Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

FAKULTÄT
FÜR MATHEMATIK, INFORMATIK
UND NATURWISSENSCHAFTEN

Language
Technology

BACHELORTHESIS

Surveying Scientific Literature with LLMs

Liam Debus

Field of Study: Software-System-Entwicklung
Matriculation No.: 7514704
1st Examiner: Prof. Dr. Chris Biemann, Universität Hamburg
2nd Examiner: Hans Ole Hatzel, Universität Hamburg

Language Technology
Department of Informatics
Faculty of Mathematics, Informatics and Natural Sciences

Universität Hamburg
Hamburg, Germany

A thesis submitted for the degree of

Bachelor of Science (B. Sc.)

Surveying Scientific Literature with LLMs

Bachelor's Thesis submitted by: Liam Debus

Date of Submission: 22.08.25

Supervisor(s):
Hans Ole Hatzel, Universität Hamburg

Committee:
1st Examiner: Prof. Dr. Chris Biemann, Universität Hamburg
2nd Examiner: Hans Ole Hatzel, Universität Hamburg

Universität Hamburg, Hamburg, Germany
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics

Language Technology

Abstract

With the ever-growing number of research papers published each day, documenting advancements across all scientific fields through so-called *survey papers* has become an essential part of the scientific process. These papers provide an overview of the current state of research in a given field and serve as valuable resources for researchers and learners alike. However, creating such a paper requires authors to sift through hundreds of publications to determine which should be included, an effort that is both time-consuming and labor-intensive.

This thesis poses the question: "Are large language models (LLMs) capable of aiding in the creation of high-quality scientific survey papers?" While extracting information from text is not a new application for LLMs, their potential to streamline the survey paper creation process has yet to be fully explored. Here, we investigate the plausibility of using different LLMs, such as GPT and LLaMA, to classify and retrieve information from papers according to our specifications.

We begin by recreating the selection process from the survey paper "Machine Learning in Computational Literary Studies" by **Hatzel et al. (2023)**, testing how well various models perform in analyzing abstracts to determine whether a paper employs machine learning methods and whether or not its main focus is computational literary studies. We then extend our experiments to unseen data, retrieving more fine-grained information from recent publications of the same conferences considered in the original study, that can be used for constructing tables in survey papers. Finally, we test the generalization capabilities of our approach by adapting prompts and applying the same methodology in a different scientific domain.

Contents

Appendices

1
Introduction

In 2024, at least 17,000 articles were published each month in the arXiv.org open-access scholarly article archive.[1] For each of these articles, the authors engaged in the rigorous scientific writing process, which requires them to gather their sources and references, develop a clear outline, acknowledge related work, conduct their research, analyze results, and present their findings after multiple rounds of reviewing and revising their work.

This is a lengthy process, yet an important one. It is undeniable that the mention of prior work is important for every robust scientific paper; yet it is not uncommon for authors to defer this part of the process until they have reached the later stages of their writing. However, as reinforced by **Doroudi (2023)** gathering prior research, "is an activity that leads to new insights into the research problem, generates new ideas, and alters the course of the research." Doroudi highlights a case where searching for previous related work had become the research process itself, since the link of various pieces of literature generated new undiscovered public knowledge in **Swanson (1986)**. This amplifies the importance of engaging deeply with prior literature throughout the research process, not merely as a final step.

This is even more so the case when looking at a specific type of scientific papers, named survey papers. The primary goal of these papers is to highlight and summarize existing research within an area of study. Survey papers provide researchers with an organized overview of a field, allowing them to understand the current landscape without manually reviewing hundreds of individual studies. Although this approach is invaluable for navigating vast amounts of information, it places a significant burden on survey authors, who must themselves engage in these exhaustive literature reviews.

To help alleviate the workload going into this process, we aim to assess if the employment of LLMs would not help authors work more efficiently, thus asking the question "Are LLMs capable of aiding in the creation of high-quality scientific survey

[1] "arXiv Monthly Submissions," 2025.

papers?"

This question presents the main focus of this thesis, with other sub-questions being asked and answered along the way. These include:

(1): "How good are the information extraction qualities of different LLMs?"
(2): "Which prompting-based model performs the best in this context?"
(3): "Which prompting strategy shows the most potential?"
(4): "Does an increase in context help our best-performing model improve?"

To address these questions, we have structured this thesis as follows. First, we contextualize our work in the background and provide information on the models and methods used. This is followed by a listing of related works that ask similar research questions to us. We then introduce the datasets we worked with during our experiments. Afterwards, we continue with presenting our experimental approach as well as the results. The experimental part of this paper is split into two distinct chapters. The first of which provides quantitative data to establish a baseline and assess performance, whereas the second chapter of the two aims to offer a more exploratory approach, where we judge both qualitatively and quantitatively how well our best-performing model generalizes and performs on more nuanced tasks. We round out the thesis by providing a conclusion where we look back on our experiments and contextualize them within the context of all our sub-questions, as well as the overarching main research question.

2
Background

2.1 Background

In this chapter we aim to provide the reader with background context that is needed to comprehend the content presented in this thesis. We start with an introduction to NLP and LLMs and then move forward with prominent LLM models that were used in this thesis. We end by presenting different prompting strategies that were utilized for our prompt-based models.

2.1.1 NLP and LLMs

The field of Natural Language Processing (NLP) has been growing rapidly over the past few years, especially driven by the rise of large language models (LLMs). **Monasterio Astobiza (2025)** defines LLMs as "a type of artificial intelligence (AI) that uses large amounts of text data to generate human-like responses to questions and instructions". A bibliometric review conducted by **Fan et al. (2024)** of research on LLMs from 2017 to 2023 shows a steady increase in publications, rising from 19 in 2017 to 392 in 2019 and reaching 2,101 in 2022. Furthermore, within this field, the largest theme of publications was "Algorithm and NLP Tasks," comprising 54% of all publications related to LLMs.

In addition to increasing popularity, LLMs have also been shown to perform well in NLP tasks. In a survey paper published by **Minaee et al. (2025)** the authors note that "LLMs have drawn a lot of attention due to their strong performance on a wide range of natural language tasks" and further present the benchmarks of different popular models. Among the tasks most relevant to this thesis, and the creation of high-quality survey papers, such as summarization, classification, and information extraction, LLMs have proven particularly effective.

To better understand why LLMs perform so well across tasks like summarization, classification, and information extraction, it is helpful to briefly outline how these models work and are trained. From a top-down perspective, LLMs are a type of neural network characterized by their large number of parameters, which allows them to perform a wide range of tasks and generate human-like text. LLMs work auto-regressively, meaning that they focus on predicting the next token, or set of tokens, in a sentence based on the previous sequence. This process, combined with the attention mechanism, enables the LLM to create a coherent and contextually appropriate output. Since the architectural details are not central to this thesis and are already well-documented in existing literature, we provide only a high-level overview here.

Instead, we focus on the training process of LLMs. As described earlier, the objective of an LLM is to predict the next token in a sequence. Achieving this with high accuracy requires training the model on a large corpus of text. Most models, including those used in this thesis, are pre-trained on general-purpose datasets sourced from the internet, such as Wikipedia articles or Stack Overflow posts. The goal of this pre-training phase is to equip the model with broad language comprehension capabilities. Once this phase is complete, the resulting model is referred to as a pre-trained model.

Following pre-training, a model can be fine-tuned on a specific dataset or task to improve its performance in that area. While fine-tuning has been shown to enhance task-specific performance, recent advances in model size and generalization capabilities have enabled some models to perform well on downstream tasks even without fine-tuning. Given the limited size of our dataset, which would likely lead to overfitting during fine-tuning, we restrict our experiments to using pre-trained models only.

With this general background in place, we now turn to a more detailed overview of the specific model families used in this thesis.

2.1.2 Prominent LLM Models

In order to properly assess how well the strong performance of LLMs in the field of NLP tasks translates into the creation of high-quality scientific survey papers, we test multiple prominent LLM models and compare their performance. To this extent, we provide some background knowledge about the different models that are referenced in this thesis.

LLaMA

The first model we used to assess the feasibility of this thesis was a LLaMA model. LLaMA (Large Language Model Meta AI) is a family of "foundation language models" developed by Meta.[1] It is well known for being trained on publicly available datasets and for its open access to the research community. The initial generation, LLaMA 1, was released in February 2023 in sizes of 7, 13, 33, and 65 billion parameters. A few months later, in July 2023, Meta released the LLaMA 2 models, available in sizes of 7, 13, and 70 billion parameters.[2] These models introduced architectural and performance improvements over the previous version and were made publicly available to “encourage

[1] Touvron, Lavril, et al., 2023.

[2] Touvron, Martin, et al., 2023.

responsible AI innovation." We used the 7B parameter LLaMA 2 model to explore the viability of our research setup in the early stages of this thesis.

Needless to say, since it was the first model we tested with, it was not as large or efficient as some of the ones presented later on. Although we did not test with the newer version, LLaMA has also released current competitive models, in the LLaMA3 and LLaMA4 series, which are competitive with current state-of-the-art systems.

LLaMA models follow a transformer-based architecture similar to that of GPT models, with a focus on next-token prediction. They introduce architectural optimizations that allow for high performance with relatively fewer parameters, making them attractive for research groups with limited computational resources.

DeepSeek

The DeepSeek family of models is developed by DeepSeekAI, a Chinese AI company, and focuses on open-weight research similar to the LLaMA family. These models are multilingual, designed to work well in both English and Chinese. DeepSeek offers a variety of models, some tailored for general instruction and reasoning tasks, while others are more specialized for math, coding, or even image processing. The architecture and training data are similar to those of the LLaMA family, as DeepSeek models are transformer-based and trained on open-source web data. The key difference lies in their multilingual capabilities, which are enabled by training on bilingual content and supported by a multilingual tokenizer.

The DeepSeek model tested in our experiments is the 7-billion-parameter variant from the DeepSeek LLM line, published in November 2023. This model focuses on instruction-following and has a larger 67-billion-parameter counterpart as well. This focus on instruction-following refers to the enhanced ability of the model to accurately respond to commands and prompts by the users. Instead of just generating text based on statistical patterns, these models aim to produce relevant outputs based on their interpretation of the instructions. According to **DeepSeek-AI et al. (2024)**, the 7-billion model outperforms LLaMA 2's 7-billion model, and similarly, the 67-billion DeepSeek model compares favorably to LLaMA 2's 70-billion model. We use the 7-billion model to evaluate whether this improved performance carries over to our specific task.

Gemma

In late 2023, Google released Gemini 1.0 as a competitor to OpenAI's GPT-4. Gemini came in three sizes, with the largest model achieving state-of-the-art performance benchmarks at the time.[3] While Gemini technology was integrated into select Google products, it was not publicly accessible.

[3] Google, 2023.

In February 2024, Google released Gemma, a publicly accessible line of LLMs developed during the same training process as Gemini.[4] The initial Gemma models included 2 billion and 7 billion parameter versions, representing some of the largest open-source LLMs available at that time. Despite this, they were generally outperformed by larger models such as GPT-4, primarily due to differences in scale.[5] Subsequent versions, Gemma2 and Gemma3, addressed this gap by increasing model size significantly and improving performance to near state-of-the-art levels.

The Gemma family uses the transformer-based, autoregressive architecture common to many contemporary LLMs, enabling it to generate coherent and contextually relevant text. Notably, Gemma3, the model used in this thesis, has 27 billion parameters and is fine-tuned for instruction-following tasks, making it well-suited for applications involving summarization and classification.

We selected Gemma3 for our experiments due to its strong balance of model size, instruction-following capabilities, and public availability, which allows us to benchmark its performance against other prominent LLMs such as LLaMA2 and DeepSeek.

OpenAI

OpenAI has constantly been at the forefront of state-of-the-art LLM development; whether it was with the initial release of GPT-1, creating the first auto-regressive model, the 2020 release of GPT-3, showing an immense development in parameter size and setting the standard for LLMs, or the release of ChatGPT in 2022 which made LLMs accessible to the general public, OpenAI has defined key milestones in the field. While other organizations have since released competitive, and sometimes superior, models, OpenAI continues to hold state-of-the-art rankings across various tasks.

In this thesis, we test two recent models from OpenAI: GPT-4o and o3. GPT-4o was released in May 2024 as an improved, multimodal variant of GPT-4 Turbo.[6] It delivers significantly faster and more cost-efficient outputs while maintaining text performance on par with its predecessor. GPT-4o also includes vision and audio comprehension capabilities, though we focus exclusively on text-based tasks in this work. As of May 28, 2025, GPT-4o ranks 28th in text performance on the community benchmarking site lmarena[7], and holds a LiveBench global average score of 53.95.[8]

The o3 model, released in April 2025 alongside o4-mini, is marketed by OpenAI as their “smartest and most capable” model to date.[9] The standout feature of this release is the reasoning capabilities that these models sprout. They are trained to "think and process for longer" before responding, thus increasing accuracy and quality. This increase in quality is also noticeable; when making use of the high reasoning, o3 scores

[4] Bouchard and Peters, 2024.

[5] Sapling, 2024.

[6] OpenAI, 2024.

[7] LMArena, 2025.

[8] LiveBench, 2024.

[9] OpenAI, 2025b.

first in the global average at LiveBench with a score of 80.71. Its reasoning average is also impressive at 93.33, being only second to Claude 4 Sonnet Thinking. On lmarena, o3 is currently ranked second in the text category, indicating strong community sentiment and performance.

Unlike the other models discussed in this thesis, OpenAI has not disclosed the parameter sizes or architectural details for GPT-4o or o3. Furthermore, these models are not open-source, and the weights are not publicly available. However, they are accessible through OpenAI's API and have been included in this thesis due to their state-of-the-art performance and widespread adoption. Consequently, our evaluation of these models is at times limited by the monetary constraints imposed by their API-based access.

2.1.3 Prompting Strategies

Throughout this thesis, we employ various prompting strategies to identify optimal prompts and assess which strategies are best suited to each model. In the following section, we shortly outline and expand on these strategies to provide a baseline understanding.

Zero-Shot

Zero-shot prompting is often seen as the most basic prompting approach. Simply put, it is defined as "when a model is asked to produce output without examples demonstrating the task".[10] Whether you ask the model to "tell me about your favorite color" or to produce a detailed output summarizing the latest news, without any other specifications, both of these prompts would be considered zero-shot. Zero-shot prompting is best employed when the prompt is centered around a context that the model already knows about, due to its large language base, since a zero-shot prompt does not provide any further information or examples on the subject. Zero-shot prompting is often used as a baseline to evaluate whether additional prompt engineering leads to improved model performance. In the context of this paper, zero-shot acts as the most basic, simple version of our prompt without further guidance.

