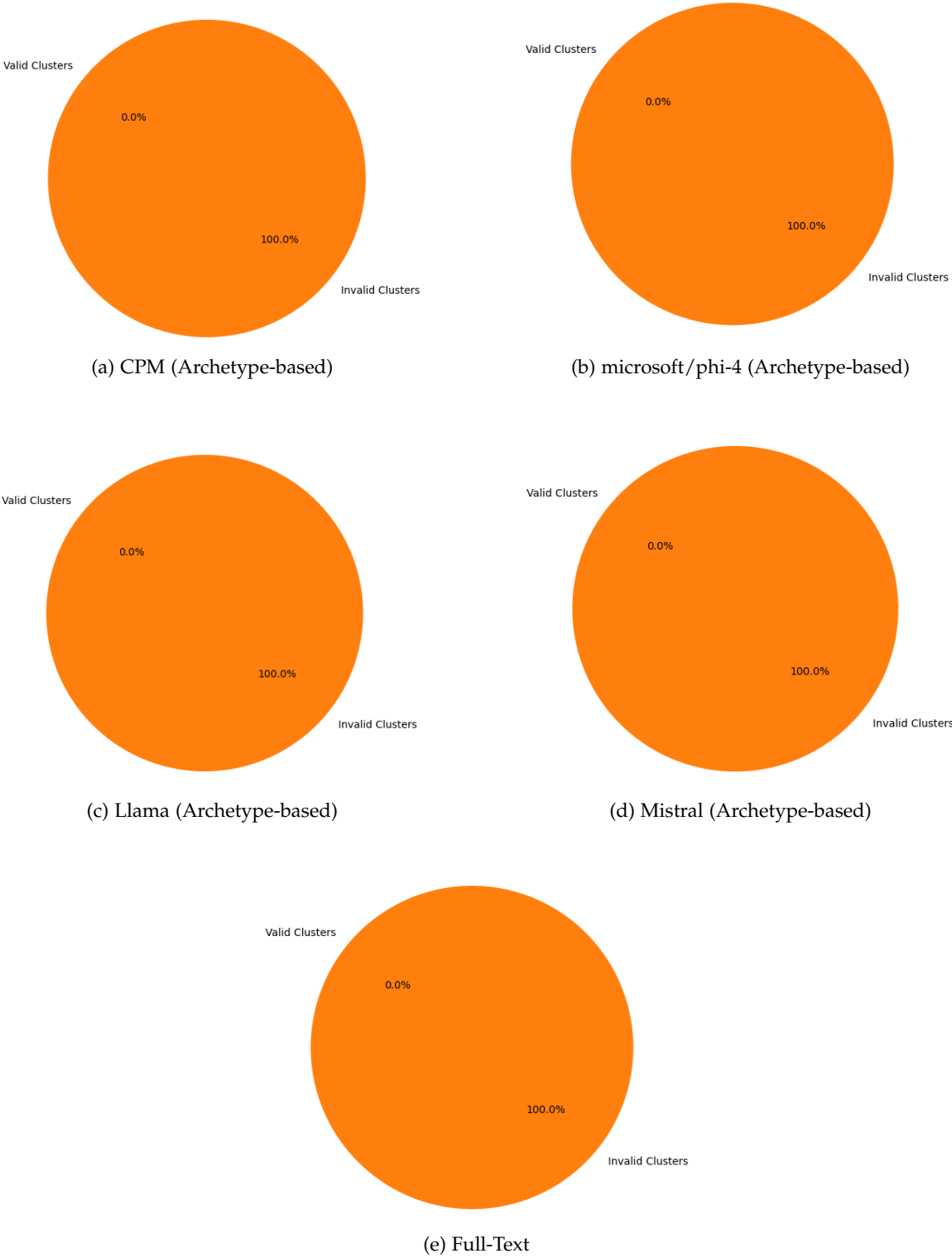
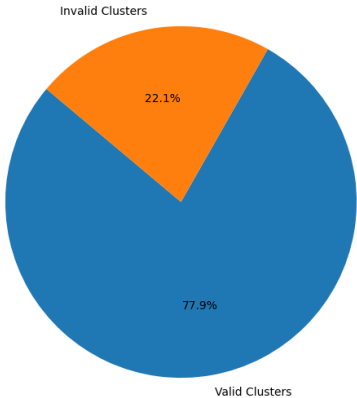
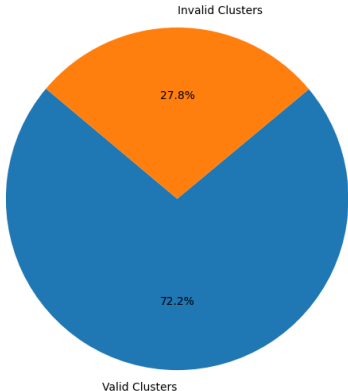


Figure 5.3: AG Dataset: K-Means (Archetype-based + Full-Text Clustering)

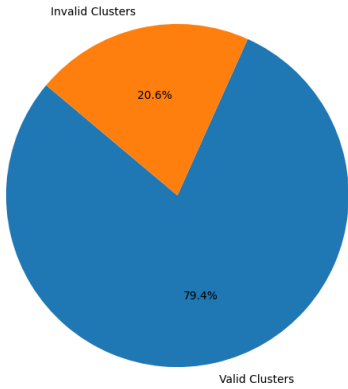
5.7. DOMAINS' DISJOINTNESS COMPARISON: ARCHETYPES VS. FULL-TEXT CLUSTERING



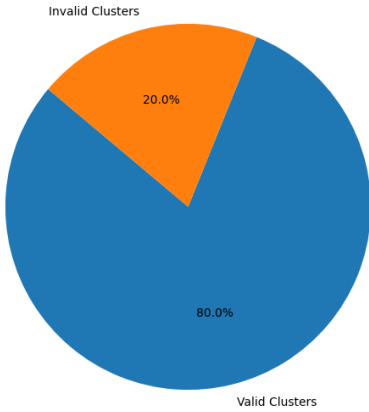
(a) CPM (Archetype-based)



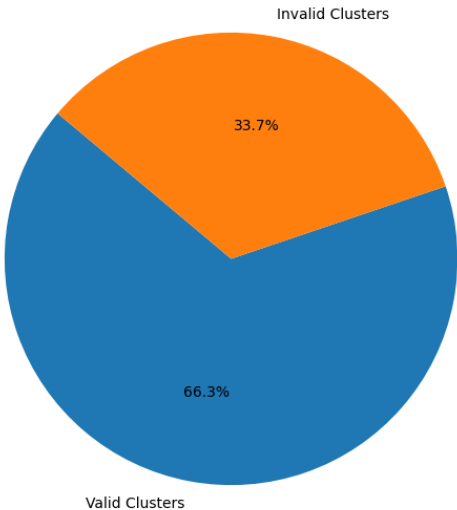
(b) microsoft/phi-4 (Archetype-based)



(c) Llama (Archetype-based)



