Extraction of tabular data from annual reports with LLMs

Using in context learning with open source models and RAG

submitted by

Simon Schäfer

Matr.-Nr.: 944 521

Department VI – Informatics and Media
Berliner Hochschule für Technik Berlin
presented Master Thesis
to aquiere the academic degree

Master of Science (M.Sc.)

in the field of

Data Science

Date of submission September 1, 2025

Gutachter

Prof. Dr. Alexander Löser	Berliner Hochschule für Technik
Prof. Dr. Felix Gers	Berliner Hochschule für Technik

Berliner Hochschule für Technik

Studiere Zukunft

Abstract

This thesis presents a comprehensive benchmark of information extraction from annual reports of German companies, focusing on the use of large language models (LLMs) to automate the retrieval of financial data from unstructured PDF documents. The study addresses two core challenges: identifying relevant pages containing financial tables and accurately extracting hierarchical tabular data for integration into relational databases.

Multiple open-weight LLMs are evaluated using diverse prompting strategies and compared to alternative methods such as regular expressions and term-frequency approaches. The work builds on and extends previous research by applying these techniques to German-language documents and open-weight models.

Results demonstrate that (even smaller) LLMs can reliably locate and extract financial information, supporting efficient audit processes. The findings highlight the potential of combining LLMs with rule-based methods to improve computational efficiency, and outline a practical workflow for a human-in-the-loop application for information extraction in regulatory and financial domains.

Zusammenfassung

Diese Arbeit präsentiert die Ergebnisse eines umfassenden Benchmarks von Methoden zur Extraktion von Informationen aus deutschen Jahresabschlussberichten. Wir legen den Fokus unserer Untersuchung auf die Nutzung von LLMs (*große Sprachmodelle*) für die Automatisierung des Auslesens von Finanzdaten aus unstrukturierten PDF (Portable Document Format) Dokumenten.

Die Arbeit verfolgt zwei Hauptziele: das Identifizieren relevanter Seiten und das Extrahieren numerischer Werte aus hierarchischen tabellenartigen Strukturen. Dabei wird ein Format der extrahierten Daten sichergestellt, das das Einlesen in eine relationale Datenbank erlaubt.

Die Performanz mehrerer open-weight LLM (large language model)s wird über verschiedene Prompting-Strategien hinweg und mit alternativen Methoden, beispielsweise reguläre Ausdrücke und Suchwortdichte-Maßen, verglichen. Die Arbeit baut auf vorherige Forschungsergebnisse auf und erweitert diese um die Anwendung auf deutschsprachigen Dokumenten und die Nutzung von open-weight Modellen.

Die Ergebnisse zeigen, dass (auch kleinere) Sprachmodelle zuverlässlich Kennzahlen aus dem Finanzwesen lokalisieren und extrahieren und so die Prüfungstätigkeiten effizient unterstützen können. Die Ergebnisse zeigen Potential auf, LLMs mit regelbasierten Methoden zu kombinieren, um die Effizienz der Informationslokalisation zu erhöhen. Wir skizzieren ein HITL (human-in-the-loop)-basiertes Software-System für die Informationsextraktion im Bereich der Finanzkontrolle.

Reading advices

We recommend to read the thesis in its digital gitbook version instead of the PDF version. Furthermore, the author recommends to read the thesis (any version) on a screen that is larger than 21” and has at least full HD resolution. The more, the merrier.

😊

Declaration of the Use of Artificial Intelligence

We utilized GitHub Copilot in Visual Studio Code to assist with coding tasks, primarily employing GPT-4.1 and occasionally Claude Sonnet 4.

In addition to traditional literature research methods, such as using Google Scholar and tracking citations in academic papers, we also employed perplexity.ai to support our literature review process.

Additionally, GPT-4.1 was used to enhance the clarity and quality of language throughout the thesis and to generate suggestions for summaries for various sections.

Personal Goals and Learnings

Several key goals were accomplished during this thesis. Notably, I succeeded in producing the thesis with bookdown in both PDF and HTML formats—a milestone I had already set during my previous PhD attempt in 2019. I also became familiar with creating Docker images and orchestrating jobs within a Kubernetes cluster. I gained practical experience in using large language models (LLMs), including leveraging restricted decoding to generate results suitable for downstream tasks.

There were, however, some areas I was interested in exploring further but could not pursue due to time constraints. These include learning to set up and administer a Kubernetes cluster independently, fine-tuning an LLM model, training a small encoder-only model from scratch, and applying visual LLMs for document parsing.