Few-Shot

Few-shot prompting provides the model with a small number of examples alongside the prompt, leveraging in-context learning. This helps the model more reliably understand the desired output. The number of examples is not set and different tasks may find a different number of examples to be optimal. While generally more complex tasks would require more examples, if the number of examples is too large the model may struggle as the input token length gets larger and the instructions are a smaller part of the prompt. In those cases, it may be best to attempt a different prompting technique. In our thesis, for the classification tasks we employ 2-3 examples with both negative and positive

[10] Bouchard and Peters, 2024.

classifications.

Step-by-Step

Step-by-step prompting is a variant of chain-of-thought prompting, which involves guiding the model to break down tasks into smaller, logical steps. Unlike standard chain-of-thought prompting, we do not provide examples; instead, we include a direct instruction such as “let’s think step-by-step” to encourage structured reasoning. This helps keep the input token count low, which is especially important when working with smaller datasets where well-balanced examples are harder to create. As Bouchard et al. note, such prompting strategies are most effective with larger models capable of generating coherent reasoning, as “smaller models often produce nonsensical thought processes”.[11]

[11] Bouchard and Peters, 2024.

3
Related Work

3.1 Related Work

A growing body of research has investigated how Large Language Models (LLMs) can support the process of scientific writing. **Scherbakov et al. (2024)** provide an overview of the stages where LLMs are currently used, showing that most applications focus on the "Searching for Publications" step. This indicates that automation of literature retrieval is a primary motivation in the field. Their analysis compares classification models such as BERT with prompting-based models such as GPT, finding that while BERT achieved higher accuracy in title and abstract screening, GPT models performed better in data extraction tasks.

Similarly, **Tang et al. (2025)** ask the question "Are LLMs Good Literature Review Writers?" and examine whether models can identify the most relevant studies for a given research topic. Their prompting strategies influenced our own experimental design, as did the follow-up study by **Agarwal et al. (2025)**, who investigate whether LLMs can generate related work sections directly from paper abstracts. While they gather that LLMs have significant potential in this field, they are not there quite yet.

Beyond these larger surveys, **Agarwal et al. (2024)** introduce LitLLM, a toolkit designed to support literature review workflows, while **Joos et al. (2024)** evaluate how LLMs can be applied to filter studies more efficiently in systematic reviews. Together, these works demonstrate a growing interest in using LLMs for automating different components of the review-writing process.

Outside of complete LLM automation, a growing body of work also exists that focuses on human and LLM collaboration to accelerate task completion while still retaining good accuracy. **Wang et al. (2021)** present a survey outlining the research body in the Human-in-the-loop (HITL) NLP frameworks. Text classification is a staple in this field and Wang et al. mention how many HITL frameworks are developed for this problem, where they train a text classifier that is then improved by humans annotating data based on the current model behaviour. This can be quite effective as "with a relatively small set of human feedback, HITL can significantly improve the model accuracy."[1]

A related line of research asks how humans compare to LLMs in information extraction tasks. **Goh et al. (2020)** and **Dasigi et al. (2021)** both ask the question of whether humans or LLMs perform better in the task of information extraction. **Goh et al. (2020)** measure the performance of both parties when it comes to classifying research abstracts. In this specific task, the classification models do outperform humans by a decent amount. They note that, "The accuracy, measured by F1 score, of ML classifiers is 2–15 standard errors higher than that of human classifiers." **Dasigi et al. (2021)** on the other hand, focuses more on in-depth information retrieval. Both humans and the models are given academic research papers as well as questions, created by NLP practitioners who have only read the title and abstract of the corresponding paper. The results are as follows, "we find that existing models that do well on other QA tasks do not perform well on answering these questions, underperforming humans by at least 27 F1 points when answering them from entire papers, motivating further research in document-grounded, information-seeking QA."

In addition to general-purpose LLMs, domain-specific approaches have also been explored. In the medical field, **Zhou et al. (2021)** tested BERT models for article classification in systematic reviews. Their experiments showed that specialized models such as srBERT, pre-trained on abstracts and fine-tuned on article titles, significantly outperformed general-purpose BERT, highlighting the value of domain adaptation. This result motivated our inclusion of BERT baselines and informed our decision to consider previously fine-tuned models in our experiments.

Within the domain of Computational Literary Studies (CLS), existing infrastructure has been established that we build upon. **Hatzel et al. (2023)** provide a database of CLS publications across several conferences, which we recreated and extended as part of our dataset construction. Likewise, we made use of the database contained in Table 2 of **Sevgili et al. (2022)** to supplement our training data. These resources provided a foundation for evaluating LLM performance in our specific field of interest.

Beyond academia, large research organizations have begun releasing tools that integrate LLMs into scientific workflows. OpenAI's "Deep Research"[2] and AI2's "ScholarQA"[3] are designed to support users in searching and synthesizing scholarly literature. While promising, these tools are not free of limitations. Derek Lowe[4] reports that Deep Research can produce plausible yet outdated or misleading claims, such as citing older studies that have since been superseded. He remarks that while the output often appears solid, unless someone with subject expertise investigates the results, errors can easily slip through. Similarly, OpenAI's own technical report acknowledges that the o3 model, which underpins Deep Research, has a tendency to produce more claims overall. Leading to both more correct statements but also more hallucinations.[5] AI2's ScholarQA, which

[1] Smith et al., 2018.

[2] OpenAI, 2025a.

[3] Allen Institute for AI, 2025.

[4] Lowe, 2025.

[5] OpenAI, 2025c.

relies on a retrieval-augmented generation pipeline, reduces some of these risks by grounding statements in evidence first, but still lacks robust contradiction detection.

Overall, prior work shows that LLMs are increasingly being explored for tasks ranging from publication search and classification to related work drafting and data extraction. Domain-specific studies underscore the value of specialized models, while industry initiatives demonstrate the potential of integrating LLMs directly into research pipelines. However, much of this work emphasizes search and filtering, whereas fine-grained tasks such as structured information extraction and table construction remain less explored. Our thesis addresses this gap by evaluating the performance of LLMs in mostly CLS-specific datasets and by systematically testing their ability to extract, organize, and present information in ways directly useful for survey paper creation.

4
Datasets

4.1 Datasets

This chapter introduces the three datasets used in our experiments. To evaluate model performance, a diverse selection of scholarly papers was required to ensure a representative and challenging testbed. Each dataset is outlined in the following sections.

4.2 Original CLS Survey Paper Dataset

Our primary dataset is a modified version of the one introduced in the survey paper by **Hatzel et al. (2023)**. It includes all publications from the 2022 issue of the *Journal of Computational Literary Studies*[1], the 2021[2] and 2022[3] proceedings of the *SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, and the 2022 conference proceedings of the *Conference on Computational Humanities Research.*[4]

We excluded one set of papers from the original dataset, the findings from the 2022 *Digital Humanities* conference, as they lacked identifiable abstracts and were significantly shorter, averaging only about 300 tokens. After this filtering step, our final dataset consists of 77 papers.

Each paper in the dataset was manually annotated with binary labels for two categories: **Machine Learning (ML)** and **Computational Literary Studies (CLS)**. If a paper applied machine learning techniques, it received an ML score of 1; otherwise, 0. Similarly, if the main topic of the paper fell under computational literary studies, it received a CLS score of 1; otherwise, 0. This allowed us to establish a quantitative

[1] "Journal of Computational Literary Studies: Volume 1 - Issue 1 - 2022," 2022.

[2] SIGHUM, 2021.

[3] Degaetano et al., 2022.

[4] Karsdorp and Nielbo, 2022.

baseline for evaluation. Of the 77 papers, 48 were labeled as CLS (1), and 54 were labeled as ML (1).

All labels were manually assigned by a single annotator. Papers that received a label of 1 in both categories were already mentioned in the survey by **Hatzel et al. (2023)**, allowing for cross-verification of those annotations. To assign the correct labels, each abstract was read thoroughly, and when necessary, relevant parts of the full paper were consulted. All papers were preprocessed into a standardized CSV format, with abstracts extracted via PDF parsing scripts or manually in cases where automated extraction failed. Each row in the dataset contains a paper number (as key), the title, abstract, and binary CLS/ML labels.

In Chapter 5, we use this dataset to evaluate model performance by providing each model with the abstract and comparing its predictions to our ground-truth labels, reporting both accuracy and F1 scores. In Chapter 6, although we do not use the labels, we continue to utilize the abstracts (and, in some experiments, the full texts, which we extract via markdown parsing scripts after converting the PDFs) to explore qualitative aspects of model performance.

4.3 Updated CLS Survey Paper Dataset

The second dataset used in this thesis is an updated version of the dataset described in Section 4.2. It includes all papers from the following sources: the 2023[5] and 2024[6] issues of the *Journal of Computational Literary Studies*, the 2023[7] and 2024[8] editions of the *SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities, and Literature*, as well as the 2023[9] and 2024[10] proceedings of the *Conference on Computational Humanities Research.*

This updated dataset comprises 199 papers, making it more than twice the size of the original set. Due to the lack of prior survey-based inclusions (as we had for the previous dataset) and the scale of the data, we did not assign binary CLS and ML labels. Instead, this dataset is used exclusively for exploratory and qualitative analysis, as outlined in Chapter 6. As with the previous dataset, both the abstracts and full texts of the papers were extracted using PDF and Markdown parsing scripts, with manual correction applied where necessary.

[5] "Journal of Computational Literary Studies: Volume 2 - Issue 1 - 2023," 2023.

[6] "Journal of Computational Literary Studies: Volume 3 - Issue 1 - 2024," 2024.

[7] Degaetano-Ortlieb et al., 2023.

[8] Bizzoni et al., 2024.

[9] Sela et al., 2023.

[10] *Proceedings of the Computational Humanities Research Conference 2024, Aarhus, Denmark, December 4-6, 2024,* 2024.

4.4 Neural Entity Linking Survey Paper Dataset

The final dataset used in this thesis originates from a domain outside of Computational Literary Studies. The motivation behind its inclusion was to evaluate how well the techniques developed in earlier experiments generalize to a dataset unrelated to CLS. To this end, we constructed a dataset based on the papers listed in Table 2 of the survey "Neural Entity Linking: A Survey of Models Based on Deep Learning" by **Sevgili et al.** **(2022)**.

In that survey, the authors manually annotated the papers with metadata such as *encoder type* and *learning type for disambiguation.* This structure mirrors the Table 3 found in the CLS survey paper by **Hatzel et al.** **(2023)**, making it possible to use similar classification-based approaches as in our earlier experiments. As such, this dataset is employed in Chapter 6 to test the generalization ability of our prompting-based models in a new research domain.

5
Analyzing abstracts

5.1 Analyzing abstracts

To explore whether large language models (LLMs) are capable of aiding in the creation of high-quality scientific survey papers, we must first evaluate their ability to extract and classify relevant information from scientific texts. This chapter presents our first set of experiments, designed to quantitatively assess the performance of LLMs in processing scientific literature and identifying key thematic categories.

The overarching goal is to classify scientific texts based on their association with two categories: Computational Literary Studies (CLS) and Machine Learning (ML).

To achieve this, we use the first dataset introduced in Chapter 4, which contains scientific texts and their annotated scores for the CLS and ML categories. Using this data, we test various models by providing them with the abstracts of these texts and asking them to assign scores:

- A score of **1 in CLS** indicates the main theme of the text falls within Computational Literary Studies; a **0** indicates otherwise.
- A score of **1 in ML** indicates the use of machine learning techniques; a **0** indicates no use of such techniques.

5.1.1 Research questions

These experiments aim to answer the following research questions:

1. *How good are the information extraction qualities of different LLMs?*
2. *Which prompting-based model performs the best in this context?*
3. *Which prompting strategy shows the most potential?*

The first question will be revisited in Chapter 6, where we examine more fine-grained information extraction tasks. While we do provide some results toward answering this question in this chapter, the focus mainly lies on answering the second and third questions.

5.1.2 Approach

Initially, we planned to test both encoder-based models and prompting-based models. Early experiments, however, revealed that encoder-based models performed poorly even on simple classification tasks. As a result, we shifted our focus to prompting-based approaches, which allowed us to tailor instructions for each model and leverage their few-shot learning capabilities.

Two types of approaches were explored in this chapter:

- **Encoder-based classification models**: These are briefly discussed in Section 5.2, where we document their results for completeness.
- **Prompting-based models**: These are examined in detail in Section 5.3, where we describe our prompt design, model selection, and evaluation.

This structured exploration allows us to compare different models and prompting strategies in a systematic way, providing insight into their relative strengths and weaknesses.

5.2 Encoder-based models

Before exploring prompt-based methods, we first experimented with two encoder-based models of the BERT architecture to evaluate whether treating our problem as a straightforward classification task could yield satisfactory performance.

In the initial, simpler approach, we concatenated each paper's title with its abstract and created a train/test split (90% training, 10% testing). The input was tokenized and passed to a `BertForSequenceClassification` model with the number of labels set to 2 for binary classification (0 or 1). After training, we evaluated the model's performance on the test set.

The results revealed a significant limitation: the model **consistently predicted the positive class** (label 1) for all examples. This is likely due to the class imbalance in our dataset, where positive labels dominate. As a result, the model achieved a CLS accuracy of approximately 54.5% and an ML accuracy of 75.3%, which mirrors the proportion of positive and negative labels in the dataset. Although the accuracies appear better than random chance, these figures are misleading, as the model failed to identify any negative cases, leading to a **100% false positive rate**, an outcome that is completely unacceptable. Attempts to mitigate this issue, such as modifying hyperparameters like the learning rate or number of epochs, were unsuccessful. Consequently, we proceeded to test a different BERT model in hopes of overcoming these shortcomings.

The next model we tested was `SciBERT`, a variant of BERT trained on scientific text. Its training corpus comprises full papers sourced from Semantic Scholar. Since `SciBERT` is designed to handle scientific language, it appeared promising for our dataset. However, simply repeating the same classification process as before was unlikely to succeed. Instead, we implemented a *masked token approach*, in which `SciBERT` is provided with a sentence containing a mask token and predicts the most likely replacement based on the abstract's context.

We experimented with two different sentences, one for each classification task. For CLS classification, we used: *"Is the main topic of this paper computational literary studies related? The answer is [MASK]."* For ML classification, the sentence was: *"Does this paper apply machine learning techniques? The answer is [MASK]."* We then compared the predicted scores for `yes` and `no` as possible replacements for the mask token, assigning a binary label (0 or 1) depending on which had the higher score. The results were again unsatisfactory: this time, the model **consistently predicted the negative class** (label 0). This behavior likely stems from the fact that `SciBERT` was not designed for such masked question-answering tasks, and in scientific texts, a negative response (`no`) is more probable in these contexts.