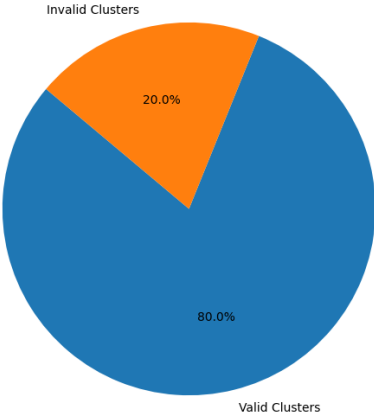
(d) Mistral (Archetype-based)



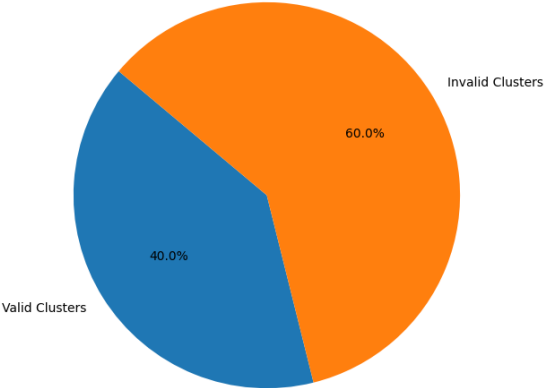
(e) Full-Text

Figure 5.4: AG Dataset: Recursive Hierarchical (Archetype-based + Full-Text Clustering)

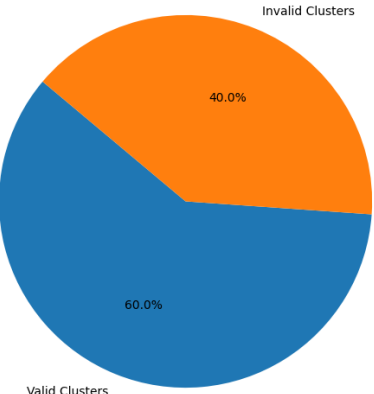
5.7. DOMAINS' DISJOINTNESS COMPARISON: ARCHETYPES VS. FULL-TEXT CLUSTERING



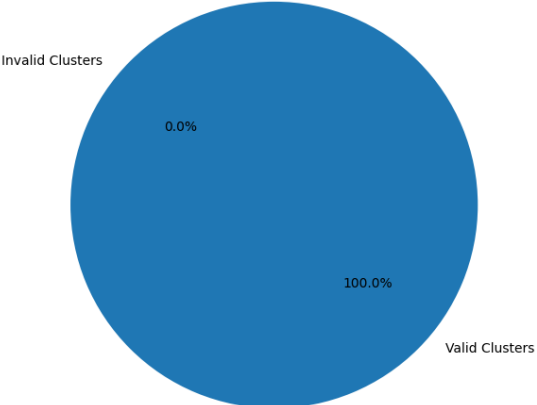
(a) CPM (Archetype-based)



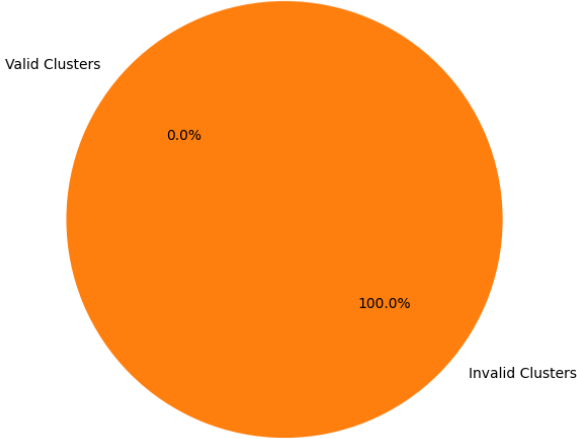
(b) microsoft/phi-4 (Archetype-based)



(c) Llama (Archetype-based)



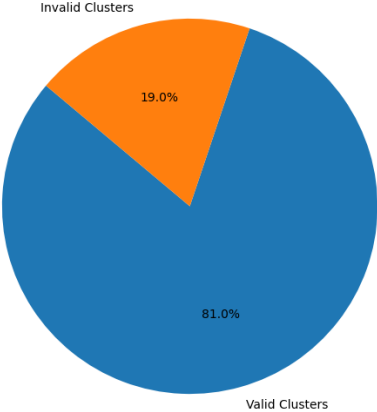
(d) Mistral (Archetype-based)



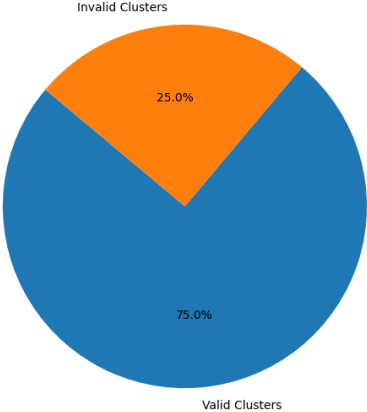
(e) Full-Text

Figure 5.5: BBC Dataset: K-Means (Archetype-based + Full-Text Clustering)

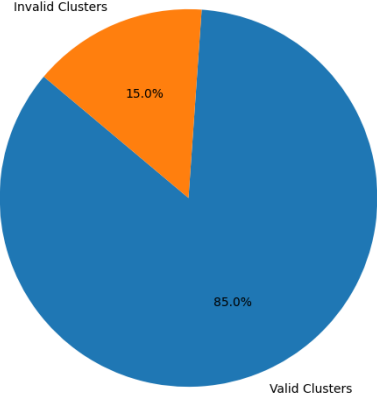
5.7. DOMAINS' DISJOINTNESS COMPARISON: ARCHETYPES VS. FULL-TEXT CLUSTERING



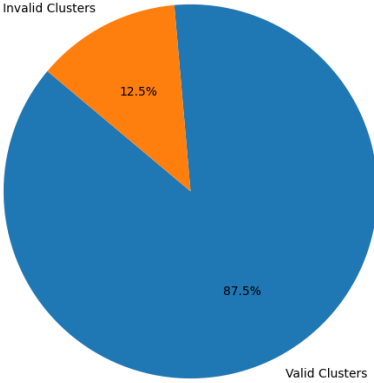
(a) CPM (Archetype-based)



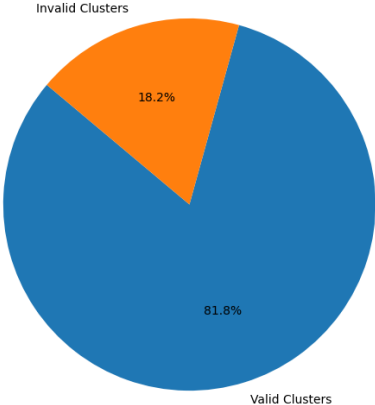
(b) microsoft/phi-4 (Archetype-based)



(c) Llama (Archetype-based)



(d) Mistral (Archetype-based)



(e) Full-Text

Figure 5.6: BBC Dataset: Recursive Hierarchical (Archetype-based + Full-Text Clustering)

6 Discussion

This chapter presents an overall discussion of the findings, focusing on the study implications, interpreting the results in area of the research objectives, the limitations of the study and proposes directions for future work.