Dedication

We would like to express our sincere gratitude to Tom Röhr, whose unwavering support and insightful feedback have been invaluable throughout the development of this thesis.

We would like to express our sincere gratitude to Micha for his valuable support in extending the ground truth dataset for the information extraction task.

We also extend our heartfelt thanks to our family and friends for their unwavering support and patience throughout the past months.

Contents

Chapter 1

Introduction

Information has always been generated, processed, and shared by humans. Initially, this sharing occurred synchronously through spoken language. Over time, humans developed ways to store information - first on clay tablets, then on paper, and most recently in digital files (Bentley, 2025). As a result, the human knowledge base has grown steadily, document by document. The field of library and information science emerged to organize this expanding body of information and enable efficient access. Since information was initially exchanged exclusively between humans, storage formats were designed to suit human perception and understanding.

However, several fundamental changes have occurred. First, the rate of information generation is increasing rapidly, and the volume of relevant knowledge now grows faster than humans can absorb (Chamberlain, 2020). In science alone, each year brings forth more new information than any individual could possibly read (Hong et al., 2021). Fortunately, recent advances in LLM (large language model)s provide tools to compress and summarize this information before it reaches the reader.

At the same time, generative AI (artificial intelligence) is accelerating the creation of new information. Today, a single sentence in natural language can initiate the generation of an entire website or book. Recently, AI agents have been developed that can autonomously react to digital or real-world triggers and generate content automatically. Combined with data produced by Internet of Things devices, the prediction that information could double every eleven hours (IBM Global Technology Services, 2006) may already be a reality. This marks a second major shift: algorithms now generate, process, and share information as well.

Despite these changes, much published information remains optimized for human consumption, such as in PDF documents. Algorithms can process information efficiently, but only if it is machine-readable. Since manual encoding by humans is inefficient and error-prone, the field of information retrieval has emerged to address this challenge. LLMs can assist in retrieving information - even in structured formats - that can be used by other algorithms in downstream tasks.

For older sources, this approach is often the only alternative. For future information sharing, however, a more direct solution exists: providing data in a machine-readable format from the outset. Otherwise, we continue to face the “Last Mile Problem” (H. Li, Wei, et al., 2023). Since the format of received data can only be changed for information one owns, users must cope with the formats provided until the need for machine-readable data is recognized and addressed by data owners.

Employees at RHvB (Rechnungshof von Berlin) typically work with data supplied by other administrations and companies, which is rarely delivered in a machine-readable format. Therefore, we explore strategies to address the “Last Mile Problem” until all stakeholders in this process transition to modern data practices.

Section 1.1 describes how the volume and format of information provided to the RHvB present challenges for the audit process. Section 1.2 specifies these challenges as concrete real-world problems, from which we derive our research questions. Section 1.3 provides an overview of our methodology, and the thesis outline is presented in section 1.4.

1

1.1 Motivation

In recent decades, the pace of digital transformation has accelerated, and “electronic documents have increasingly supplanted paper documents as the primary medium for information exchange across various industries” (Q. Zhang et al., 2024, p. 1). In the finance industry as well, large volumes of information are stored in unstructured digital formats, such as PDF files (H. Li, Gao, et al., 2023). This reliance on unstructured data not only impedes investment decisions (El-Haj et al., 2020) and academic research (Jr et al., 2015), but also regulatory processes (H. Li, Wei, et al., 2023).

Regulatory processes may be initiated or informed by audit reports produced by the RHvB. The RHvB contributes to the transparent use of public funds in Berlin by auditing public administrations and companies in which the state of Berlin is a shareholder or which receive public funding. These audits help prevent corruption and ensure efficient allocation of resources.

During the audit process, employees at the RHvB must handle large amounts of information embedded in PDF documents. Some of these are born-digital files, while others are simply scanned paper pages. For example, in the audit of the UEFA football championship, gigabytes of data are received. Extracting the information necessary for the audit is a significant challenge and would require substantial human resources. Algorithmic assistance is therefore highly desirable.

Because the focus of each audit changes frequently, traditional rule-based approaches offer limited benefits when considering the time required to program such systems versus their actual usage. Leveraging the flexible automation capabilities of “programming by example” with LLMs (W.-D. Li & Ellis, 2024) appears promising. The need for automation is further heightened by the impending shortage of experienced staff due to a wave of retirements.