From these two experiments, we conclude that encoder-based models pre-trained on large datasets are not well suited to our task. To improve their performance, we would need to fine-tune these models specifically for our classification problem. However, our dataset is both limited in size and nuanced in nature. As we observed with `BertForSequenceClassification`, using 90% of the dataset for training introduced a problematic pattern where the model consistently predicted the dominant class. Increasing the training dataset size might help the model capture more subtle distinctions, but it also raises the risk of overfitting and poor generalization on unseen data.

On the other hand, relying solely on a pre-trained model trained on scientific text (e.g., `SciBERT`) led to predictions that merely reflect corpus probabilities, for instance, favoring `"no"` over `"yes"` in our context. While encoder-based models have been used successfully for classification tasks in other work (e.g., **Zhou et al. (2021)**), they proved ineffective in our setting. For these reasons, we turned to prompt-based models instead, as they allow us to work with a small dataset and design prompts that communicate our task and expected outputs more directly.

5.3 Prompt-based models

Having concluded that simple classification approaches using encoder-based models were insufficient for our task, we turned to prompt-based methods instead. This approach allows us to clearly define the goal and provide the model with context beyond the immediate data.

Throughout our experiments, we tested a wide variety of models. Each exhibited its own quirks in response to prompting, and we observed that a strategy that worked well for one model often performed poorly on another. As a result, we found it necessary to develop tailored prompts for each model. However, for the sake of objective comparison, we also established a baseline prompt: this ensured that performance differences could not be attributed solely to variations in prompt design. For each model, we tested both the baseline and a more optimized prompt, and we present their comparative performance at the end of this chapter. We now introduce the baseline prompts for both CLS and ML classification tasks.

CLS Baseline Prompt

```
You are an expert in Computational Literary Studies (CLS).
**CLS Definition:**
CLS applies computational methods (e.g., text mining, stylometry, sentiment analysis) to literary texts (e.g., novels, poetry, drama). It excludes studies focused on historical records, cultural trends, or linguistic change unless literature is central.
Your task is to analyze the text below and determine if it falls under computational literary studies or not. Return only your analysis.
**Text:**
"{content}"
**Analysis:**
```

ML Baseline Prompt

```
You are an expert in Machine Learning (ML).
**ML Definition:**
ML focuses on the development of algorithms that improve automatically through experience. It includes methods such as supervised learning, reinforcement learning, and natural language processing. Simple rule-based algorithms and statistical evaluation are not machine learning.
Your task is to analyze the text below and determine if it mentions machine learning techniques or not. Return only your analysis.
**Text:**
"{content}"
**Analysis:**
```

5.3.1 LLaMA 2 Prompting

The first model we used to test our approach was from the LLaMA 2 family, specifically the 7B chat fine-tuned model.[1] Developed by Meta and released in July 2023, this variation represents the smallest size in the LLaMA 2 series, with larger models featuring 13B and 70B parameters. It is designed exclusively for text input and generation without image processing capabilities. Given its smaller size and performance relative to current state-of-the-art models, we did not expect exceptional results but rather insights into whether scaling up might yield significant improvements.

[1] Meta, 2023.

We first evaluated this model using the baseline prompt. As the initial model tested, and also the least capable in our set, it shaped the constraints for our baseline prompt. Unlike subsequent models, LLaMA 2 7B was **unable to consistently assign coherent values of 0 or 1** to our categories when directly prompted. Consequently, the baseline prompt was simplified to request only an analysis of the text. The inability to assign scores directly likely stems from the increased input length when requesting both an analysis and numerical scoring, effectively adding a sub-task to the model's workload. After obtaining the analyses, we manually reviewed the outputs and assigned scores to the categories based on the provided reasoning. The results of this process were as follows:

Table 5.1: Performance of using the LLaMA 2 7B model on the original CLS dataset with the baseline prompt

Category	Accuracy (%)	F1 Score
CLS	57.14	0.57
ML	68.83	0.77

Overall, these results shown in Table 5.1 are rather poor but not unexpected given the limitations of this smaller model. For CLS, performance was close to random chance, and although results in the ML category were somewhat better, they remained far from satisfactory.

We then attempted to optimize the prompt. However, this process was hampered by frequent issues where the model would repeat the prompt verbatim instead of generating a meaningful output. This behavior was likely due to the combined length of the prompt and abstract, which exceeded the model's capacity to handle longer inputs effectively. To address this, we set a token limit for the output using the `max_new_tokens` parameter to discourage repetition and further streamlined the prompt to reduce input length. Despite these adjustments, no significant improvements were observed. Consequently, the results obtained with the baseline prompt also represent the best performance achieved with this model.

Nevertheless, this experiment demonstrated that our overall approach is viable and capable of producing results. It suggests that employing a larger and more capable model could lead to substantial improvements.

5.3.2 DeepSeek Prompting

Before moving on to larger models, we first tested a model of comparable size that boasts better reported performance. This model, developed by DeepSeek, is a quantized version of the DeepSeek LLM 7B Chat model.[2] While the architectural details of this model have already been discussed in Chapter 2, it is worth emphasizing that the quantization played an important role in reducing memory requirements and making it feasible to run on the hardware available for our experiments. We selected this model because, like LLaMA, DeepSeek is widely adopted within the open-source community. It provides a

[2] Deepseek LLM 7B Chat - GPTQ (Hugging Face), 2025.

direct comparison to LLaMA 2, offering similar parameter size but reportedly superior performance on several benchmarks due to its more refined architecture. Here, we evaluate whether these claimed advantages translate to improved results for our specific information extraction task.

An interesting observation from this phase of testing is that the split between classifying texts for the CLS and ML categories using two separate prompts originated with the DeepSeek model. Unlike LLaMA, DeepSeek struggled to reliably handle classification for both categories within a single prompt. Often it would output only one score or analysis for a category, omitting the other entirely. To address this, we introduced two separate prompts, one targeting CLS classification and another for ML. This adjustment resolved the issue and, when retroactively applied to the LLaMA model, also improved its performance to the values reported in the previous subsection.

When we ran the baseline prompt for DeepSeek, following the same process as for the LLaMA model, obtaining an analysis and manually assigning scores, the results were as follows:

Table 5.2: Performance of using the Deepseek 7B model on the original CLS dataset with the baseline prompt

Category	Accuracy (%)	F1 Score
CLS	62.33%	0.73
ML	70.13%	0.76

While these results in Table 5.2 represent a clear improvement over the LLaMA model, they remain **somewhat underwhelming**. This is not unexpected, as the model is still relatively small at 7B parameters, though its architecture appears better optimized for our task than LLaMA's. These findings reinforce the idea that model size alone is not the sole determinant of performance; architectural refinements and training data also play a critical role.

Efforts to optimize the prompt revealed further challenges. Like LLaMA, DeepSeek struggled with the added complexity of assigning scores directly in the original prompt. In several cases, the model would correctly analyze a text, stating, for example, "This text uses machine learning techniques", yet assign an incorrect score of 0 for ML. This suggests that the scoring step introduces cognitive overhead that exceeds the model's effective context handling capabilities at this scale. To mitigate this, we concentrated on refining the analysis portion of the prompt and relied on an automated post-processing step to extract scores from the generated analysis. This separation of concerns proved more robust and highlights a practical strategy for working with smaller models.

In optimizing the prompt, we attempted to provide the model with additional context about what constitutes a positive or negative classification for both categories. While Deepseek was able to handle this increased context, unlike the LLaMA 2 model, it did not consistently lead to improved scores. In fact, the results for the CLS category did not improve during our testing and often performed slightly below the baseline by a few percentage points. In contrast, the scores for the ML category showed notable improvement, reaching an **accuracy of 70.13%** and an **F1 score of 0.76**.

Overall, while Deepseek outperformed LLaMA 2 in both categories at the baseline level, the improvement was modest. We attribute this to the relatively small size of both models and their limited capacity to handle more complex prompts that could have improved scoring. Nevertheless, this experiment provided the valuable insight that **improved model architecture alone can yield measurable gains**. As we progress to larger models with stronger performance, we hypothesize that increases in size alone will eventually reach a performance plateau. At that stage, further advancements will likely depend on architectural improvements rather than scaling. For now, we turn to larger models to observe the expected gains in performance.

5.3.3 Gemma3 Prompting

The first larger model we tested was the quantized version of the Gemma3 27B Instruct model. At 27 billion parameters, which is nearly four times larger than the previous 7B models, this model offers significantly more computational power. This difference was evident in testing, as the model demonstrated a much stronger grasp of the task compared to earlier models. We encountered fewer issues during both the baseline prompting and the optimization process. This overall improvement in performance is also reflected in the results of the baseline prompting, which we present below:

Table 5.3: Performance of using the Gemma3 27B model on the original CLS dataset with the baseline prompt

Category	Accuracy (%)	F1 Score
CLS	89.61%	0.90
ML	75.32%	0.82

Table 5.3 highlights a substantial improvement in the CLS category, where the increase in performance is particularly notable. While the ML results did not improve as dramatically, they still show a clear gain over the previous models. Interestingly, in contrast to the LLaMA and DeepSeek models, where the ML category consistently outperformed CLS, this trend is reversed here, with CLS now achieving higher scores. This also marks the largest disparity between the two categories observed so far. To explore whether this trend continues and to further improve results, we experimented with prompt optimization and alternative prompting strategies.

While the Gemma3 model is highly capable and able to handle analysis for both CLS and ML within a single prompt, we observed improved performance when splitting the prompts. Consequently, we chose to keep them separate. Including instructions for the model to score the analysis itself did not negatively impact results, so this component was retained in the prompts, as opposed to done through post-processing as we had with the previous two models.

In our initial attempts at optimizing the prompts, we provided more detailed definitions to help the model better understand the requirements for each category, while maintaining a zero-shot prompting style. The impact of this change was minor: accuracy and F1 scores for CLS remained almost identical, but ML scores improved slightly, reaching an accuracy of 79.22% and an F1 score of 0.84.

Next, to address our third research question, we tested alternative prompting techniques. We began with few-shot prompting, where the model was shown examples of both positive and negative classifications. To avoid unnecessarily increasing context length and potential overhead, we used only a single sentence for each example instead of full abstracts.

This few-shot prompting approach, however, did not affect the performance of either classification. Scores for both categories remained the same as those achieved with zero-shot prompting.

The final prompting technique we tested was step-by-step prompting. Here, we used a more concise definition along with step-by-step instructions guiding the model through the analysis process. We outline an example prompt for our CLS classification now; for more information on our prompts, including the ML variation of this one, see appendix.

CLS Step-by-Step Prompt

```
You are an expert in Computational Literary Studies (CLS).
Analyze the following text and determine whether its **main topic** falls
under CLS.
**CLS Definition:**
CLS applies computational methods (e.g., text mining, stylometry,
sentiment analysis) to literary texts (e.g., novels, poetry, drama).
It excludes studies focused on historical records, cultural trends, or
linguistic change unless literature is central.
**Task:**
Analyze the text below step by step:
1. Identify computational methods used if any.
2. Determine if the text focuses on literary texts.
3. Classify the text a CLS (1) or Not CLS (0) based on your findings.
4. Provide a brief explanation justifying your decision.
**Text:**
"abstract"
**Output Format:**
CLS Score: [1 or 0]
[Brief Explanation]
"""
```

This approach led to a modest improvement in ML scores, achieving an accuracy of 80.52% and an F1 score of 0.86. For CLS, however, we observed a substantial improvement, with accuracy increasing to 92.21% and the F1 score rising to 0.93.

Table 5.4 summarizes the results across all prompting techniques:

Table 5.4: Performance of using the Gemma3 27B model on the original CLS dataset with various prompts. Notice that Step-by-Step prompting performs best in both categories.

Prompting Technique	Accuracy (%)	F1 Score
CLS Results		
Zero-shot	89.61%	0.91
Few-shot	89.61%	0.91
Step-by-Step	92.21%	0.93
ML Results		
Zero-shot	79.22%	0.84
Few-shot	79.22%	0.84
Step-by-Step	80.52%	0.86

Overall, our experiments with the Gemma3 model provide several important insights. First, an increase in model size results in a substantial improvement in classification performance, highlighting the value of larger LLMs for this task. Second, our results demonstrate that the CLS and ML categories respond differently to various prompting approaches. Techniques that improve performance for one category may, in some cases, not do the same for the other. Notably, step-by-step prompting produced the highest scores for CLS and modest gains for ML, suggesting it may be particularly well suited for more nuanced tasks like CLS classification.

These findings highlight the importance of not relying on a single prompting strategy across categories. Instead, adopting a category-specific or mixed prompting approach may lead to better overall performance. To see if these findings hold up, we move onto a bigger model yet again and experiment with our prompting techniques.

5.3.4 GPT-4o Prompting

The last two models we examine are both developed by OpenAI. They differ from the previous models in that they are the first **proprietary** systems we tested and feature a substantially higher parameter count. We begin with the first OpenAI model we tested, GPT-4o, before moving on to the second, o3.

Although OpenAI has not disclosed the exact number of parameters for GPT-4o, it is “estimated to be well over one trillion”[3], making it several orders of magnitude larger than our previously largest model, which had 27 billion parameters. OpenAI also offers a dedicated Python module to facilitate prompting their models. This module adopts an “instruction” and “input” structure: in our use case, we used the “instruction” component to assign a role to the model and the “input” field to provide the relevant

[3] Shahriar et al., 2024.

context for its analysis.

Using this structure, we introduced our baseline and received the following results:

Table 5.5: Performance of using the GPT-4o model on the original CLS dataset with the baseline prompt

Category	Accuracy (%)	F1 Score
CLS	94.81%	0.95
ML	84.41%	0.89

As shown in Table 6.1, GPT-4o outperforms our previous best model, Gemma3, across both categories. Remarkably, even this baseline prompt achieves results that surpass those of Gemma3 under more engineered prompting conditions. However, this performance appears to plateau near these benchmarks. Our attempts to further improve scores using advanced prompting techniques yielded only marginal gains or, in some cases, slight declines. Using the same prompting strategies applied to Gemma3 (with minor adjustments; see Appendix for prompt details), we observed the following:

Table 5.6: Performance of using the GPT-4o model on the original CLS dataset with various prompts. Notice that CLS results are worse than the baseline while ML improves with zero-shot.