6.1 Study Implications

The findings of this thesis reveal compelling implications for the future of NLP and unsupervised representation learning. At a high level, the results suggest that semantically meaningful, interpretable, and functionally useful text archetypes can be distilled from unlabeled data using LLMs with rivaling, and in some cases surpassing, that of traditional supervised pipelines. This serves as a crucial step toward **label-free NLP**, where high-level understanding and categorization of data become attainable without explicit human annotation.

The Bigger Picture: Toward Zero-Label NLP

While this study utilized benchmark labeled datasets for evaluation purposes, the core methodology of clustering and archetype distillation was agnostic to those labels. The labels were only used as ground truth references, not as inputs to the system. This raises a provocative question: *what if there were no labels at all?* The methods developed and evaluated here demonstrate that it is feasible to **infer structure, semantics, and even downstream task suitability** from purely unlabeled corpora. The ability to synthesize archetypes that cluster cleanly by topic and perform well in zero-shot or few-shot settings hints at a powerful paradigm: **unsupervised semantic structuring**.

A Paradigm Shift for Domain Adaptation and Corpus Exploration

In domains where annotation is prohibitively expensive (e.g., legal, biomedical, multilingual, or low-resource languages), this framework offers an efficient alternative. Instead of relying on labeled samples to bootstrap task-specific systems, one can generate domain-specific archetypes, organize them semantically via clustering, and then fine-tune small classifiers or search engines on top of these distilled representations. This drastically lowers the barrier for deploying NLP tools in new or specialized domains.

6.2 Results Interpretations

This section focuses on illustrating the results presented in the results chapter. It starts by analyzing the performance of various LLMs, highlighting key insights into their capabilities for semantic abstraction. This is followed by a comparison of classification outcomes using the original full-text data versus the generated archetypes, revealing their utility in downstream tasks. The discussion then explores the interpretability, domain relevance, and completeness of the archetypes, assessing how well they capture the semantic richness of the original full-text. Further, it examines the disjointness between domain texts and archetypes to determine whether the generated representations offer non-redundant and non-mixed domains insights.

6.2.1 Insights from LLMs Performance

The performance of the evaluated LLMs varied considerably in terms of reliability, consistency, and ability to produce well-structured JSON outputs. Models such as Gemma-2B, Phi-3.5-Mini-Instruct, and Qwen2.5-1.5B-Instruct frequently exhibited prompt and rule repetition, inconsistent output formatting, and general unreliability. These issues made them unsuitable for tasks requiring structured and strict adherence to the prompt specifications.

In contrast, Mistral-7B-v0.3, OpenBMB-MiniCPM-4B, Phi-4, and Meta-Llama-3-8B showed notably better results. These models generally adhered well to prompt constraints and produced more consistent outputs. Mistral-7B-v0.3, in particular, demonstrated strong capability by generating nearly all archetypes correctly. While Meta-Llama-3-8B used in its 8-bit quantized form showed promising performance, it occasionally failed to complete the full set of archetypes, likely due to formatting errors.

The DeepSeek-R1-Distill-Qwen-14B (4-bit quantized) and DeepSeek-R1-Distill-Llama-8B (8-bit quantized) models generated lengthy responses that often did not only focused on including the archetypal JSON objects, but also other reasonings. Thus, both tended to exceed token limits due to excessive reasoning and verbose outputs, resulting in incomplete or malformed JSON structures. These models also introduced noise such as Chinese characters and extra tokens that disrupted JSON syntax, reducing their effectiveness despite their advanced architectures.

Phi-4, in its 4-bit quantized form, showed high accuracy and consistency with minimal omissions. Similarly, OpenBMB-MiniCPM-4B, though unquantized, occasionally produced output with missing keys or unexpected language artifacts. Overall, while some of the more recent models showed stronger performance, increased model size did not consistently lead to better results. In fact, several smaller models such as OpenBMB-MiniCPM-4B, Meta-Llama-3-8B, and Mistral-7B-v0.3 outperformed their larger counterparts in reliability and output formatting. It is also worth noting that all models used in this study fall within the category of lightweight LLMs, selected for their deployability and efficiency.

6.2.2 Classification Performance: Domain Full Text vs. Generated Archetypes

This section compares the classification performance of models trained on domain full-text datasets with those trained on generated archetypes. The evaluation spans both AG News and BBC datasets, leveraging semantic search, accuracy, precision, recall, and F1-score metrics.

For the AG dataset as shown in heatmap Figures 6.2, the classifier trained on archetypes achieved competitive results across all metrics. In particular, its F1-scores and overall accuracy approached or even surpassed those of the full-text baseline on several classes. This suggests that despite their abstraction, the archetypes retained high discriminative power in downstream classification tasks. Similar trends were observed in the BBC dataset observed in heatmap Figures 6.1, where archetype-based training resulted in relatively higher classification scores, highlighting the effectiveness of distilled representations in maintaining semantic alignment with class labels, even though that was not for all datasets as AG dataset’s case. However, this could be because of the very small dataset size of BBC compared to the AG dataset.

The former also explains why fine-tuning the classifier with the BBC dataset for 1 epoch shows almost zero accuracies and other metrics in most cases. A deeper look reveals that the average number of archetypes generated for one domain in the BBC dataset is 69, whereas for the AG dataset it is 502. Thus, during 1 epoch training on the BBC dataset, the model has significantly fewer examples and semantic archetypes to learn meaningful patterns from, leading to underfitting. This scarcity not only limits the generalization capacity of the classifier but also reduces its exposure to the diverse semantic structures necessary for robust learning. In contrast, the AG dataset provides a richer and more varied archetypal landscape, allowing the model to quickly establish meaningful feature representations even with minimal fine-tuning.

In addition to classification tasks, semantic search evaluations provided deeper insight into the representational fidelity of the archetypes. Across both AG and BBC settings, archetypes generated by `mistral-7B`, `phi-4`, `miniCPM3-4B`, and `meta-llama-3-8B` achieved high semantic retrieval accuracy when evaluated on diverse datasets including short descriptions, BBC and AG, and domain-specific splits like only sport and 20 news groups datasets. The performance improvement when averaging over top-3 retrieved results (Top-3 accuracy) further demonstrated the robustness of archetypes in capturing class semantics. However, performance varied slightly across datasets. Full-text representations were marginally more effective on very short or ambiguous text segments, while archetypes excelled in settings requiring clarity and conceptual consistency. Notably, models such as `meta-llama-3-8B` and `miniCPM3-4B` consistently delivered top scores in Top-3 semantic retrieval, underscoring their generative precision and domain fidelity.

Overall, the results validate the potential of LLM-generated archetypes as viable and lightweight alternatives to full-text corpora for both classification and semantic retrieval tasks, specially `meta-llama-3-8B`. They demonstrate that, when crafted with high-quality prompts and appropriate LLMs, archetypes can encapsulate key class-level semantics and offer competitive performance across evaluation paradigms.