1.2 Objectives

The sixth division of the RHvB is responsible for auditing companies in which Berlin is a shareholder (see Figure 1.1. As part of this process, they must analyze balance sheets and profit and loss statements, which are fundamental sources of information. This information is provided through the companies’ annual reports, typically in the form of PDF files. Automating the extraction of this information would be an excellent starting point for AI-assisted information retrieval from PDFs at the RHvB. This task is particularly well-suited for a thesis, as exporting financial data from the 59 companies is a recurring requirement.

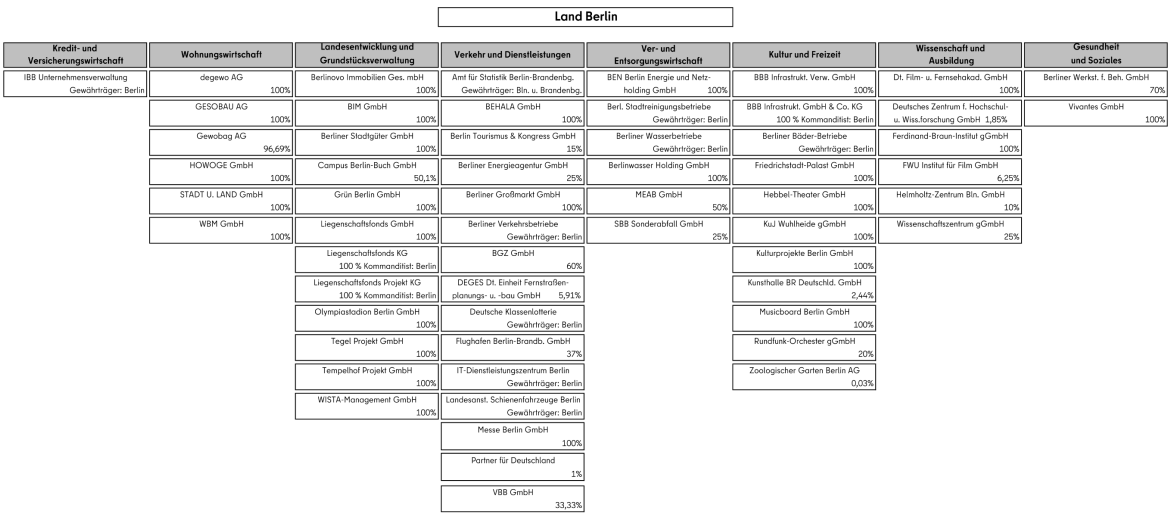


Figure 1.1: Overview of the 59 companies in which Berlin is a shareholder, including the percentage ownership for each company.

The annual reports provided often differ from publicly available versions in terms of information granularity and design, and are treated as non-public information. For this thesis we use the publicly available versions to allow a comparison of open-weight models with OpenAIs GPT (generative pre-trained transformers) models. We focus on open-weight models because the information in the non-public reports may be confidential. The “Berliner IKT-Richtlinie” prohibits processing such information in public clouds and encourages the use of open-source solutions.

For this thesis, we narrow the broad field of information retrieval to the extraction of the assets table, which is part of the balance sheet. To achieve a high degree of automation, we also investigate the possibility of detecting the table without requiring the user to specify the page number or area. We formulate our two main research questions:

Q1 How can we use LLMs effectively to locate specific information in a financial report?

Q2 How can we use LLMs effectively to extract this information from the document?

Since the results of this thesis will be used to create an application with a HITL (human-in-the-loop) approach, we also investigate an additional research question. The concept for the planned application can be seen in the Appendix E.1. The user should review the information extraction results and resolve issues that the system alone cannot handle. However, redundant double work should be minimized. Therefore, we formulate our third research question:

Q4 Can we use additional information from the extraction process to guide the user on which values need to be checked and which can be trusted as they are?

The following section briefly describes our methodology for investigating these research questions. The corresponding hypotheses are formulated in section 3.2.1.

1.3 Methodology

This thesis aims to provide recommendations on how to best solve the described extraction task. Accordingly, it is situated within the field of applied research. We benchmark a broad variety of approaches and conduct experiments to identify general predictors of task performance. Our investigation begins by implementing the framework described by H. Li, Gao, et al. (2023). Figure 1.2 illustrates the two stages they propose, which align with our main research questions.