Prompting Technique	Accuracy (%)	F1 Score
CLS Results		
Zero-shot	92.21%	0.93
Few-shot	93.51%	0.94
Step-by-Step	89.61%	0.90
ML Results		
Zero-shot	85.71%	0.90
Few-shot	84.42%	0.90
Step-by-Step	84.41%	0.89

Table 5.6 shows that the baseline prompt remained the strongest for CLS classification, while engineered prompts produced only minor improvements for ML classification. These results suggest diminishing returns for prompt engineering in large proprietary models. GPT-4o's baseline performance leaves little headroom for improvement, indicating that gains from advanced prompting may be minimal when the model's internal logic is already highly optimized.

Interestingly, CLS classification continues to be an easier task for larger models such as GPT-4o, whereas the smaller models (e.g., LLaMMA and DeepSeek) performed comparatively better on ML classification.

5.3.5 o3 Prompting

The other OpenAI model we tested, o3, is a recent release that offers state-of-the-art performance as of May 2025. Known for its advanced reasoning abilities and strong generalization across tasks, o3 has consistently outperformed earlier models, including GPT-4o, in most benchmarks. Based on this, we anticipated that o3 would achieve superior results in our classification tasks as well.

Before testing engineered prompts, we first evaluated o3 using a baseline prompt. The results are summarized in Table 5.7.

Table 5.7: Performance of using the o3 model on the original CLS dataset with the baseline prompt

Category	Accuracy (%)	F1 Score
CLS	89.61%	0.90
ML	90.90%	0.94

The results in Table 5.7 are unexpected. As previously noted, we anticipated that o3 would surpass GPT-4o's performance, even if only by a small margin. However, o3 performs noticeably worse than GPT-4o in classifying the CLS category, with a drop of approximately 5%. Conversely, for ML classification, o3 achieves about a 5% higher accuracy than GPT-4o. These findings suggest that performance in these two tasks may depend heavily on how well the models align with the specific demands of each classification problem.

We also examined the effects of different prompting techniques on o3's performance:

Table 5.8: Performance of using the o3 model on the original CLS dataset with various prompts. Notice that CLS results are highest with few-shot prompting while ML performs best at baseline.

Prompting Technique	Accuracy (%)	F1 Score
CLS Results		
Zero-shot	92.21%	0.93
Few-shot	93.51%	0.94
Step-by-Step	92.21%	0.93
ML Results		
Zero-shot	89.61%	0.93
Few-shot	89.61%	0.93
Step-by-Step	85.71%	0.90

As shown in Table 5.8, the trends differ from those observed with GPT-4o. While GPT-4o exhibited no improvements for CLS classification and only minor gains for ML, o3 shows a notable improvement in CLS performance when using prompting techniques such as few-shot and step-by-step prompting, increasing from 89.61% (baseline) to 93.51%

(few-shot). This challenges our earlier hypothesis that advanced prompting provides only minimal benefits for larger models. On the other hand, for ML classification, prompting techniques offer no significant gains and in some cases (e.g., step-by-step) even lead to notable performance degradation.

These results suggest that the impact of prompting techniques is not uniform across models or tasks. For o3, prompting refinements substantially improve CLS classification but have little effect on ML classification. This reinforces the idea that prompting effectiveness depends on both the task and the model.

Furthermore, when considering all prompting techniques, CLS classification remains the easier task for larger models overall. However, o3 narrows the performance gap between CLS and ML classification compared to GPT-4o.

In summary, both o3 and GPT-4o demonstrate impressive capabilities, achieving around 90% accuracy across tasks. These results highlight both the potential and the task-specific variability in how large language models respond to prompt engineering. We discuss the broader implications of these findings in the next section.

5.4 Conclusion

Overall, these experiments provided valuable insights into the performance of different models on our task. We established baselines for a range of models and evaluated how they handle information extraction in the context of aiding the scientific survey paper creation process. Throughout this testing process, several patterns emerged, allowing us to confirm or reject prior hypotheses and refine our understanding of model capabilities. We now turn our attention to addressing our research questions.

Question: "How good are the information extraction qualities of different LLMs?"
Answer: Our experiments showed that simpler classification approaches without prompting (e.g., encoder-based models) did not produce promising results. While these models can theoretically perform classification, in our context, the dataset was too small to avoid overfitting, which led to poor performance. Utilizing pre-trained encoder-based models also provided poor performance as they were unable to perform within our context. By contrast, prompting-based approaches demonstrated strong performance without any additional training. This suggests that leveraging pre-trained LLMs through prompt engineering is a more effective strategy for modestly complex tasks where limited training data is available. To further look into how well these models performed, we move into the second research question.

Question: "Which prompting-based model performs the best in this context?"
Answer: While the encoder-based models performed very poorly in our experiments, the prompting-based models showed strong potential. Interestingly, we did not identify a single model that excelled universally; instead, different models performed best in each classification category. For the CLS classification task, OpenAI's GPT-4o achieved the highest performance with an accuracy of 94.81% and an F1 score of 0.95. In contrast, for the ML classification task, the best results came from o3, which achieved an accuracy

of 90.90% and an F1 score of 0.94.

Although it is not surprising that these large, state-of-the-art models achieved the best results overall, it is notable that GPT-4o outperformed o3 in CLS classification, despite o3 achieving higher scores on general testing metrics as outlined in Chapter 2. While we cannot pinpoint the exact reason for this difference, it is plausible that variations in model architecture and pre-training data could lend one model an advantage on certain task types.

Beyond the top performers, our experiments also showed that Gemma3 achieved decent results, while the smaller prompt-based models struggled significantly. This underscores the importance of model size for performance in tasks of this complexity. For instance, although Deepseek slightly outperformed LLaMa at the 7B parameter scale, it was unable to reach high accuracy without an increase in parameter size.

Finally, an interesting pattern emerged when comparing categories: ML classification appeared easier for models to achieve moderate performance (~70% accuracy), but harder to achieve excellent results (~90%). Conversely, CLS classification seemed more challenging for lower-performing models but allowed higher-performing models to achieve exceptional results more readily.

Question: "Which prompting strategy shows the most potential?"
Answer: For the stronger performing models, we tested several prompting strategies to evaluate their impact on classification performance. One key conclusion from these experiments is that no single prompting strategy consistently outperforms the others across all models. For example, in the case of Gemma3, step-by-step prompting led to a notable performance increase in both classification categories. In contrast, for the OpenAI models, step-by-step prompting was never the best approach; for GPT-4o in particular, it even caused a significant drop in performance. Interestingly, GPT-4o achieved its best CLS results using a simple zero-shot baseline prompt, and the ML results for this model improved by only about 1% with alternative prompting strategies. This suggests that highly optimized models may derive limited benefit from advanced prompting techniques.

However, this hypothesis was challenged by the results for o3, where few-shot prompting improved CLS classification accuracy by up to 4%. These findings indicate that while no single prompting strategy dominates, tailoring the prompting approach to the specific model can lead to meaningful performance gains.

Further illustration for the effectiveness of the different prompting strategies can be found in the appendix.

6
Model Application and Data Exploration

6.1 Model Application and Data Exploration

In the previous chapter we evaluated different large language models (LLMs) for their ability to extract and classify relevant information. From these experiments we established a baseline that shows how well our models perform at this task. In this chapter we intend to go further and move towards a more practical application of our best performing model from the previous chapter. The experiments presented in this chapter aim to illustrate how our model may be used to aid in the creation of high-quality scientific survey papers.

We start our with a brief section showing our attempts at clustering our data, using our best-performing model, in order to possibly observe and analyze certain trends within the data. The main part of this chapter focuses on reproducing sections of survey papers, specifically tables as they are filled with large amounts of structured information about scientific papers, and evaluate how effectively our model performs this task. Afterwards, we extend this approach by prompting our model to return the required information for a new table based on an updated dataset. Finally, we aim to judge how well our model generalizes and test our same approach in a different field than CLS.

6.1.1 Research Questions

Through the experiments conducted, we aim to answer the following research questions:

1. *How good are the information extraction qualities of different LLMs?*
2. *Does an increase in context help our best-performing model improve?*

As previously mentioned in Chapter 5 the first question is being revisited here, in greater depth, as these experiments allow us to provide a more detailed answer. The second question will be answered for the first time here, since these sets of experiments offered us the best opportunity to use the full text of our papers and not just the abstracts.

6.2 Charting the Data

Before attempting to recreate parts of established survey papers, we first explored our datasets to identify potential patterns or trends using our best-performing model, OpenAI's o3. The first dataset we use is based on the survey paper by Hatzel et al. (2023), for more information see Chapter 4. As a starting point, we prompted the model to extract the main topic of each paper from its abstract, following the approach used in Chapter 5. The resulting topics were collected into a CSV file and subsequently used for clustering. For our clustering we used embeddings together with the *all-MiniLM-L6-v2* sentence transformer.[1]

To properly create and assign clusters, we established groups of predefined clusters with names and values. The following is an example of one such cluster:

```
    "Sentiment and Emotion Analysis":
 ["sentiment analysis", "emotion detection", "opinion mining"]
```

We encoded the representative terms for each cluster and computed an average embedding to serve as the cluster vector. Similarly, the extracted main topics were normalized, encoded, and compared against the cluster vectors using cosine similarity. A topic was assigned to the cluster with the highest similarity score, provided the score exceeded a threshold of 0.35. Otherwise, the topic was labeled as "Uncategorized." After assignment, cluster vectors were recalculated to account for the newly added terms.

To possibly increase our coverage and minimize the amount of "Uncategorized" terms we additionally sent all "Uncategorized" topics, along with the predefined cluster names to our model using a separate prompt. The model was prompted to suggest new potential clusters based on the group of uncategorized terms, keeping in mind that clusters should not be either overly broad or overly specific. The output of this prompt was then put into a cluster suggestion list where we manually reviewed them to see if we should add any new predefined clusters.

The first result of this clustering process can be seen in the following graph. Here we decided to simply show the spread of main topics within our data in a bar chart.

From Figure 6.1 we observe that, the largest cluster within our data is "Literary and Textual Studies", which is characterized by terms such as "narrative analysis" or "style analysis". This outcome is expected, as 48 out of 77 papers in this dataset belong to the field of CLS, a field which focuses heavily on literary and textual studies. The second largest cluster is "Uncategorized". Due to the relatively small dataset size, we were not able to find any new potential predefined clusters from these uncategorized terms, that were not either overly broad or under-populated and highly specific for our dataset.

[1] Sentence Transformers (Hugging Face), 2025.

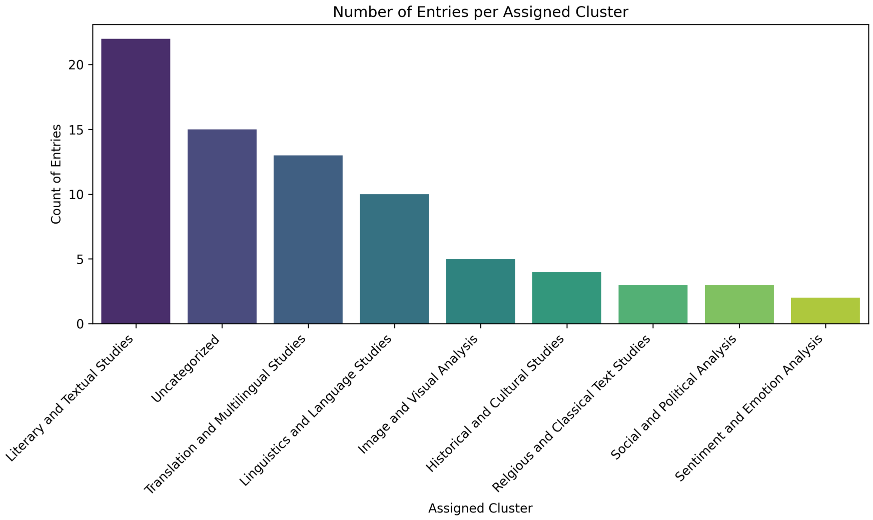

Figure 6.1: Main topic clusters for original CLS survey paper dataset by Hatzel et al. (2023) measured by count of entries per cluster

This resulted in about 19% of our dataset not falling into a predefined cluster. The rest of our data follows a relatively even distribution, though the more CLS adjacent clusters seem to be at the forefront, which is consistent given our data.

While this bar chart shows us a decent visualization of how our data is distributed, it does not show us any trends. Therefore, we further attempted to map our clusters using dimensionality reduction techniques such as PCA and UMAP to potentially see some patterns emerge. Unfortunately, these visualizations did not reveal any meaningful structure and produced uncorrelated spreads. For this reason, we decided to omit these results and proceed towards the next dataset.

The second dataset we analyzed was an updated version of the previous dataset, using the same conference papers but from 2023 and 2024 instead. It is further outlined in Chapter 4. Since this dataset still uses the same sources as the previous one, just with updated data from 2023 and 2024, we should see similar distributions here. This dataset is also more than double the size, thus ideally allowing us to create new predetermined clusters that should now be more populated. The results can be seen in Figure 6.2.

In this dataset, the largest cluster once again corresponds to Literary and Textual Studies, followed by Uncategorized. However, unlike in the previous analysis, the uncategorized portion was reduced through the introduction of two new model-suggested categories: **Authorship and Stylometry** and **Music and Audio Analysis**. This adjustment lowered the proportion of uncategorized data to about 17%. We also observe a slight increase in "Historical and Cultural Studies" with a minor decrease to "Religious and Classical Text Studies", though these changes are not substantial enough to indicate a trend.

As with the first dataset, applying clustering to this data did not provide us with any more discernible trends or patterns. While this visualization helps us get a better look at the data; to truly retrieve more details from our datasets, we need to dig deeper

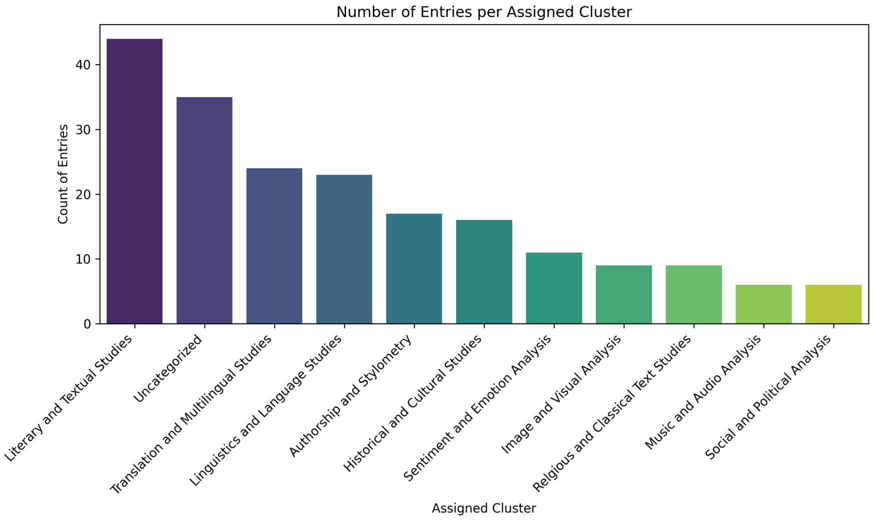

Figure 6.2: Main topic clusters for updated CLS survey paper dataset measured by count of entries per cluster

into the data.