6.2. RESULTS INTERPRETATIONS

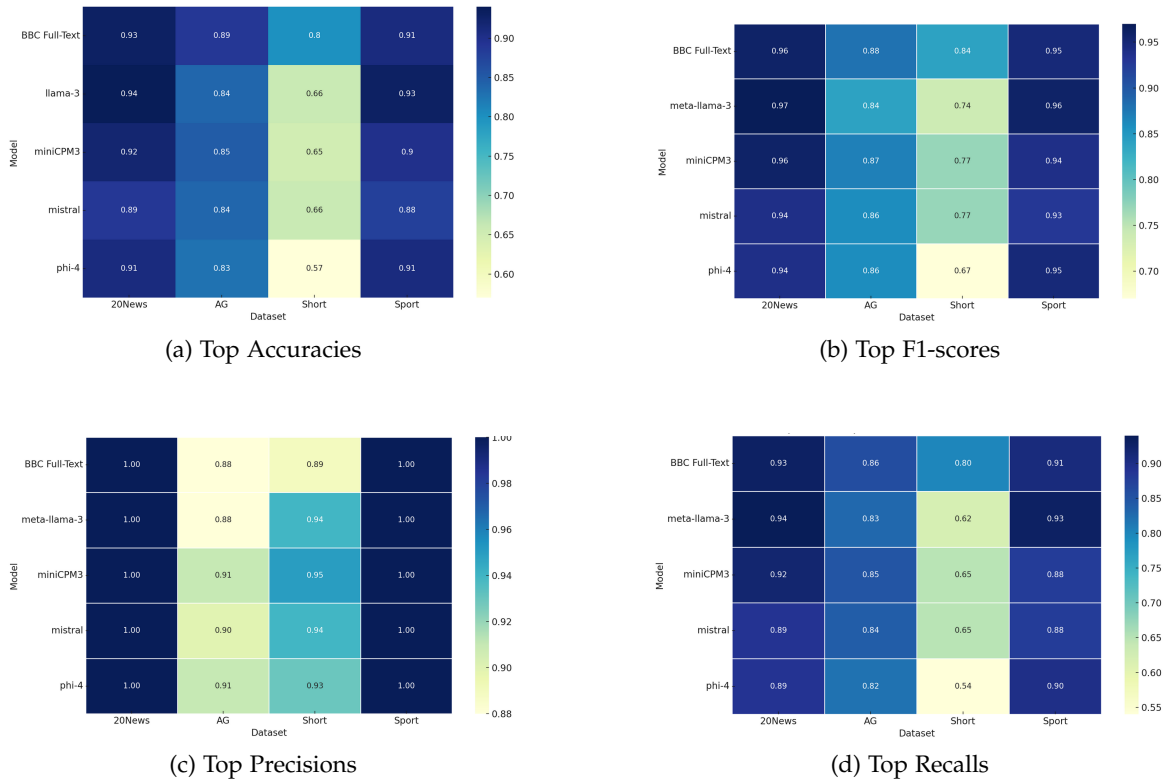


Figure 6.1: Heatmaps visualizing the classification performance of a RoBERTa classifier fine-tuned on BBC full-text and archetype representations across five target datasets. Each heatmap highlights the top values achieved for Accuracy, Recall, F1-Score, and Precision, regardless of the number of fine-tuning epochs.

6.2.3 Archetypes Interpretability, Domain Relevance, and Completeness

This section evaluates the generated archetypes based on human-centered criteria, specifically interpretability, domain relevance, and completeness. A multi-faceted survey was conducted involving non-domain experts, but participants who are good in filling AI-related surveys to qualitatively assess the outputs of various LLMs, including Mini-CPM, Llama-3.2, Mistral, and Phi-4.

Across all respondents, Phi-4 emerged as the most preferred model by a significant margin. In terms of overall model selection counts, Phi-4 accounted for approximately half of all choices across categories, with Mistral and Llama-3.2 following distantly, and Mini-CPM being least preferred. This strong preference for Phi-4 was consistent not only in general preference data but also across professional roles and disciplinary backgrounds.

Evaluation scores along three key dimensions Relevance, Coverage, and Comprehension further followed these trends. Phi-4 consistently achieved the highest mean ratings across all three dimensions, with particularly notable performance in Relevance (4.14), Coverage (4.01), and Comprehension (3.96). Other models such as Mistral and Llama-3.2 performed

6.2. RESULTS INTERPRETATIONS

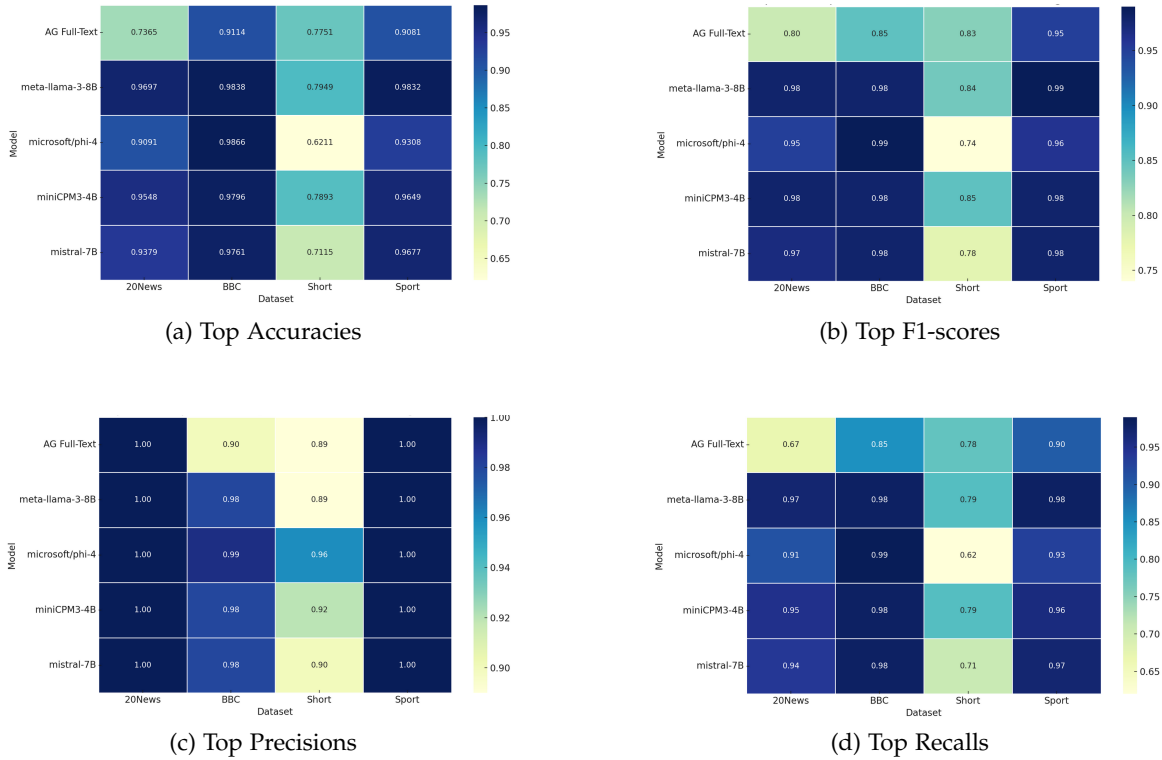


Figure 6.2: Heatmaps visualizing the classification performance of a RoBERTa classifier fine-tuned on AG full-text and archetype representations across five target datasets. Each heatmap highlights the top values achieved for Accuracy, Recall, F1-Score, and Precision, regardless of the number of fine-tuning epochs.

moderately well, but were outperformed by Phi-4 in all aspects. Notably, Mini-CPM scored the lowest, suggesting limited effectiveness in capturing domain-specific semantics and providing interpretable, complete archetypes.