First, we examine different approaches to identifying the page that contains the target information and how to combine these methods efficiently. In addition to a regex (regular expression)-based and a LLM-driven TOC (table of contents) approach, we test a LLM-driven classification method as well as a term-frequency-based ranking approach.

Second, we evaluate whether LLMs can effectively extract the target information using the appropriate prompting strategy. We extend H. Li, Gao, et al. (2023) work by testing how well LLMs can extract multiple values in a single prompt and by designing experiments to measure the effects of various influencing factors. Beyond evaluating model- and prompt-specific predictors, we also systematically examine how features of the tabular structure influence information extraction performance.

Furthermore, we test the upper limits of extraction performance using a synthetic dataset that is free of unknown target row identifiers. This allows us to determine whether a simple text extract is sufficient as input, or if additional effort - such as document layout analysis or specialized table extraction techniques - is required to accurately extract the structure of the assets table.

Our work also addresses whether the presented framework is effective for more heterogeneous documents, how open-weight models perform, and whether the German language of the annual reports presents unique challenges.

1.4 Thesis Outline

Chapter 2 provides an overview of the theoretical background and references key literature relevant to the concepts used in this thesis.

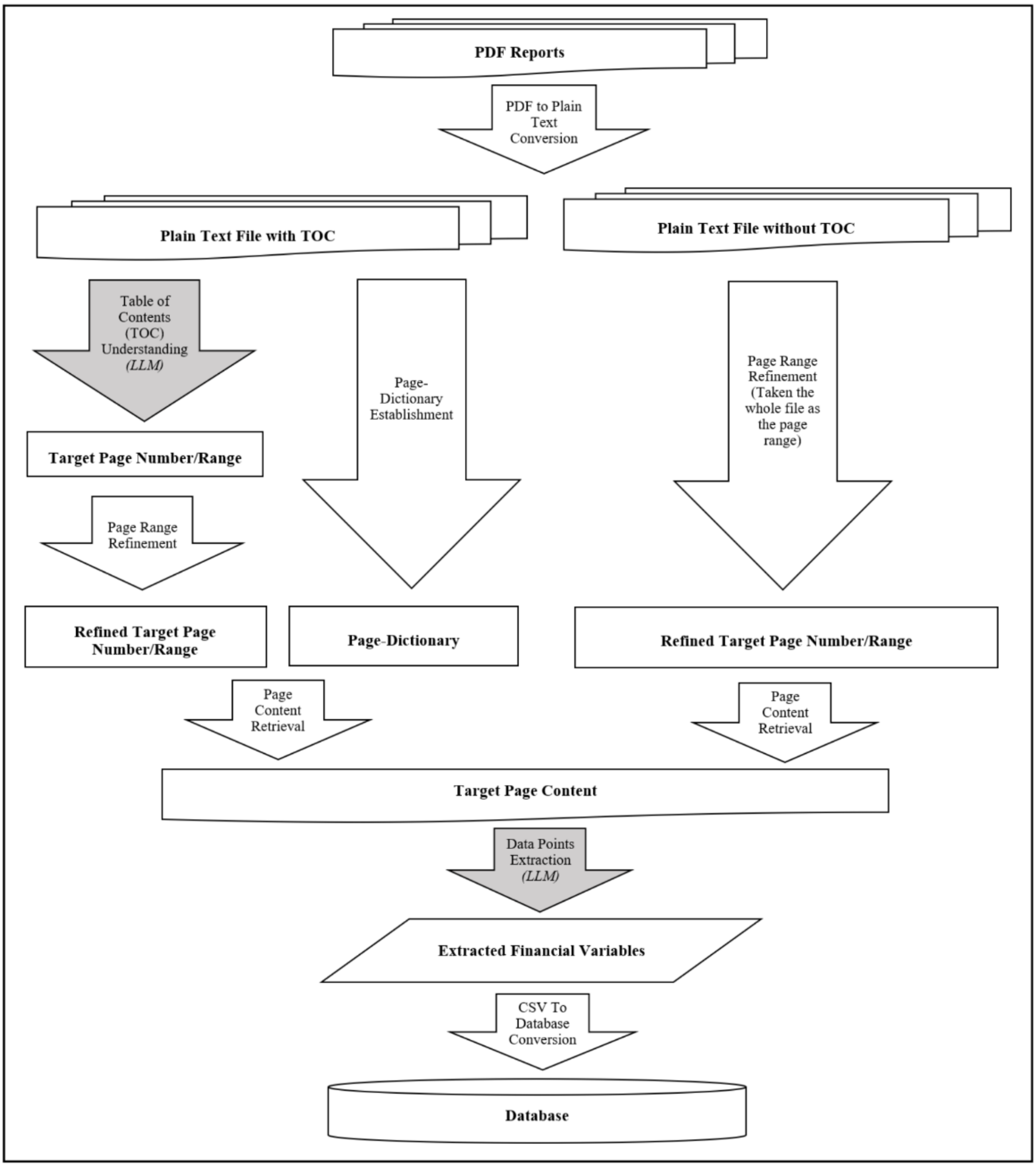


Figure 1.2: Flowchart of the extraction framework of H. Li, Gao, et al. (2023)

Chapter 3 outlines the research design, research questions, and hypotheses. For each of the three main research questions, it details the evaluation and data strategies, and provides an overview of the experimental setup, including the evaluation methods and expected error types.

Chapter 4 describes the hardware and software used for the experiments. It also presents a flow chart and a detailed description of the data processing workflow.

Chapter 5 summarizes the findings for the three main research questions. Detailed explanations of how the results were obtained are provided in the appendix.

Chapter 6 interprets the results in relation to the research questions and hypotheses. It includes an error analysis, discusses the limitations of the study, and outlines areas not covered. The chapter also provides an outlook on how the results can be applied to address real-world problems.

Chapter 7 summarizes the answers to the research questions.

1.5 Summary

This chapter outlined the challenges posed by the ever-increasing volume of information and the obstacles that non-machine-readable data present for algorithmic processing. It described the specific problem addressed in this thesis: extracting financial information from annual reports to support the audit processes at RHvB. We formulated our research questions, outlined our methodology for investigating them, and provided an overview of the subsequent chapters.