6.3 Survey Paper Table Recreation

Although our clustering efforts did not produce any visible trends, they provided a useful overview of our data as well as a distribution of the main topics presented in our papers. However, to gain more detailed insights into the papers and to explore how such findings could aid in the creation of survey papers, we need to examine the data more closely. As such, we use this section to search and extract more details in our papers. Specifically, we start by looking for the information that is present within Table 3 of the paper by **Hatzel et al. (2023)** and attempt to recreate it. Thus, we are able to assess how well our model can extract specific information relevant to survey paper creation.

6.3.1 Original survey paper

Table 3 in the original survey paper by **Hatzel et al. (2023)** contains data for all papers that have a score of 1 in both CLS and ML. For these papers several attributes are listed: "Machine learning method", "Model name", "LS question/topic", "Annotations", and "Rules". The "machine learning method" and "model name" columns are relatively straight-forward while the "LS question/topic" column "denotes the question or topic of the paper with regard to literary studies."[2] "Annotations" and "Rules" is a simple yes/no column that indicates which approach the paper used; if it was more of a rule-based approach with pre-trained models or an annotation one with the authors training their own model and scaling their annotations.

[2] Hatzel et al., 2023.

To recreate this table, we first prompted our model for each quality with a zero-shot approach, providing our model with the abstract of the papers it previously scored as 1 in both the categories of CLS and ML as per the classification in Chapter 5, as well as the definitions for our qualities in accordance to the ones given in the original survey paper. We leave out the "Features" column for now as it is rather undefined. We do expect a difference in papers mentioned in our recreated table, since our accuracy is not at 100%, however, the overlap is big enough that we can use it to judge how well our model performed. Using just the abstract of our papers, our results can be seen in Table 6.4. For improved readability, we have moved it to the end of this chapter.

When we examine Table 6.4 and compare the entries for the papers that are also present in Table 3 of the original paper by **Hatzel et al. (2023)** we can already see a number of differences. Some of these can be attributed to variation in writing style between the original authors and our model; for example, classifying a method as “Supervised learning” rather than “Transformers.” Others, however, are genuine errors. As mentioned earlier, some papers that should be included in this recreation are missing, due to the model’s imperfect classification accuracy, such as the paper "‘This book makes me happy and sad and I love it’. A Rule-based Model for Extracting Reading Impact from English Book Reviews" by Koolen et al.[3] Conversely, due to the inaccuracy during our model’s scoring, a couple of new papers are included that should not be present such as "‘Entrez!’ she called: Evaluating Language Identification Tools in English Literary Texts" by Ketzan et al.[4]

A recurring pattern is our model’s preference to use the term "Supervised learning" in the "Machine learning method" column where the original table had used "Transformers". While this identification is not false, we do notice that our model often defaults to this term even when more specific methods are mentioned that should be written down instead. In those cases, we are not working with a matter of preference anymore but rather a false classification. One example of this is the paper by Steg et al., which is classified as "Supervised learning" within the "Machine learning method" column with more specifically "Linear Regression" in the "Model name" column. In comparison, the original table classifies the machine learning method as "Theil-Sen regressor, doc2vec" with no specific model name.

We also notice inaccuracies in the other columns, most notably in the "Annotations" and "Rules" columns. Here, our model overwhelmingly defaults to a score of neither and performs very poorly. This poor performance, together with the lackluster performance in the machine learning and model name category, can most likely be attributed to a lack of information for these categories within just the abstract of each paper. We create this hypothesis since our model’s performance in the "LS question/topic" column is very strong. After manually reviewing this category, we noticed that most of the time our model correctly identifies the topic. To support this claim, we ran a function to calculate the BERT score between our generated "LS question/topic" and the one in the original table. The results are as follows: no pair of topics received a score lower than 0.8 and many went over 0.85 with the highest being at 0.94. Since it is more likely that the main

[3] Koolen et al., 2022.

[4] Ketzan and Werner, 2022.

topic of a paper is referenced within the abstract of the paper, we assume that the lack of context is the cause for our poor results in the other categories and our model is indeed capable of extracting more complex information as long as the relevant details are present.

We decided to put this hypothesis to the test and run the same task, but providing our model with the full text for each paper this time rather than just the abstract. Thus, we are able to see whether an increase in context has a big impact on our model's performance.

However, since the move from abstract to full text is a big jump in total token length for each input, we were unable to use our best performing model, o3, since the costs are too high. Therefore, we decided to switch to the best alternative that would decrease cost, preserve performance as best as possible and be similar enough in architecture to our current model. The alternative we landed on was the o3-mini model. Since it is part of the same model series, the architecture is similar and being a smaller model, the costs per input token are halved. To see how well it performs, we decided to run a baseline test for this model in a similar fashion to the ones ran in Chapter 5. While o3-mini did not perform as well as o3 the results are within an acceptable range. In the ML category, we observed a 5% decrease in accuracy while the accuracy for CLS stayed stable.

Having established this trade-off, we proceeded with o3-mini for the extraction of column entries, while retaining the entries o3 procured for the classification step. Therefore, we reduce the impact of performance differences between models, as we stay with the same set of papers and only change models for the analysis itself.

Table 6.5 displays the results of our full text testing approach. Immediately, we notice improvements in many columns that we struggled with beforehand. The two columns with the weakest results in the abstract-only setting, "Annotation" and "Rules" show major improvements. While previously our model overwhelmingly defaulted to a score of neither, only sparsely going for a different outcome, our model now demonstrates a much better accuracy. Considering only the papers that are in both our recreated table and in the original table by Hatzel et al. (2023), we achieve an accuracy of about 70% for the "Annotation" and "Rules" columns.

As for the "Model name" and "Machine learning method" columns, we do see improvements as well. Often times in the abstract-only table the model would generate a method that does fit but was more general, such as supervised learning, and would fail to provide a specific model name. This is not the case here. For example, one paper by Schmidt et al. has the following properties for the "Machine learning method" and "Model name", in our original table: "Transformer" and "c2f". In the previous Table 6.4 it is classified as "Neural Network Approach" and "None". While "Neural Network Approach" is fine and works in this context and is not necessarily a misclassification, "None" is a misclassification. Our newer Table 6.5, improves this and does assign the correct value of "c2f" to the Model name column. This also happens again with "bookNLP" in the case of the Piper et al. paper where the previous model struggled with finding a correct model name while this one manages to find the correct classification.

The only category where we do not improve much is the "LS question/topic" column. Since our performance within this column was already high, increasing the context to the full text of the paper does not provide much more additional information to our model for this category, as the main topic of the paper is, in most cases, already laid out in the abstract. Thus we achieve a similar BERT score here as we had previously at about 0.86.

Overall, these results support our earlier hypothesis: increasing the available context allows the model to retrieve more detailed information and substantially improves accuracy in categories that are not always well represented in abstracts.

6.3.2 Updated Dataset

After making this discovery and generating good quantitative data that shows how our model performs at this task, we proceeded to evaluate it on an updated CLS dataset, using the same method. This updated dataset, introduced earlier in the chapter, comprises newly added papers from the conferences included in the original dataset by **Hatzel et al. (2023)**. Since no prior results or tables exist for this updated dataset, we create a new table to present our findings and offer a qualitative assessment of their plausibility. Our goal is to explore how the model can assist in generating tables commonly used in survey papers. The resulting tables are again presented at the end of this chapter.

Table 6.6 displays the results of our experiment. At first glance, no obvious trends emerge, but our results seem solid and reliable. The split between Annotation and Rule-based approaches is roughly 50/50 which is in line with our previous results in Table 6.5. While many classic ML methods are present here in this table, we do notice a slight upwards trend of more specific methods and models such as YOLO. To that extent, our model tends to describe these specific methods in more detail, increasing the content within each machine learning method column entry. Nevertheless, transformer models remain the predominant subset.

Topics on the other hand range wildly from a more literary focus to historical events to emotional analysis. To conclude this section, the results shown here seem authentic for our dataset and aim to prove that LLMs can aid in the recreation of parts of high-quality survey papers. However, because we created this table from an updated dataset that has no previous ground-truth table associated with it, we are unable to verify the authenticity of these results. While we would expect a 80-90% accuracy, based on previous results, we do not know which parts of this table are accurate and which are not. This is a crucial challenge and will be addressed, though for now we leave this table here to show the capabilities of our model and move onto the next section, which takes place in a different domain, to test our generalization capabilities.

6.3.3 Table Recreation in a Different Domain

After experimenting with our two datasets based on the paper by **Hatzel et al. (2023)**, we decided to explore a different domain to evaluate how well our approach generalizes beyond CLS-focused papers. To this end, we created a dataset from a different survey paper: “Neural entity linking: A survey of models based on deep learning” by **Sevgili et al. (2022)**. The dataset was constructed from the entries in Table 2 of that paper. More details about this dataset can be found in Chapter 4.

Our goal with this new dataset remains similar: to reconstruct the table given only the column names, their definitions, and the papers we analyze. We used the same model as before, o3-mini, testing with both abstracts and full texts to investigate the impact of increased context on model performance. Due to token limitations, we limited this experiment to a single column of the table.

We chose the column “Encoder type” for this experiment because it contains a diverse set of labels, each defined by the authors, making it well-suited for our prompting approach. We prompted the model to identify the encoder type and provided the list of expected labels along with their definitions as short descriptions. We first tested using only the abstracts, then proceeded to testing with full texts. The results can be seen in Table 6.2 and Table 6.3

To evaluate accuracy, we assigned a score of 1 for correct classifications and 0 for incorrect ones. Partial matches, such as identifying “CNN” when the correct label is “LSTM + CNN,” received a score of 0.5. Scores were adjusted proportionally when multiple types were present. Using abstracts only, the model achieved a score of 25.25/55 (45.91%). Although better than random guessing, this score is below expectations, likely due to the limited context in abstracts. Testing with full text was very effective, increasing the score to 40.55/55 (73.73%), indicating a substantial improvement from increased context. This result demonstrates that our model can effectively assist in creating survey papers, even across different domains.

Table 6.1: Performance of using the o3 and o3-mini models to recreate the "encoder type" column of Table 2 in the neural entity linking survey paper by Sevgili et al. (2022). Results are shown for using only the abstract during prompting compared to the whole text.

Method	Accuracy (%)	Score
Abstract only	45.91%	25.25/55
Full text	73.73%	40.55/55

6.4 Conclusion

From these experiments, we gained valuable insights as well as qualitative and quantitative results that demonstrate the capabilities of our best-performing model in aiding in the creation of high-quality survey papers. We were able to see what our data looks like as well as recreate tables from our datasets and even create a completely new table from an updated dataset. However, with the creation of a new table comes a new challenge as well. Since we do not have a ground-truth to compare to, we are unable to verify how accurate our resulting table is. While it does look convincing and reflects trends that we expect, a certain amount of human investigation is necessary to fully prove that the results are sound. We come back to this challenge in the following Chapter 7 and now turn our attention towards our research questions.

Question: How good are the information extraction qualities of different LLMs?
Answer: This question was first addressed in Chapter 5, where we compared multiple LLMs and evaluated their performance. In this chapter, we extended that investigation by focusing specifically on our previously best-performing model, o3, and its more cost-efficient variant, o3-mini. Here, the models were tasked with extracting detailed information from both abstracts and full-text papers across multiple categories, such as *main topic*, *machine learning method*, *encoder_type*, and others. This task differed from that in Chapter 5, where we only asked the broader questions of whether a paper used ML methods and/or belonged to the field of CLS.

From these experiments, we find that the models can accurately extract structured information from scientific texts. Broader categories remain easier to identify, as reflected by the strong performance in the *LS question/topic* column, but the models also perform well in more nuanced categories, such as differentiating between rule-based and annotation-based approaches, where o3-mini achieved 70% accuracy, with o3 likely performing slightly better, though we were unable to test that assumption. These results suggest that while LLMs excel at high-level categorization, they can also achieve strong performance on fine-grained information extraction, particularly when applied to well-structured scientific writing.

Question: Does an increase in context help our best-performing model improve?
Answer: The experiments in this chapter allowed us to examine whether increasing the available context for our model would enhance its performance or potentially overwhelm it. The rationale for this increase was straightforward: since we were seeking fine-grained information that was often absent from abstracts, we needed to provide the model with sections of text where it could actively locate the relevant details. As this information was not confined to any specific section in the papers, the most reliable approach was to supply the full text. This significantly expanded the context and token count, which, due to cost constraints, required switching from o3 to its more cost-efficient variant, o3-mini. The overall results of this increase in context are as follows.

Working with the full text rather than just abstracts allowed the model to retrieve more detailed and accurate information. Performance improved across multiple categories, particularly those involving information too specific to appear in the abstract. For example, in the neural entity linking paper (Section 6.3.3), accuracy for identifying the encoder type increased from about 46% to 74%. We also observed an improvement in the quality of the reproduced table in Section 6.3.1. Overall, these results indicate that higher-end models like o3 and o3-mini not only handle large context windows effectively, but can also deliver improved accuracy when extracting fine-grained details from full-length scientific texts.