It is important to note, however, that these subjective assessments do not necessarily align with earlier findings from the classification tasks. One likely reason for this divergence is that Phi-4’s archetype outputs were noticeably longer nearly double the length of those generated by other models. This increased verbosity may have contributed to the perception of higher coverage and comprehension among survey participants, as longer texts are often intuitively associated with completeness.

6.2.4 Analysis of Domain Full Text vs. Archetypes Disjointness

Following, classification results insights, clustering on archetypes particularly those generated by high-performing LLMs such as Phi-4, Llama-3.2, Mini-CPM, and Mistral produced substantially better domain disjointness. utilizing recursive hierarchical clustering, all four models consistently yielded large numbers of valid clusters across domains. For instance,

Phi-4’s archetypes for AG News were clustered into dozens of domain-pure groups for tech, sport, and business, with fewer mixed-domain clusters compared to full-text. BBC hierarchical clustering reflected similar improvements, especially in domains like entertainment and politics, which were otherwise more prone to semantic overlap in full-text clustering.

K-Means remained less effective for both raw texts and archetypes, suggesting that its underlying assumption of spherical clusters may not suit the nuanced semantics of either representation. However, even utilizing K-Means, LLM-based archetypes achieved valid clusters for certain domains (e.g., sport and politics in the BBC dataset), which was not achieved when on full-text.

6.3 Limitations

Despite the promising performance of archetype-based representations and their potential to enhance domain disjointness, this study is not without limitations. These limitations are important to contextualize the findings and inform future work.

Length Bias in Human Evaluation

One of the most notable limitations emerged during the human evaluation phase. Specifically, models like Phi-4 consistently received higher ratings across dimensions such as interpretability, comprehension, and completeness. However, this trend may partially stem from a length bias. Phi-4’s archetypes were significantly longer often double the average length of outputs from other models. As longer texts may intuitively appear more complete and detailed, participants may have been predisposed to equate verbosity with coverage and depth, even if not all added content was substantively beneficial. This possible conflation between quantity and quality introduces a subjective skew that may not accurately reflect the inherent semantic value of the outputs.

Inconsistencies Between Classification and Survey-Based Results

Another limitation lies in the misalignment between quantitative classification results and qualitative human preferences. While some models (e.g., Mini-CPM) demonstrated strong classification performance, their archetypes were not always rated highly by human evaluators. Conversely, Phi-4, which underperformed in certain classification scenarios, was the most preferred in subjective evaluations. This inconsistency suggests that performance metrics alone may not fully capture the qualities valued in human-centric use cases such as interpretation, domain understanding, or content summarization.

Restricted Evaluation Scope

The current evaluation is limited to two news classification datasets: AG News and BBC News. While these datasets offer a manageable benchmark for evaluating archetype disjointness and interpretability, they do not reflect the complexity of more nuanced domains (e.g., biomedical,

legal, or multilingual corpora). Furthermore, the domain labels in both datasets are relatively coarse-grained, which may mask finer-grained interpretive challenges that would arise in more sophisticated cases.

6.4 Future Work

To build upon the findings of this study and address its limitations, several directions for future research are proposed:

Expand Evaluation to Complex and Fine-Grained Domains

Future work should explore archetype extraction and clustering in more nuanced and high-stakes contexts, including:

- Biomedical, legal, or financial domains with highly specialized vocabulary.
- Multilingual corpora to test cross-lingual consistency and domain abstraction.
- Fine-grained multi-label settings, where documents span overlapping or hierarchical topic structures.

This would validate the generalizability of the proposed approach and identify domain-specific challenges.

Investigate Model-Specific Archetype Styles

Given the variations in archetype length and style across LLMs (e.g., Phi-4 vs. Mini-CPM), future research should investigate:

- How architectural or alignment differences shape summarization behavior.
- Whether archetype verbosity correlates with redundancy, or informativeness.
- How these properties influence human preference and downstream task utility.

This could inform future prompt design and model selection strategies for generating interpretable archetypes.

Together, these directions aim to create a more principled, fair, and scalable framework for evaluating and deploying domain-disjoint, complete, and human-interpretable archetypes.

7 Conclusion

This thesis introduced a promising framework for the distillation of semantically coherent and domain-representative archetypes from textual corpora by leveraging LLMs. The approach motivated by the growing need for domain interpretable, compact, and disjoint summaries demonstrated that archetype generation can facilitate the understanding of abstract domain structures while preserving semantic completeness. Through a multi-stage pipeline involving context windowing, semantic grouping via clustering, and archetypal generation with prompt-based LLM queries, this study operationalized a workflow capable of extracting interpretable representations across diverse topics.

The key contributions of this research are as follows:

- **Domain Archetypes Generation:** A pipeline is proposed to generate semantically disjoint archetypes per domain using hierarchical clustering and LLM-driven contextual distillation, offering an alternative to traditional full-text classification pipelines.
- **Model Evaluation via Dual Metrics:** A dual evaluation strategy is introduced combining quantitative classification-based assessments (semantic search, accuracy, F1, precision, recall) and qualitative human-centric dimensions (interpretability, domain relevance, completeness).
- **Human-Centric Evaluation Insights:** Through a survey involving domain experts and practitioners, insights about certain models were uncovered, particularly Phi-4, was consistently preferred in terms of comprehension and coverage though this preference was potentially skewed by text length.
- **Domain Disjointness Analysis:** Clustering techniques were experimented on both full-text and archetype representations, revealing that archetype-based clusters are significantly more domain-pure and interpretable, especially when using hierarchical clustering.
- **LLM Comparison and Behavior Analysis:** The thesis also documented varied behaviors across LLMs e.g., Phi-4 tended to generate longer outputs with more explicit labeling, while Llama-3 maintained coherence and stylistic conciseness highlighting architectural and alignment influences on output style.

This work contributes to the broader understanding of how LLMs can be repurposed for interpretable NLP, specifically in conditions where human readability and domain fidelity matter more than raw predictive power. By treating archetypes as disjoint, and complete semantic anchors, the focus is drawn towards structured symbolic representations that can

aid in explainability, content moderation, educational summarization, and domain-centric retrieval.

Despite the promising results, this study is not without limitations. Most notably, length bias in human evaluation, and the coarse granularity of the AG and BBC datasets could limit the full generalizability of the approach. Furthermore, archetype content, while disjoint at the domain level, may still vary subtly within sub-topics.

To strengthen and extend this line of inquiry, future work should explore controlled normalization of outputs to eliminate verbosity bias in human evaluation, and apply this framework in high-complexity or multilingual domains (e.g., biomedical, legal).