1

Chapter 2

Literature review

This chapter presents all literature relevant to our research questions and provides essential background on the concepts and methods employed in this thesis. As outlined in Chapter 1, the problem addressed here lies within the field of information retrieval and is approached using NLP (natural language processing) techniques. Accordingly, Section 2.1 reviews techniques for locating information within and extracting it from documents. Subsection 2.1.4 then explains the mechanisms and architectures of modern LLM (large language model)s, with particular attention to the MoE (mixture of experts) architecture.

Subsequently, Subsection 2.1.5 discusses the prompting strategy *in-context learning*, which leverages the "programming by example" paradigm, and explores how RAG (retrieval augmanted generation) integrates into this framework. We also demonstrate how guided decoding can be used to generate structured responses suitable for downstream tasks.

Section 2.2 introduces the SHAP (SHapley Additive exPlanations) framework, a unified model for explaining machine learning predictions, applicable to complex models such as deep neural networks and random forests. The latter are briefly described as well. We employ random forests and SHAP to test our hypotheses regarding potential predictors for the information extraction task. Our hypotheses are stated in Section 3.2.1.

2.1 Computer science

Information retrieval and NLP are important areas in computer science, enabling the efficient extraction and utilization of information from vast datasets (Raj & Mishra, 2025). We will describe both in the following subsections.

2.1.1 Information retrieval

Information retrieval deals with finding relevant information within large collections of unstructured text. It is used in search engines, of dialogue, question-answering, and recommender systems (Zhu et al., 2024).

The term frequency $\mathrm{tf}_{t,d}$ is one of the oldest measures used in retrieval algorithms. It just counts the number of occurrences of a term in a document. Document is an abstraction in this case. It can be a sentence, a page or a file. Since longer documents might have higher term frequency for each term, it is useful to normalize the value by the document length $|d|$. This measure could be called term rate:

$$\mathrm{tr}_t = \frac{\mathrm{tf}_{t,d}}{|d|} \tag{2.1}$$

It is part of well established measures as TF-IDF (Frequency-Inverse Document Frequency) and Okapi BM25 (best matching 25). Both are used for ranking, how relevant a document is for a given search query and are widely used in information retrieval systems (S. Robertson, 2004; S. Robertson & Zaragoza, 2009) and thus can be part of a RAG architecture too. BM25 is one of the "most successful Web-search and corporate-search algorithms" (S. Robertson & Zaragoza, 2009, p. 1).

The IDF (Inverse Document Frequency) is often used as a weighting function. If the ranking of possible results of a search query is simply calculated as sum of all term frequencies in a document, that are present in the query as well less informative terms get equal weight.