Table 6.2: Neural entity paper table recreation with abstracts

author	predicted encoder type	actual encoder type	score received
Sun et al	CNN, Tensor net.	CNN, Tensor net.	1
Francis-Landau et al	CNN	CNN	1
Fang et al	n/a	word2vec-based	0
Yamada et al	word2vec-based	word2vec-based	1
Zwicklbauer et al	n/a	word2vec-based	0
Tsai and Roth	word2vec-based	word2vec-based	1
Nguyen et al	CNN	CNN	1
Globerson et al	Atten.	n/a	0
Cao et al	n/a	word2vec-based	0
Eshel et al	n/a	GRU + Atten.	0
Ganea and Hofmann	Atten.	Atten.	1
Moreno et al	n/a	word2vec-based	0
Gupta et al	n/a	LSTM	0
Nie et al	Atten.	LSTM + CNN	0
Sorokin and Gurevych	n/a	CNN	0
Shahbazi et al	n/a	Atten.	0
Le and Titov	n/a	Atten.	0
Newman-Griffis et al	word2vec-based	word2vec-based	1
Radhakrishnan et al	n/a	n/a	1
Kolitsas et al	n/a	LSTM	0
Sil et al	CNN, Tensor net.	LSTM + Tensor net.	0.5
Upadhyay et al	n/a	CNN	0
Cao et al	Atten.	Atten.	1
Raiman and Raiman	n/a	n/a	1
Mueller and Durrett	Atten., CNN	GRU + Atten. + CNN	0.75
Shahbazi et al	ELMo	ELMo	1
Logeswaran et al	BERT	BERT	1
Gillick et al	n/a	FFNN	0

Table 6.2: Neural entity paper table recreation with abstracts (cont.)

author	predicted encoder type	actual encoder type	score received
Peters et al	BERT, Atten.	BERT	0.5
Le and Titov	n/a	LSTM	0
Le and Titov	n/a	Atten.	0
Fang et al	n/a	LSTM	0
Martins et al	LSTM	LSTM	1
Yang et al	Atten.	Atten.	1
Xue et al	n/a	CNN	0
Zhou et al	n/a	n/a	1
Broscheit	BERT	BERT	1
Hou et al	n/a	Atten.	0
Onoe and Durrett	n/a	ELMo + Atten. + CNN + LSTM	0
Chen et al	BERT	BERT	1
Wu et al	BERT	BERT	1
Banerjee et al	Atten.	fastText	0
Wu et al	n/a	ELMo	0
Fang et al	Atten.	BERT	0
Chen et al	Atten., BERT	Atten., BERT	1
Botha et al	n/a	BERT	0
Yao et al	BERT	BERT	1
Li et al	n/a	BERT	0
Poerner et al	BERT	BERT	1
Fu et al	n/a	M-BERT	0
Mulang' et al	BERT	Atten. or CNN or BERT	1
Yamada et al	BERT	BERT	1
Gu et al	n/a	BERT	0
Tang et al	BERT, Atten.	BERT	0.5
De Cao et al	BERT	BART	0

Table 6.3: Neural entity paper table recreation with full text

author	predicted encoder type	actual encoder type	score received
Sun et al	CNN, Tensor net.	CNN, Tensor net.	1
Francis-Landau et al	CNN	CNN	1
Fang et al	n/a	word2vec-based	0
Yamada et al	word2vec-based	word2vec-based	1
Zwicklbauer et al	word2vec-based	word2vec-based	1
Tsai and Roth	word2vec-based	word2vec-based	1
Nguyen et al	CNN, GRU	CNN	0.5
Globerson et al	Atten.	n/a	0
Cao et al	word2vec-based	word2vec-based	1
Eshel et al	GRU, Atten.	GRU, Atten.	1
Ganea and Hofmann	Atten.	Atten.	1
Moreno et al	word2vec-based	word2vec-based	1
Gupta et al	LSTM, CNN, FFNN	LSTM	0.33
Nie et al	LSTM	LSTM + CNN	0.5
Sorokin and Gurevych	CNN	CNN	1
Shahbazi et al	Atten.	Atten.	1
Le and Titov	FFNN	Atten.	0
Newman-Griffis et al	word2vec-based	word2vec-based	1
Radhakrishnan et al	word2vec-based	n/a	0
Kolitsas et al	LSTM	LSTM	1
Sil et al	CNN, LSTM, and Tensor net.	LSTM + Tensor net	0.67
Upadhyay et al	CNN	CNN	1
Cao et al	FFNN	Atten.	0
Raiman and Raiman	LSTM	n/a	0
Mueller and Durrett	GRU + Atten. + CNN	GRU + Atten. + CNN	1
Shahbazi et al	ELMo	ELMo	1
Logeswaran et al	BERT	BERT	1
Gillick et al	FFNN	FFNN	1

Table 6.3: Neural entity paper table recreation with full text (cont.

author	predicted encoder type	actual encoder type	score received
Peters et al	BERT	BERT	1
Le and Titov	LSTM	LSTM	1
Le and Titov	FFNN	Atten.	0
Fang et al	LSTM	LSTM	1
Martins et al	LSTM	LSTM	1
Yang et al	CNN, Atten.	CNN, Atten.	1
Xue et al	CNN	CNN	1
Zhou et al	LSTM	n/a	0
Broscheit	BERT	BERT	1
Hou et al	n/a	Atten.	0
Onoe and Durrett	ELMo + Atten. + CNN + LSTM	ELMo + Atten. + CNN + LSTM	1
Chen et al	BERT	BERT	1
Wu et al	BERT	BERT	1
Banerjee et al	LSTM	fastText	0
Wu et al	ELMo. Atten	ELMo	0.5
Fang et al	BERT	BERT	1
Chen et al	BERT	Atten. + BERT	0.5
Botha et al	BERT	BERT	1
Yao et al	BERT	BERT	1
Li et al	BERT	BERT	1
Poerner et al	BERT	BERT	1
Fu et al	BERT	M-BERT	1
Mulang' et al	LSTM, BERT	Atten. or CNN or BERT	0.5
Yamada et al	BERT	BERT	1
Gu et al	BERT	BERT	1
Tang et al	BERT	BERT	1
De Cao et al	BERT	BART	0

Table 6.4: Original CLS survey paper table recreation with abstracts using o3 (cont.)

Author	Machine learning method	Model name	LS question/topic	Anno.	Rules
Kunilovskaya et al	Supervised learning	None	Translationese in Russian literature		
Cooper et al	Supervised learning; Topic modeling	None	Storyteller personalities in Boccaccio's Decameron		
Tian et al	Supervised learning (text classification)	None	Language change in Chinese Biji	x	
Cranenburgh et al	Supervised learning	Cosine Delta-based Stylometric Classifier	Stylometric measurement of literariness		
Steg et al	Supervised learning	Linear Regression	Narrativity detection in texts		
Völkl et al	Topic modeling	None	Gender in Spectator periodicals		
Brottrager et al	Automated sentiment detection (Sentiment Analysis)	None	Textual features and historical literary reception		
Du et al	Classification	None	Dispersion-based keyness measures		
Schröter at al	Topic modeling	None	Validation of topic modeling for thematic analysis in narrative fiction		
Abdibayev	Deep learning language modeling	None	Computational poetics of limericks		
Ehrmanntraut	Neural sentence embeddings	None	Early modernism in German poetry	x	
Weimer et al	Supervised learning	None	Literary comment in narrative texts		
Shin et al	Unsupervised learning	Word2Vec	Usage of "queer" in Modernist literature		x

Table 6.5: Original CLS survey paper table recreation with full text using o3 and o3-mini

Author	Machine learning method	Model name	LS question/topic	Anno.	Rules
Parigini et al	Fine-tuning BERT for Named Entity Recognition	Italian-xxl-cased	Computational detection of dubitative text	x	
Zhang	Fine-tuned transformer-based text classification	ECCO-BERT	Detecting genre shifts in texts	x	
Piper et al	Predictive modeling using BERT-based tagging	bookNLP	Role of things in fiction		x
Perri et al	Graph Neural Networks	None	Graph analysis of Tolkien's legendarium	x	
Konle et al	Topic Modeling (Latent Dirichlet Allocation)	None	Plot modeling with temporal graphs		x
Ketzan et al	Subword embedding text classification	fasttext	Evaluating language identification in literature		x
Eder et al	Cosine Delta classifier	GloVe	Optimizing word frequencies for authorship		x
Zundert et al	Topic Modeling (Top2Vec)	Standard multilingual universal sentence encoder	Topic models as genre proxy		
Grotti et al	Support Vector Machine (SVM)	None	Collaborative authorship in Good Omens	x	
Schmidt et al	Neural networks	c2f	Automated extraction of character networks	x	x
Wöckener et al	Conditioned recurrent neural network (RNN) language model	GPT-2	Learning poetic style from examples	x	
Schmidt et al	Transformer-based language model fine-tuning	gbert-large	Emotion classification in German plays	x	
Schneider et al	Logistic regression classifier	de_core_news_lg	Automatic chiasmus detection in literature	x	x

Table 6.5: Original CLS survey paper table recreation with full text using o3 and o3-mini (cont.)

Author	Machine learning method	Model name	LS question/topic	Anno.	Rules
Kunilovskaya et al	Support Vector Machine (SVM) with RBF kernel	None	Translationese in Russian literary texts	x	
Cooper et al	Logistic Regression	None	Distinct storyteller personalities in Decameron	x	
Tian et al	SVM	Guwen-RoBertA	Computational dating of Chinese texts	x	
Cranenburgh et al	Regularized logistic regression	None	Quantifying King's literariness via stylometry	x	
Steg et al	Theil-Sen Regressor	None	Computational narrativity through reader perception	x	
Völkl et al	Latent Dirichlet Allocation (LDA)	None	Gender discourse in 18th-century periodicals		
Brottrager et al	Support Vector Machine (SVM)	Sentence-BERT	Predicting literary reception from texts	x	
Du et al	Multinomial Naive Bayes	None	Evaluating measures of literary distinctiveness		
Schröter at al	Latent Dirichlet Allocation (LDA)	None	Topic modeling and literary aboutness	x	x
Abdibayev	Transformer-based language models using probabilistic sequence (log-probability) evaluation	GPT-2	Evaluating language models' poetic abilities		x
Ehrmanntraut	Siamese neural network using triplet margin loss for similarity learning	paraphrase-mpnet	Measuring similarity in German poetry	x	
Weimer et al	Supervised classification using decision tree and logistic regression	None	Literary comment concept consistency	x	
Shin et al	Word2Vec	HistWords	Woolf's queer sentiment analysis	x	

Table 6.6: Updated CLs survey paper dataset Table creation with full text using o3 and o3-mini

Author	Machine learning method	Model name	LS question/topic	Anno.	Rules
Lang et al	YOLO object detection	YOLOv8	Detecting alchemical apparatus in prints	x	
Zhou et al	Machine learning-based event extraction from audio description	None	Evaluating ML narrative event extraction		
Sarbach-Pulicanii	SVM	None	Profiling anonymous Corsican authors	x	
Verkijk et al	Ontology-driven rule-based reasoning	None	Historical event reconstruction ontology		x
Kaše et al	Monte Carlo Simulation	None	Temporal uncertainty in historical data		
Craig et al	Sentence alignment via dynamic programming on pre-trained sentence embeddings	LaBSE	Sentence alignment for classical texts		x
Bambaci et al	Deep Learning–based Handwritten Text Recognition	Kraken	Enhancing HTR via scholarly editions	x	x
Nielbo et al	Latent Dirichlet Allocation and sentiment classification	BERTweet	Oscillatory dynamics in Teresa's writings		x
Camps et al	doc2vec	None	Medieval French love and war	x	
Zundert et al	Stanford Multi-Pass Sieve Coreference Resolution	None	Character detection and gender dynamics		x
Koolen et al	Hierarchical clustering (unsupervised clustering on cosine distances of word n-grams)	None	Computational analysis of formulaic expressions		x
Ströbel et al	Generative text-to-image model	DALL-E 3	Evaluating AI Historical Image Reenactment		x
Lassen et al	Random Forest	None	Gender bias in literary canonicity	x	

Table 6.6: Updated CLs survey paper dataset Table creation with full text using o3 and o3-mini (cont.)

Author	**Machine learning method**	**Model name**	**LS question/topic**	**Anno.**	**Rules**
Vidal-Gorène et al	Word-based Convolutional Recurrent Neural Network	RASAM	Enhancing Arabic handwritten text recognition	x	
Gabay et al	Object detection using YOLO-based deep learning	YOLOv8L	Computational analysis of French orthography	x	
Bekker-Nielsen Dumbar et al	Minimum Edit Distance (Levenshtein Distance)	None	Investigating influenza vs. grippe naming		x
Schöch et al	Support Vector Machine classifier	None	Language effects on stylometric attribution	x	
Bamman et al	Classification	Llama 3 8B	LLMs for literary classification sense-making	x	
Arnold et al	Multimodal LLM-based caption generation combined with cosine similarity–based text embedding	GPT-4 Turbo	Explainable visual heritage search		x
Kurar-Barakat et al	Multi-label Convolutional Neural Network	VGG-19	Computational Paleography of Hebrew Manuscripts	x	
Jacobsen et al	Rule-based textual analysis using predefined metrics and sentiment scoring	spaCy's large English pre-trained model and VADER	Fanfiction vs published narrative quality		x
Ziegler	Sequence tagging with a single-layered Bi-LSTM + CRF decoder	Finetuned de-model (FlairNLP contextual character embeddings)	Event extraction in historical records	x	x
Abel et al	None	None	Quantifying live setlist variety		x

Table 6.6: Updated CLs survey paper dataset Table creation with full text using o3 and o3-mini (cont.)

Author	Machine learning method	Model name	LS question/topic	Anno.	Rules
Sobotkova et al	None	None	Regional burial mounds intervisibility analysis		x
Vozhik et al	Latent Dirichlet Allocation (LDA)	None	Censorship and literary topical dissociation		
Ryan et al	Fine-tuned BERT for text classification	bert-base-multilingual-cased	Tracking multilingualism in book titles	x	
Illmer et al	None	None	Statistical analysis of one-act plays		x
Maksimova et al	Zero-shot classification	CLIP	Zero-shot classification of historical photos		x
Nguyen et al	Transformer-based Cross-Encoder	BigBird	Transformer models for authorship verification	x	
Bizzoni et al	Adaptive Fractal Analysis	None	Fractal analysis of scientific writing		
Cortal et al	Discrete emotion classification using fine-tuned CamemBERT	CamemBERT	Emotion component dynamics in narratives	x	
Lassen et al	Named Entity Recognition	DaCy-large	Intersectional bias in NER models		x
Öhman et al	Lexicon-based emotion detection using affect intensity lexicons and word embeddings	RoBERTa	Emotional arcs and literary quality		x
Tudor et al	Supervised transformer-based Named Entity Recognition	RA	Historical Swedish named entity recognition	x	
Stüssii et al	Transformer-based fine-tuning for sequence tagging	GPT-3.5-Turbo	Latin POS tagging via GPT	x	
Löfgren et al	Fine-tuning a pre-trained transformer model for sequence-to-sequence text correction	ByT5	OCR correction for historical texts	x	

Table 6.6: Updated CLs survey paper dataset Table creation with full text using o3 and o3-mini (cont.)