List of Figures

| | | |
|-----|---|----|
| 2.1 | NLP Market Forecasted Revenue [6] | 6 |
| 2.2 | LMs Development Over Past Years | 8 |
| 2.3 | LLMs Approach for Text Classification | 11 |
| 2.4 | Example Utilization of Clustering for User Preferences | 20 |
| 3.1 | Illustration of LLM Training on PCW vs. Single Context Window | 26 |
| 4.1 | Thesis Work Pipeline Illustration | 34 |
| 4.2 | Final LLM Prompt for Clusters' Archetypes Generation | 42 |
| 4.3 | Grouping Clusters per each LLM Call According to Model's Tokenizer limit | 43 |
| 4.4 | Example of a survey section showing the presentation of a text cluster with multiple AI-generated summaries | 50 |
| 4.5 | Qualified AI Taskers group on Prolific | 51 |
| 4.6 | Open-ended feedback section where participants provided suggestions for improving summary clarity and coverage | 52 |
| 4.7 | Demographic questionnaire used to collect information on participants' age, education, profession, domain, and technical skills | 53 |
| 4.8 | Embedded attention-check question instructing participants to select the middle answer to ensure attentiveness | 54 |
| 4.9 | Evaluation questions used to assess summary quality in terms of coverage, clarity, and relevance | 55 |
| 5.1 | LLMs' archetype preferences across overall participants, disciplines, and roles. | 72 |
| 5.2 | Averaged ratings for archetype interpretability, relevance, and completeness across different groups. | 73 |
| 5.3 | AG Dataset: K-Means (Archetype-based + Full-Text Clustering) | 74 |
| 5.4 | AG Dataset: Recursive Hierarchical (Archetype-based + Full-Text Clustering) | 75 |
| 5.5 | BBC Dataset: K-Means (Archetype-based + Full-Text Clustering) | 76 |
| 5.6 | BBC Dataset: Recursive Hierarchical (Archetype-based + Full-Text Clustering) | 77 |
| 6.1 | Heatmaps visualizing the classification performance of a RoBERTa classifier fine-tuned on BBC full-text and archetype representations across five target datasets. Each heatmap highlights the top values achieved for Accuracy, Recall, F1-Score, and Precision, regardless of the number of fine-tuning epochs. | 81 |

- 6.2 Heatmaps visualizing the classification performance of a RoBERTa classifier fine-tuned on AG full-text and archetype representations across five target datasets. Each heatmap highlights the top values achieved for Accuracy, Recall, F1-Score, and Precision, regardless of the number of fine-tuning epochs. . . . 82

List of Tables

| | | |
|-----|---|----|
| 4.1 | Overview of Employed Datasets | 37 |
| 4.2 | Generated Archetypes by LLMs for Game Evaluation Cluster Example | 45 |
| 5.1 | Datasets Details used for Evaluation and Training | 56 |
| 5.2 | BBC archetypes semantic search accuracy results across various LLMs and datasets. Top-3 indicates averaged score from top-3 retrieved results. Bold values indicate the highest score in each column. | 57 |
| 5.3 | BBC full-text semantic search accuracy across four datasets. Accuracy@1 is for the top retrieved result, while Accuracy@3 is averaged over the top-3 retrieved results. | 57 |
| 5.4 | AG full-text semantic search accuracy across four datasets. Accuracy@1 is for the top retrieved result, while Accuracy@3 is averaged over the top-3 retrieved results. | 57 |
| 5.5 | AG archetypes semantic search accuracy results across various LLMs and datasets. Top-3 indicates averaged score from top-3 retrieved results. Bold values indicate the highest score in each column. | 58 |
| 5.6 | Accuracy of text classifiers across different datasets. Bold values indicate the best accuracy per dataset. | 58 |
| 5.7 | Classification performance of BBC dataset’s archetypes vs. full-text on AG dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column. | 59 |
| 5.8 | Classification performance of BBC dataset’s archetypes vs. full-text on Only Sport dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column. | 60 |
| 5.9 | Classification performance of BBC dataset’s archetypes vs. full-text on short description dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column. | 61 |

| | | |
|------|---|----|
| 5.10 | Classification performance of BBC dataset’s archetypes vs. full-text on BBC dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column. | 62 |
| 5.11 | Classification performance of BBC dataset’s archetypes vs. full-text on 20 news groups dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column. . . . | 63 |
| 5.12 | Classification performance of AG dataset’s archetypes vs. full-text on AG dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column. | 64 |
| 5.13 | Classification performance of AG dataset’s archetypes vs. full-text on only sport dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column. | 65 |
| 5.14 | Classification performance of AG dataset’s archetypes vs. full-text on short descriptions dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column. . . . | 66 |
| 5.15 | Classification performance of AG dataset’s archetypes vs. full-text on BBC dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column. | 67 |
| 5.16 | Classification performance of AG dataset’s archetypes vs. full-text on 20 news groups dataset across 1, 3, and 5 epochs. Bold values indicate the highest score in each column per epoch. Underlined bold values indicate the overall highest score in each column. Bold models indicate the best model per epoch. Underlined bold models indicate the overall best model in each column. . . . | 68 |
| 5.17 | Training times (in seconds) for fine-tuning classifier on BBC full-text dataset & archetypes. Bold values indicate the lowest training time in each column. . . . | 69 |
| 5.18 | Training times (in seconds) for fine-tuning classifier on AG full-text dataset & archetypes. Bold values indicate the lowest training time in each column. . . . | 69 |
| 5.19 | GPU configuration used during experiments | 69 |

Bibliography

- [1] A. Iorliam and J. A. Ingio. “A comparative analysis of generative artificial intelligence tools for natural language processing”. In: *Journal of Computing Theories and Applications* 1.3 (2024), pp. 311–325.
- [2] M. M. Maestre, I. Martinez-Murillo, T. J. Martin, B. Navarro-Colorado, A. Ferrández, A. S. Cueto, and E. Lloret. “Beyond generative artificial intelligence: Roadmap for natural language generation”. In: *arXiv preprint arXiv:2407.10554* (2024).
- [3] Y. Chai, L. Jin, S. Feng, and Z. Xin. “Evolution and advancements in deep learning models for natural language processing”. In: *Applied and Computational Engineering* 77 (2024), pp. 144–149.
- [4] B. Yadav. “Generative AI in the Era of Transformers: Revolutionizing Natural Language Processing with LLMs”. In: *J. Image Process. Intell. Remote Sens* 4.2 (2024), pp. 54–61.
- [5] C. S. Kulkarni. “The evolution of large language models in natural language understanding”. In: *Journal of Artificial Intelligence, Machine Learning and Data Science* (2023).
- [6] artsmart.ai. *Natural Language Processing (NLP) Statistics for 2024 and Beyond*. Accessed: 2025-03-23. 2024. URL: <https://artsmart.ai/blog/natural-language-processing-nlp-statistics-2024/>.
- [7] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, and L. He. “A survey on text classification: From traditional to deep learning”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 13.2 (2022), pp. 1–41.
- [8] Y. Arslan, K. Allix, L. Veiber, C. Lothritz, T. F. Bissyandé, J. Klein, and A. Goujon. “A comparison of pre-trained language models for multi-class text classification in the financial domain”. In: *Companion proceedings of the web conference 2021*. 2021, pp. 260–268.
- [9] M. Osnabrügge, E. Ash, and M. Morelli. “Cross-domain topic classification for political texts”. In: *Political Analysis* 31.1 (2023), pp. 59–80.
- [10] T. Li, X. Chen, Z. Dong, W. Yu, Y. Yan, K. Keutzer, and S. Zhang. “Domain-adaptive text classification with structured knowledge from unlabeled data”. In: *arXiv preprint arXiv:2206.09591* (2022).
- [11] Z. Wang, Z. Chu, T. V. Doan, S. Ni, M. Yang, and W. Zhang. “History, development, and principles of large language models: an introductory survey”. In: *AI and Ethics* (2024), pp. 1–17.
- [12] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang. “A survey on large language model (llm) security and privacy: The good, the bad, and the ugly”. In: *High-Confidence Computing* (2024), p. 100211.