Author	Machine learning method	Model name	LS question/topic	Anno.	Rules
Dekker et al	Conditional Random Field	BERT	Annotated early modern chronicles corpus	x	
Arnold et al	Neural network binary classification using a fine-tuned language model	German uncased BERT	Short quotation linking in literature	x	
Szemes et al	k-means algorithm	None	Sentence structure in Hungarian novels		x
Wagner et al	Supervised text classification using transformer-based models with dynamic programming segmentation based on PMI	Distilroberta	Topical segmentation of Holocaust testimonies	x	
Chen et al	IBM Model 2 word alignment	IBM Model 2	Visualizing connotation in classical poetry		x
Wijers et al	Cluster analysis (Delta method) and principal component analysis (PCA)	None	Mankell style evolution and translation		x
Kugler et al	Supervised token classification using a neural network	BERT	Reconstructing texts from embeddings	x	
Guhr et al	Transfer Learning-based Named Entity Recognition	bert-base-cased	Ambient sound in Gothic fiction	x	
Szemes et al	Sentence embeddings with cosine similarity using SBERT	all-MiniLM-L6-v2	Measuring innovation in dramatic dialogue		x
Mélanie-Becquet et al	BiLSTM-CRF	CamemBERT	Computational analysis of French literature	x	

7
Conclusion

7.1 Conclusion

In this thesis, we have tackled the overarching question: "Are LLMs capable of aiding in the creation of high-quality scientific survey papers?". To do so, we constructed several datasets as outlined in Chapter 4 and used them for our experiments in the two following chapters. Chapter 5 sets a baseline for us and provides quantitative results regarding the capabilities of different LLMs to extract information from scientific papers. This aims to show how different LLMs may gather information to aid authors in survey paper creation. We take this approach one step further in Chapter 6, where we retrieve more fine-grained information and use the results of this extraction to reconstruct tables in the same manner they appear in our survey papers. We then compare our reconstructed tables to the ground-truth tables and evaluate our best-performing model's performance. Further, we also create a table from the ground up without having a ground-truth table to compare to, thus showing a real application for our model in this field.

From these experiments, we have gained a multitude of insights into LLM performance in aiding the creation of high-quality scientific survey papers. Higher-end models such as o3 and GPT-4o show promising performance in the field of information extraction, with accuracies of up to 90% in our field of CLS. While they perform better at high-level categorization, o3 in particular is also able to extract fine-grained information from large context fields, such as full scientific papers, with decent accuracy. To this extent, when given the full context of each paper, o3 was able to reconstruct tables for our survey paper with only a few errors in both the field of CLS and neural entity linking.

While these results show promise for LLMs in this field, we do wish to raise concern regarding one challenge when using these models in real-life applications. As we noticed during the creation of a completely new table based on an updated dataset from an existing survey paper, if no ground-truth table or similar reference is available, we are unable to confirm the accuracy of the resulting output without manual review. While we may achieve an accuracy of 70–80% in the individual columns beforehand, and thus expect this accuracy to hold in the future, we are unable to identify errors in our output at first glance. Therefore, we highlight that while the actual performance of LLMs in aiding the creation of high-quality scientific survey papers is solid, future work into the verification of results is needed.

7.1.1 Limitations

While our results are promising, several limitations of this thesis should be highlighted. First, our datasets are relatively small in scale and domain specific. As such, results may vary if larger or more diverse datasets were used, and we cannot guarantee that the observed accuracies generalize across different fields of research. Second, while we evaluated general-purpose LLMs such as o3 and GPT-4o, specialized domain specific models (e.g., biomedical or legal LLMs) may achieve stronger performance in their respective areas of expertise. Finally, hallucination remains a challenge in using LLMs for scientific text generation. Although our results show strong accuracy in information extraction and decent accuracy in table construction, errors may still occur without being immediately visible, as mentioned in the previous section. In future iterations of this work, hallucination detection mechanisms, such as KnowHalu by **Zhang et al. (2024)** or HaluCheck by **Heo et al. (2025)** could be integrated into the pipeline to reduce the likelihood of unnoticed errors.

7.1.2 Future and Current Work

We are not the first to ask this question or to test the information extraction capabilities of LLMs. In February 2025, OpenAI announced "Deep Research," an update to ChatGPT that aims to provide users with information from credible online sources, including scientific papers.[1] OpenAI leverages o3, the same model we conducted our research with, to "interpret and analyze massive amounts of text, images, and PDFs on the internet, pivoting as needed in reaction to information it encounters." While this deep research mode provides a detailed output with references included, cases of the model providing inaccurate information are still prominent, as described in Derek Lowe's article[2] and OpenAI's own technical report on o3's hallucination tendencies.[3]

Similar tools to deep research, and to a lesser extent the results of this thesis, also exist, such as "Ai2 ScholarQA,"[4] which aims to provide users with help for literature reviews. Similar to deep research by OpenAI, it uses Retrieval-Augmented Generation (RAG), so sources are provided for each claim. While hallucinations are reduced here in comparison to deep research, as this tool relies on an "evidence-first" pipeline, meaning citations are gathered before statements are made, errors still occur, as there is no contradiction detection or similar mechanism built in.

[1] OpenAI, 2025a.

[2] Lowe, 2025.

[3] OpenAI, 2025c.

[4] Allen Institute for AI, 2025.

7.1.3 Outlook

Overall, progress is clearly being made in this field; however, human verification of model output remains essential. Despite innovations that reduce hallucinations and contradictions, errors can still slip through. As outlined in Section 7.1.1, improvements in dataset size, specialized models, and hallucination detection could increase reliability, but only when models achieve human-level accuracy across diverse datasets can we begin to consider skipping human oversight. This may be realistic for simpler classification tasks, as demonstrated by **Goh et al. (2020)** and our 90% accuracy in Chapter 5. Yet, for more fine-grained information retrieval, such as in Chapter 6 and **Dasigi et al. (2021)**, significant challenges remain. Even large-scale efforts like OpenAI's "Deep Research" continue to struggle with hallucination. We therefore conclude that while LLMs can meaningfully support the creation of scientific survey papers, especially in surface-level or supporting tasks, their output must still be verified and cannot yet be trusted blindly.

Appendices

A

Additional Material

A.1 Prompts

We present some of the prompts used during our testing in Chapter 5.

CLS Baseline Prompt for LLaMMA2, Deepseek, and Gemma3

```
You are an expert in Computational Literary Studies (CLS).
**CLS Definition:**
CLS applies computational methods (e.g., text mining, stylometry,
sentiment analysis) to literary texts (e.g., novels, poetry, drama).
It excludes studies focused on historical records, cultural trends, or
linguistic change unless literature is central.
Your task is to analyze the text below and determine if it falls under
computational literary studies or not. Return only your analysis.
**Text:**
"{content}"
**Analysis:**
```

ML Baseline Prompt for LLaMMA2, Deepseek, and Gemma3

```
You are an expert in Machine Learning (ML).
**ML Definition:**
ML focuses on the development of algorithms that improve automatically
through experience. It includes methods such as supervised learning,
reinforcement learning, and natural language processing. Simple
rule-based algorithms and statistical evaluation are not machine
learning.
Your task is to analyze the text below and determine if it mentions
machine learning techniques or not. Return only your analysis.
**Text:**
"{content}"
**Analysis:**
```

CLS Zero Shot Prompt for Gemma3

```
You are an expert in Computational Literary Studies (CLS).
Analyze the following text and determine whether its **main topic** falls
under CLS.
**CLS Definition:**
CLS applies computational methods (e.g., text mining, stylometry,
sentiment analysis) to literary texts (e.g., novels, poetry, drama).
It excludes studies focused on historical records, cultural trends, or
linguistic change unless literature is central.
**Task:**
1. Classify the text as **1** or **0** where 1 represents a score of
’CLS’ and 0 a score of ’Not CLS’.
2. Provide a **one-sentence explanation** justifying your decision.
**Text:**
"abstract"
**Output Format:**
CLS Score: [1 or 0]
[Brief Explanation]
```

ML Zero Shot Prompt for Gemma3

```
You are an expert in Machine Learning (ML).
Analyze the following text and determine whether it uses **Machine
Learning** methods.
Machine Learning methods are defined as follows:
**ML Definition:**
Machine Learning (ML) focuses on algorithms that improve automatically
through experience. It includes supervised learning (e.g.,
classification, regression),unsupervised learning (e.g., clustering),
and reinforcement learning. Algorithms that do not learn from data
(e.g., rule-based systems) are not considered ML.
**Task:**
1. Classify the text as **1** or **0** where 1 represents a score of ’ML’
and 0 a score of ’Not ML’.
2. Provide a **one-sentence explanation** justifying your decision.
**Text:**
"abstract"
**Output Format:**
ML Score: [1 or 0]
[Brief Explanation]
```

CLS Few Shot Prompt for Gemma3

```
You are an expert in Computational Literary Studies (CLS).
Analyze the following text and determine whether its **main topic** falls under CLS.
**CLS Definition:**
CLS applies computational methods (e.g., text mining, stylometry, sentiment analysis) to literary texts (e.g., novels, poetry, drama). It excludes studies focused on historical records, cultural trends, or linguistic change unless literature is central.
**Task:**
1. Classify the text as **1** or **0**, where **1** represents ’CLS’ and **0** represents ’Not CLS’.
2. Provide a **one-sentence explanation** justifying your decision.
**Text:**
"abstract"
**Output Format:**
CLS Score: [1 or 0]
[Brief Explanation]
**Examples:**
Input:
"Poem generation with language models requires the modeling of rhyming patterns. We propose a novel solution for learning to rhyme, based on synthetic data generated with a rule-based rhyming algorithm."
Output:
1
"Computational methods are applied to a literary text (poem generation)."
Input:
"This paper explores the capacity of computer vision models to discern temporal information in visual content, focusing specifically on historical photographs."
Output:
0
"The study is centered on historical photographs, not literary texts."
```

ML Few Shot Prompt for Gemma3

```
You are an expert in Machine Learning (ML).
Analyze the following text and determine whether it uses **Machine
Learning** methods.
Machine Learning methods are defined as follows:
**ML Definition:**
Machine Learning (ML) focuses on algorithms that improve automatically
through experience. It includes supervised learning (e.g.,
classification, regression), unsupervised learning (e.g., clustering),
and reinforcement learning. Algorithms that do not learn from data (e.g.,
rule-based systems) are not considered ML.
**Task:**
1. Classify the text as **1** or **0** where 1 represents a score of 'ML'
and 0 a score of 'Not ML'.
2. Provide a **one-sentence explanation** justifying your decision.
**Text:**
"abstract"
**Output Format:**
ML Score: [1 or 0]
[Brief Explanation]
**Examples:**
Input:
"We fine-tune a GPT-2 English model with 124M parameters on 142 MB of
natural poems and find that this model generates consecutive rhymes
infrequently (11
Output:
1
"The study involves fine-tuning a language model, which is a machine
learning method."
Input:
"We find a statistically significant correlation between violent
discourse and emotional expression throughout the analyzed period."
Output:
0
"The study only mentions a focus on statistical correlation, not
necessarily machine learning methods."
```

CLS Step-by-step Prompt for Gemma3

```
You are an expert in Computational Literary Studies (CLS).
Analyze the following text and determine whether its **main topic** falls under CLS.
**CLS Definition:**
CLS applies computational methods (e.g., text mining, stylometry, sentiment analysis) to literary texts (e.g., novels, poetry, drama). It excludes studies focused on historical records, cultural trends, or linguistic change unless literature is central.
**Task:**
Analyze the text below step by step:
1. Identify computational methods used if any.
2. Determine if the text focuses on literary texts.
3. Classify the text a CLS (1) or Not CLS (0) based on your findings.
4. Provide a brief explanation justifying your decision.
**Text:**
"abstract"
**Output Format:**
CLS Score: [1 or 0]
[Brief Explanation]
```

ML Step-by-step Prompt for Gemma3

```
You are an expert in Machine Learning (ML).
Analyze the following text and determine whether it uses **Machine Learning** methods.
Machine Learning methods are defined as follows:
**ML Definition:**
Machine Learning (ML) focuses on algorithms that improve automatically through experience. It includes supervised learning (e.g., classification, regression), unsupervised learning (e.g., clustering), and reinforcement learning. Algorithms that do not learn from data (e.g., rule-based systems) are not considered ML.
**Task:**
Analyze the text below step by step:
1. Identify machine learning methods used if any.
2. Consider whether the methods used are explicitly ML and not rule-based.
3. Classify the text as ML (1) or Not ML (0) based on your findings.
4. Provide a brief explanation justifying your decision.
**Text:**
"abstract"
**Output Format:**
ML Score: [1 or 0]
[Brief Explanation]
```

The OpenAI API prefers prompts to be given through *instructions* and *prompts* separately, so it received the same instructions for every task with only the prompt part changing.