- [13] J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han. "Awq: Activation-aware weight quantization for on-device llm compression and acceleration". In: *Proceedings of Machine Learning and Systems 6* (2024), pp. 87–100.
- [14] M. van Baalen, A. Kuzmin, M. Nagel, P. Couperus, C. Bastoul, E. Mahurin, T. Blankevoort, and P. N. Whatmough. "GPTVQ: The Blessing of Dimensionality for LLM Quantization". In: *CoRR* (2024).
- [15] K. Egashira, M. Vero, R. Staab, J. He, and M. Vechev. "Exploiting LLM Quantization". In: *NeurIPS 2024*. 2024.
- [16] J. Fields, K. Chovanec, and P. Madiraju. "A survey of text classification with transformers: How wide? how large? how long? how accurate? how expensive? how safe?" In: *IEEE Access 12* (2024), pp. 6518–6531.
- [17] Y. Chae and T. Davidson. "Large language models for text classification: From zero-shot learning to fine-tuning". In: *Open Science Foundation 10* (2023).
- [18] M. J. J. Bucher and M. Martini. "Fine-Tuned'Small'LLMs (Still) Significantly Outperform Zero-Shot Generative AI Models in Text Classification". In: *arXiv preprint arXiv:2406.08660* (2024).
- [19] Y. Zhang, M. Wang, C. Ren, Q. Li, P. Tiwari, B. Wang, and J. Qin. "Pushing the limit of LLM capacity for text classification". In: *arXiv preprint arXiv:2402.07470* (2024).
- [20] Y. Zhang, R. Yang, X. Xu, R. Li, J. Xiao, J. Shen, and J. Han. "TELEClass: Taxonomy Enrichment and LLM-Enhanced Hierarchical Text Classification with Minimal Supervision". In: *THE WEB CONFERENCE 2025*. 2025.
- [21] J. Park and S. Choo. "Generative AI prompt engineering for educators: Practical strategies". In: *Journal of Special Education Technology* (2024), p. 01626434241298954.
- [22] A. Bozkurt. *Tell me your prompts and I will make them true: The alchemy of prompt engineering and generative AI*. 2024.
- [23] A. Bozkurt and R. C. Sharma. "Generative AI and prompt engineering: The art of whispering to let the genie out of the algorithmic world". In: *Asian Journal of Distance Education 18.2* (2023), pp. i–vii.
- [24] D. Park, G.-t. An, C. Kamyod, and C. G. Kim. "A Study on Performance Improvement of Prompt Engineering for Generative AI with a Large Language Model". In: *Journal of Web Engineering 22.8* (2023), pp. 1187–1206.
- [25] N. Moratanch and S. Chitrakala. "A survey on extractive text summarization". In: *2017 international conference on computer, communication and signal processing (ICCCSP)*. IEEE, 2017, pp. 1–6.
- [26] R. Jia, Y. Cao, H. Shi, F. Fang, Y. Liu, and J. Tan. "Distilsum: Distilling the knowledge for extractive summarization". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020, pp. 2069–2072.

- [27] J. Du and Y. Gao. "Domain Adaptation and Summary Distillation for Unsupervised Query Focused Summarization". In: *IEEE Transactions on Knowledge and Data Engineering* (2023).
- [28] L. Ragazzi, G. Moro, L. Valgimigli, and R. Fiorani. "Cross-Document Distillation via Graph-based Summarization of Extracted Essential Knowledge". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2024).
- [29] Y. Liu, Y. Yang, and X. Chen. "Improving Long Text Understanding with Knowledge Distilled from Summarization Model". In: *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2024, pp. 11776–11780.
- [30] A. M. Mansourian, R. Ahmadi, M. Ghafouri, A. M. Babaei, E. B. Golezani, Z. Y. Ghamchi, V. Ramezani, A. Taherian, K. Dinashi, A. Miri, et al. "A Comprehensive Survey on Knowledge Distillation". In: *arXiv preprint arXiv:2503.12067* (2025).
- [31] K. Acharya, A. Velasquez, and H. H. Song. "A survey on symbolic knowledge distillation of large language models". In: *IEEE Transactions on Artificial Intelligence* (2024).
- [32] C. Yang, Y. Zhu, W. Lu, Y. Wang, Q. Chen, C. Gao, B. Yan, and Y. Chen. "Survey on knowledge distillation for large language models: methods, evaluation, and application". In: *ACM Transactions on Intelligent Systems and Technology* (2024).
- [33] Z. Abbasi-Moud, H. Vahdat-Nejad, and J. Sadri. "Tourism recommendation system based on semantic clustering and sentiment analysis". In: *Expert Systems with Applications* 167 (2021), p. 114324.
- [34] M. Zubair, M. A. Iqbal, A. Shil, M. Chowdhury, M. A. Moni, and I. H. Sarker. "An improved K-means clustering algorithm towards an efficient data-driven modeling". In: *Annals of Data Science* (2022), pp. 1–20.
- [35] S. Sharma, N. Batra, et al. "Comparative study of single linkage, complete linkage, and ward method of agglomerative clustering". In: *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE. 2019, pp. 568–573.
- [36] K.-Y. Chen, S.-H. Liu, B. Chen, and H.-M. Wang. "An information distillation framework for extractive summarization". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26.1 (2017), pp. 161–170.
- [37] D. S. Asudani, N. K. Nagwani, and P. Singh. "Impact of word embedding models on text analytics in deep learning environment: a review". In: *Artificial intelligence review* 56.9 (2023), pp. 10345–10425.
- [38] Q. Liu, M. J. Kusner, and P. Blunsom. "A survey on contextual embeddings". In: *arXiv preprint arXiv:2003.07278* (2020).
- [39] R. Patil, S. Boit, V. Gudivada, and J. Nandigam. "A survey of text representation and embedding techniques in nlp". In: *IEEE Access* 11 (2023), pp. 36120–36146.
- [40] A. Celikyilmaz, E. Clark, and J. Gao. "Evaluation of text generation: A survey". In: *arXiv preprint arXiv:2006.14799* (2020).

- [41] A. Conneau and D. Kiela. “SentEval: An Evaluation Toolkit for Universal Sentence Representations”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.
- [42] C. Van Der Lee, A. Gatt, E. Van Miltenburg, S. Wubben, and E. Krahmer. “Best practices for the human evaluation of automatically generated text”. In: *Proceedings of the 12th International Conference on Natural Language Generation*. 2019, pp. 355–368.
- [43] J. Jung, X. Lu, L. Jiang, F. Brahma, P. West, P. W. Koh, and Y. Choi. “Information-Theoretic Distillation for Reference-less Summarization”. In: *CoRR* (2024).
- [44] E. Clark, T. August, S. Serrano, N. Haduong, S. Gururangan, and N. A. Smith. “All That’s ‘Human’ Is Not Gold: Evaluating Human Evaluation of Generated Text”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2021, pp. 7282–7296.
- [45] D. Nguyen. “Comparing automatic and human evaluation of local explanations for text classification”. In: *16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. 2018, pp. 1069–1078.
- [46] T. Vuong, S. Andolina, G. Jacucci, and T. Ruotsalo. “Does more context help? Effects of context window and application source on retrieval performance”. In: *ACM Transactions on Information Systems (TOIS)* 40.2 (2021), pp. 1–40.
- [47] N. Ratner, Y. Levine, Y. Belinkov, O. Ram, I. Magar, O. Abend, E. Karpas, A. Shashua, K. Leyton-Brown, and Y. Shoham. “Parallel Context Windows for Large Language Models”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2023, pp. 6383–6402.
- [48] K. Yamada, R. Sasano, and K. Takeda. “Verb Sense Clustering using Contextualized Word Representations for Semantic Frame Induction”. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (2021).
- [49] Y. Du, H. Sun, and M. Abdollahi. “Toward deep multi-view document clustering using enhanced semantic embedding and consistent context semantics”. In: *Knowledge and Information Systems* (2024), pp. 1–28.
- [50] A. Seibicke. “Extracting Semantically Meaningful Context Windows around Class-Specific Keywords”. PhD thesis. Master’s Thesis. Technical University of Munich, 2023. url: [https ...](https://www.techconf.uni-muenchen.de/theses/2023/Seibicke.pdf), 2023.
- [51] J. L. Martinez-Rodriguez, A. Hogan, and I. Lopez-Arevalo. “Information extraction meets the semantic web: a survey”. In: *Semantic Web* 11.2 (2020), pp. 255–335.
- [52] M. P. Naik, H. B. Prajapati, and V. K. Dabhi. “A survey on semantic document clustering”. In: *2015 IEEE international conference on electrical, computer and communication technologies (ICECCT)*. IEEE. 2015, pp. 1–10.

- [53] Y. Zhang, H. Jin, D. Meng, J. Wang, and J. Tan. “A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods”. In: *arXiv preprint arXiv:2403.02901* (2024).
- [54] H. Zhang, P. S. Yu, and J. Zhang. “A systematic survey of text summarization: From statistical methods to large language models”. In: *arXiv preprint arXiv:2406.11289* (2024).
- [55] Z. Kolagar and A. Zarcone. “Aligning uncertainty: Leveraging llms to analyze uncertainty transfer in text summarization”. In: *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertainNLP 2024)*. 2024, pp. 41–61.
- [56] A. Mullick, S. Bose, R. Saha, A. K. Bhowmick, A. Vempaty, P. Goyal, N. Ganguly, P. Dey, and R. Kokku. “Leveraging the power of llms: A fine-tuning approach for high-quality aspect-based summarization”. In: *arXiv preprint arXiv:2408.02584* (2024).
- [57] J. Fang, C.-T. Liu, J. Kim, Y. Bhedaru, E. Liu, N. Singh, N. Lipka, P. Mathur, N. K. Ahmed, F. Dernoncourt, et al. “Multi-LLM Text Summarization”. In: *arXiv preprint arXiv:2412.15487* (2024).
- [58] S. Shah, D. Chandrasekaran, S. Ryali, and R. Venkatesh. “Topic driven text summarization with defragmentation using llms”. In: *Authorea Preprints* (2025).
- [59] U. Jain, P. Mishra, A. Dash, and A. Pandey. “Multi-label multi-class text classification-enhanced attention in transformers with knowledge distillation”. In: *Journal of Applied Research and Technology* 23.1 (2025), pp. 82–93.
- [60] Y. Meng, Y. Zhang, J. Huang, C. Xiong, H. Ji, C. Zhang, and J. Han. “Text Classification Using Label Names Only: A Language Model Self-Training Approach”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 9006–9017.
- [61] S. Meisenbacher, T. Schopf, W. Yan, P. Holl, and F. Matthes. “An Improved Method for Class-specific Keyword Extraction: A Case Study in the German Business Registry”. In: *Proceedings of the 20th Conference on Natural Language Processing (KONVENS 2024)*. Ed. by P. H. Luz de Araujo, A. Baumann, D. Gromann, B. Krenn, B. Roth, and M. Wiegand. Vienna, Austria: Association for Computational Linguistics, Sept. 2024, pp. 159–165. URL: <https://aclanthology.org/2024.konvens-main.18/>.
- [62] M. Nakhla. *Cluster-Based Corrective Filtering for Class Specific Keyword Sets*. Accessed: 2025-04-14. 2024. URL: <https://www.matthes.in.tum.de/pages/cpbttod3n9vz/Guided-Research-Maria-Nakhla>.
- [63] Z. Chen, C. Li, X. Xie, and P. Dube. “OnlySportsLM: Optimizing Sports-Domain Language Models with SOTA Performance under Billion Parameters”. In: *NeurIPS Efficient Natural Language and Speech Processing Workshop*. PMLR. 2024, pp. 596–610.
- [64] X. Zhang, J. Zhao, and Y. LeCun. “Character-level convolutional networks for text classification”. In: *Advances in neural information processing systems* 28 (2015).
- [65] B. Bose. *BBC News Classification*. <https://kaggle.com/competitions/learn-ai-bbc>. Kaggle. 2019.

- [66] K. Crawford. *20 Newsgroups*. <https://www.kaggle.com/datasets/crawford/20-newsgroups>. Accessed: 2025-03-29. 2019.
- [67] R. Misra. *News Category Dataset*. <https://www.kaggle.com/datasets/rmisra/news-category-dataset>. Accessed: 2025-03-29. 2018.
- [68] S. Sturua, I. Mohr, M. K. Akram, M. Günther, B. Wang, M. Krimmel, F. Wang, G. Mastrapas, A. Koukounas, A. Koukounas, N. Wang, and H. Xiao. *jina-embeddings-v3: Multilingual Embeddings With Task LoRA*. 2024. arXiv: 2409.10173 [cs.CL]. URL: <https://arxiv.org/abs/2409.10173>.
- [69] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, et al. "Qwen2 Technical Report". In: *CoRR* (2024).
- [70] S. Hu, Y. Tu, X. Han, C. He, G. Cui, X. Long, Z. Zheng, Y. Fang, Y. Huang, W. Zhao, et al. "MiniCPM: Unveiling the Potential of Small Language Models with Scalable Training Strategies". In: *CoRR* (2024).
- [71] AI@Meta. "Llama 3 Model Card". In: (2024). URL: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.
- [72] DeepSeek-AI. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. 2025. arXiv: 2501.12948 [cs.CL]. URL: <https://arxiv.org/abs/2501.12948>.