CLS instructions for GPT-4o and o3

```
You are an expert in Computational Literary Studies (CLS).
Definition:
CLS applies computational methods (e.g., text mining, stylometry,
sentiment analysis) to literary texts (e.g., novels, poetry, drama).
It excludes studies focused on historical records, cultural trends, or
linguistic change unless literature is central.
```

ML instructions for GPT-4o and o3

```
You are an expert in Machine Learning (ML).
Definition:
ML focuses on the development of algorithms that improve automatically
through experience. It includes methods like supervised learning,
reinforcement learning, and natural language processing.It does NOT
include studies focused only on statistical modeling, data visualization,
or database management unless they involve ML-specific techniques.
```

CLS Zero Shot Prompt for GPT-4o and o3

```
Task:
1. Classify the text as 1 (CLS) or 0 (Not CLS).
2. Provide a brief explanation justifying your decision.
Text:
"abstract"
Return your answer *exactly* in this format:
CLS Score: [1 or 0]
CLS Explanation: [Brief explanation in one paragraph, no Markdown or
formatting]
```

ML Zero Shot Prompt for GPT-4o and o3

```
Task:
1. Classify the text as 1 (ML) or 0 (Not ML).
2. Provide a brief explanation justifying your decision.
Text:
"abstract"
Return your answer *exactly* in this format:
ML Score: [1 or 0]
ML Explanation: [Brief explanation in one paragraph, no Markdown or
formatting]
```

CLS Few Shot Prompt for GPT-4o and o3

```
Task:
1. Classify the text as 1 (CLS) or 0 (Not CLS).
2. Provide a brief explanation justifying your decision.
Text:
"abstract"
Return your answer *exactly* in this format:
CLS Score: [1 or 0]
CLS Explanation: [Brief explanation in one paragraph, no Markdown or formatting]
**Examples:**
Input:
"Poem generation with language models requires the modeling of rhyming patterns. We propose a novel solution for learning to rhyme, based on synthetic data generated with a rule-based rhyming algorithm."
Output:
1
"Computational methods are applied to a literary text (poem generation)."
Input:
"This paper explores the capacity of computer vision models to discern temporal information in visual content, focusing specifically on historical photographs."
Output:
0
"The study is centered on historical photographs, not literary texts."
```

ML Few Shot Prompt for GPT-4o and o3

```
Task:
1. Classify the text as 1 (ML) or 0 (Not ML).
2. Provide a brief explanation justifying your decision.
Text:
"abstract"
Return your answer *exactly* in this format:
ML Score: [1 or 0]
ML Explanation: [Brief explanation in one paragraph, no Markdown or
formatting]
**Examples:**
Input:
"We fine-tune a GPT-2 English model with 124M parameters on 142 MB of
natural poems and find that this model generates consecutive rhymes
infrequently (11
Output:
1
"The study involves fine-tuning a language model, which is a machine
learning method."
Input:
"We find a statistically significant correlation between violent
discourse and emotional expression throughout the analyzed period."
Output:
0
"The study only mentions a focus on statistical correlation, not
necessarily machine learning methods."
```

CLS Step-by-step Prompt for GPT-4o and o3

```
Task:
Analyze the text below step by step:
1. Identify computational methods used if any.
2. Determine if the text focuses on literary texts.
3. Classify the text a CLS (1) or Not CLS (0) based on your findings.
4. Provide a brief explanation justifying your decision.
Text:
"abstract"
Return your answer *exactly* in this format:
CLS Score: [1 or 0]
CLS Explanation: [Brief explanation in one paragraph, no Markdown or
formatting]
```

ML Step-by-step Prompt for GPT-4o and o3

```
Task:
Analyze the text below step by step:
1. Identify machine learning methods used if any.
2. Consider whether the methods used are explicitly ML and not rule-based.
3. Classify the text as ML (1) or Not ML (0) based on your findings.
4. Provide a brief explanation justifying your decision.
Text:
"abstract"
Return your answer *exactly* in this format:
ML Score: [1 or 0]
ML Explanation: [Brief explanation in one paragraph, no Markdown or formatting]
```

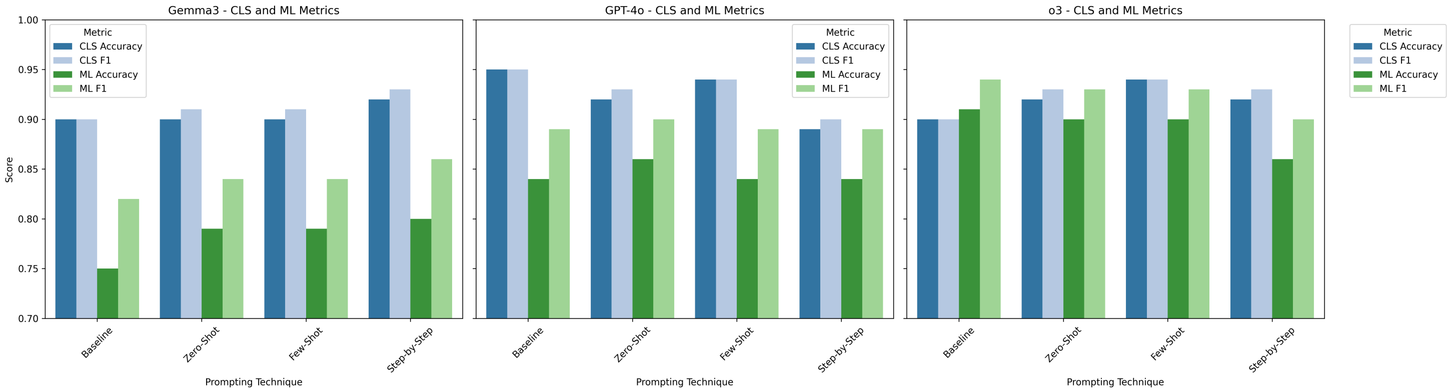

Figure A.1: Prompting Techniques compared for different models on the original CLS dataset using accuracy and F1-scores

References

Shubham Agarwal, Issam H. Laradji, Laurent Charlin, and Christopher Pal. 2024. LitLLM: A Toolkit for Scientific Literature Review. (Cited on page 9).

Shubham Agarwal, Gaurav Sahu, Abhay Puri, Issam H. Laradji, Krishnamurthy DJ Dvijotham, Jason Stanley, Laurent Charlin, and Christopher Pal. 2025. LitLLMs, LLMs for Literature Review: Are we there yet? (Cited on page 9).

Allen Institute for AI. 2025. Introducing Ai2 ScholarQA. AI2 Blog, January. (Cited on pages 10, 49).

arXiv Monthly Submissions. 2025. https://arxiv.org/stats/monthly_submissions. (Cited on page 1).

Bizzoni, Yuri, Degaetano-Ortlieb, Stefania, Kazantseva, Anna, and Szpakowicz, Stan, eds. 2024. Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024). St. Julians, Malta: Association for Computational Linguistics, March. (Cited on page 13).

Louis-François Bouchard and Louie Peters. 2024. Building LLMs for Production: Enhancing LLM Abilities and Reliability with Prompting, Fine-Tuning, and RAG. Independently published. (Cited on pages 6 sqq.).

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A Dataset of Information-Seeking Questions and Answers Anchored in Research Papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* edited by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, 4599–4610. Online: Association for Computational Linguistics, June. (Cited on pages 10, 50).

Deepseek LLM 7B Chat - GPTQ (Hugging Face). 2025. (Cited on page 19).

DeepSeek-AI, : Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. 2024. DeepSeek LLM: Scaling Open-Source Language Models with Longtermism. (Cited on page 5).

Degaetano, Stefania, Kazantseva, Anna, Reiter, Nils, and Szpakowicz, Stan, eds. 2022. Proceedings of the 6th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature. Gyeongju, Republic of Korea: International Conference on Computational Linguistics, October. (Cited on page 12).

Degaetano-Ortlieb, Stefania, Kazantseva, Anna, Reiter, Nils, and Szpakowicz, Stan, eds. 2023. Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature. Dubrovnik, Croatia: Association for Computational Linguistics, May. (Cited on page 13).

Shayan Doroudi. 2023. What is a related work? A typology of relationships in research literature. *Synthese* 201 (24). (Cited on page 1).

Lizhou Fan, Lingyao Li, Zihui Ma, Sanggyu Lee, Huizi Yu, and Libby Hemphill. 2024. A Bibliometric Review of Large Language Models Research from 2017 to 2023. *ACM Trans. Intell. Syst. Technol.* (New York, USA) 15, no. 5 (October). (Cited on page 3).

Yeow Chong Goh, Xin Qing Cai, Walter Theseira, Giovanni Ko, and Khiam Aik Khor. 2020. Evaluating human versus machine learning performance in classifying research abstracts - scientometrics, July. (Cited on pages 10, 50).

Google. 2023. Introducing Gemini: our largest and most capable AI model. https://blog.google/technology/ai/google-gemini-ai/. (Cited on page 5).

Hans Ole Hatzel, Haimo Stiemer, Chris Biemann, and Evelyn Gius. 2023. Machine learning in computational literary studies. *it - Information Technology* 65 (4-5): 200–217. (Cited on pages i, 10, 12 sqq., 29 sqq.).

Sangwoo Heo, Sungwook Son, and Hyunwoo Park. 2025. HaluCheck: Explainable and verifiable automation for detecting hallucinations in LLM responses. *Expert Systems with Applications* 272:126712. (Cited on page 49).

Lucas Joos, Daniel A. Keim, and Maximilian T. Fischer. 2024. Cutting Through the Clutter: The Potential of LLMs for Efficient Filtration in Systematic Literature Reviews. (Cited on page 9).

Journal of Computational Literary Studies: Volume 1 - Issue 1 - 2022. 2022. https://jcls.io/issue/84/info/. (Cited on page 12).

Journal of Computational Literary Studies: Volume 2 - Issue 1 - 2023. 2023. https://jcls.io/issue/105/info/. (Cited on page 13).

Journal of Computational Literary Studies: Volume 3 - Issue 1 - 2024. 2024. https://jcls.io/issue/109/info/. (Cited on page 13).

Karsdorp, Folgert and Nielbo, Kristoffer L., eds. 2022. Proceedings of the Computational Humanities Research Conference 2022, CHR 2022, Antwerp, Belgium, December 12-14, 2022. Vol. 3290. CEUR Workshop Proceedings. CEUR-WS.org. (Cited on page 12).

Erik Ketzan and Nicolas Werner. 2022. 'Entrez!' she called: Evaluating Language Identification Tools in English Literary Texts. In *Proceedings of the Computational Humanities Research Conference (CHR 2022),* edited by Folgert Karsdorp and Kristoffer L. Nielbo, 3290:366–373. CEUR Workshop Proceedings. Antwerp, Belgium: CEUR-WS.org, December. (Cited on page 32).

Marijn Koolen, Julia Neugarten, and Peter Boot. 2022. “This book makes me happy and sad and I love it”. A Rule-based Model for Extracting Reading Impact from English Book Reviews. *Journal of Computational Literary Studies* 1 (1). (Cited on page 32).

LiveBench. 2024. LiveBench: A Challenging, Contamination-Free LLM Benchmark. https://livebench.ai/. (Cited on page 6).

LMArena. 2025. LMArena: Open Platform for Crowdsourced AI Benchmarking. https://lmarena.ai/. (Cited on page 6).

Derek Lowe. 2025. An Evaluation of “Deep Research” Performance. Science Blog (In the Pipeline), AAAS. (Cited on pages 10, 49).

Meta. 2023. Llama-2-7B-chat-hf (Hugging Face). (Cited on page 18).

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2025. Large Language Models: A Survey. arXiv: 2402.06196 `[cs.CL]`. (Cited on page 3).

Aníbal Monasterio Astobiza. 2025. The role of LLMs in theory building (May). (Cited on page 3).

OpenAI. 2024. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/. (Cited on page 6).

———. 2025a. Introducing deep research, February. https://openai.com/index/introducing-deep-research/. (Cited on pages 10, 49).

———. 2025b. Introducing OpenAI o3 and o4-mini. https://openai.com/index/introducing-o3-and-o4-mini/. (Cited on page 6).

———. 2025c. OpenAI o3 and o4-mini System Card, April. https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf. (Cited on pages 10, 49).

Proceedings of the Computational Humanities Research Conference 2024, Aarhus, Denmark, December 4-6, 2024. 2024. Vol. 3834. CEUR Workshop Proceedings. CEUR-WS.org. (Cited on page 13).

Sapling. 2024. Gemma vs. LLaMA: Which LLM is Better? https://sapling.ai/llm/gemma-vs-llama. (Cited on page 6).

Dmitry Scherbakov, Nina Hubig, Vinita Jansari, Alexander Bakumenko, and Leslie A. Lenert. 2024. The emergence of Large Language Models (LLM) as a tool in literature reviews: an LLM automated systematic review. (Cited on page 9).

Sela, Artjoms, Jannidis, Fotis, and Romanowska, Iza, eds. 2023. Proceedings of the Computational Humanities Research Conference 2023, Paris, France, December 6-8, 2023. Vol. 3558. CEUR Workshop Proceedings. CEUR-WS.org. (Cited on page 13).

Sentence Transformers (Hugging Face). 2025. all-MiniLM-L6-v2. (Cited on page 29).

Özge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. 2022. Neural entity linking: A survey of models based on deep learning. *Semantic Web* 13 (3): 527–570. eprint: https://journals.sagepub.com/doi/pdf/10.3233/SW-222986. (Cited on pages 10, 14, 35).

Sakib Shahriar, Brady Lund, Nishith Reddy Mannuru, Muhammad A Arshad, Kadhim Hayawi, Ravi Varma Kumar Bevara, Aashrith Mannuru, and Laiba Batool. 2024. Putting GPT-4o to the Sword: A Comprehensive Evaluation of Language, Vision, Speech, and Multimodal Proficiency, June. (Cited on page 23).

Degaetano-Ortlieb, Stefania, Kazantseva, Anna, Reiter, Nils, and Szpakowicz, Stan, eds. 2021. Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature. Punta Cana, Dominican Republic (online): Association for Computational Linguistics, November. (Cited on page 12).

Alison Smith, Varun Kumar, Jordan Boyd-Graber, Kevin Seppi, and Leah Findlater. 2018. User-Centered Design and Evaluation of a Human-in-the-Loop Topic Modeling System. In *Intelligent User Interfaces.* (Cited on page 10).

Don R. Swanson. 1986. Undiscovered Public Knowledge. *The Library Quarterly* 56 (2): 103–118. (Cited on page 1).

Xuemei Tang, Xufeng Duan, and Zhenguang G. Cai. 2025. Are LLMs Good Literature Review Writers? Evaluating the Literature Review Writing Ability of Large Language Models. (Cited on page 9).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models, February. (Cited on page 4).

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. (Cited on page 4).

Zijie J. Wang, Dongjin Choi, Shenyu Xu, and Diyi Yang. 2021. Putting Humans in the Natural Language Processing Loop: A Survey. In *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing,* edited by Su Lin Blodgett, Michael Madaio, Brendan O'Connor, Hanna Wallach, and Qian Yang, 47–52. Online: Association for Computational Linguistics, April. (Cited on page 9).

Jiawei Zhang, Chejian Xu, Yu Gai, Freddy Lecue, Dawn Song, and Bo Li. 2024. KnowHalu: Hallucination Detection via Multi-Form Knowledge Based Factual Checking. (Cited on page 49).

Qian Zhou, Meng Li, Jiawei Zhan, Kai Xu, and Lianwen Chen. 2021. Automatic Article Classification Model for Systematic Review Using BERT. *Systematic Reviews* 10 (1): 1–13. (Cited on pages 10, 17).

Affidavit

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Bachelorstudiengang Software-System-Entwicklung selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe. Sofern im Zuge der Erstellung der vorliegenden Abschlussarbeit generative Künstliche Intelligenz (gKI) basierte elektronische Hilfsmittel verwendet wurden, versichere ich, dass meine eigene Leistung im Vordergrund stand und dass eine vollständige Dokumentation aller verwendeten Hilfsmittel gemäß der Guten Wissenschaftlichen Praxis vorliegt. Ich trage die Verantwortung für eventuell durch die gKI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate.

19.08.25

Date

Liamdebus

Signature

(Liam Debus)

Schneverdingen

Ort