Looking at the search query: "Is the positron blue?", helps to illustrate the problem. The terms *is*, *the* and *blue* might be present often in a document for children that is talking about the sky or sea. Such a document could get high score, even though *positron* is never mentioned. It would be good, if it is most important if the term *positron* is in the document. We can achieve this by multiplying all term frequencies with the IDF score (Manning et al., 2008, p. 118):

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t} \tag{2.2}$$

N is the number of documents in the collection of documents and df_t the number of documents, that contain term t. While the term frequencies $\mathrm{tf}_{t,d}$ are calculated separate for each document, the IDF score is computed once for the whole collection. The TF-IDF score is then defined by:

$$\text{tf-idf}_t = \mathrm{idf}_t \cdot \mathrm{tf}_{t,d} \tag{2.3}$$

The more advanced measure BM25 is derived in Manning et al. (2008). It introduce a saturation term, that prevents a term occurring very often in a document will increasing the score linearly. The 25 in BM25 simply refers to the fact, that it is the 25th version of the Okapi system, that was developed at City University London (S. E. Robertson & Walker, 1994; S. Robertson & Zaragoza, 2009).

Measures as TF-IDF or BM25 are also used for classification tasks, i.e. in the context of sentiment analysis (Carvalho & Guedes, 2020) and semantic understanding (Rathi & Mustafi, 2023). They can be used for RAG and are called sparse retrievers in this context.

2.1.2 Natural language processing

NLP (natural language processing) is a field of artificial intelligence that focuses on enabling computers to understand, interpret, and generate human language (Mahmudi, 22 Dec 2024 6:46pm). NLP encompasses a wide range of tasks, including sentiment analysis, information extraction, table and dialog understanding, as well as summarization, code generation and machine translation (Qin et al., 2025).

Methods of NLP can enhance the process of information retrieval. Vector embedding based techniques emerged recently and are called dense retrievers (Zhu et al., 2024).

In the context of this thesis, information retrieval techniques are applied to locate financial information from annual reports. Techniques for information extraction are used to generate responses formatted for downstream tasks.

2.1.3 Text processing

Many information extraction tasks target information that is found in natural texts or tables. In order to use NLP techniques it is a prerequisite to extract those texts. Adhikari & Agarwal (2024) show in a benchmark study that *pypdfium* and *PyMuPDF* perform best among rule-based systems. But the transformer-based vision and document understanding tool *Nougat* (Blecher et al., 2023) performs even better.

Adhikari & Agarwal (2024) also benchmark different table extraction tools. They find, that the transformer-based *Table Transformer* (Smock et al., 2022) extracts tables most accurate in most cases. Rule-based tools are more accurate for table extraction tasks from tenders and manuals.

Sometimes there is only a visual representation of texts, resulting from scanned pages. In this case OCR (optical character recognition) techniques can be used, to transform the visual information back into a text representation. A comparison study for recent OCR solutions can be found at Francis & Sangeetha (2025).

2.1.3.1 Document Layout Analysis

An important step in the process of extracting information from complex documents is to recognize the layout of a document (Zhong et al., 2019). Common tasks are determining the correct order of texts, correctly align captions to tables and figures, identifying headings, tables and figures. This can be used to separate the contents of smaller tables, that contain the information of interest, from surrounding text and thus strip potentially distracting content. Paul (2025) gives a broad overview for state-of-the-art document layout analysis architectures and presents a comparison of different models performance.

2.1.4 Large Language Models

Minaee et al. (2025) give a broad overview on the field of language models. They discuss, how they are built, used and augmented. They present popular datasets, performance benchmarks and outline challenges and future directions. Their survey is neither limited on decoder-only nor large language models (\acr{LLM}). But their time scope ends before 2024, so recent advancements are not reflected. (not a good introduction survey)

For readers, who want to build understanding for modern LLM we recommend to work through the book "Dive into Deep Learning" (A. Zhang et al., 2023).

2.1.4.1 Transformers

Wichtig

seit 2017

training on token prediction (base models); masking,

2.1.4.2 Attention

The most obvious challenge is computational cost. The amount of processing power required scales quadratically with the length of the input (Tahir, 2025).

Sliding window attention (e.g., as popularized by Mistral) or Gemma 2; Sliding window attention is mainly used to improve computational performance (Raschka, 2025)

Group-query attention (like in Llama 2 and 3)

A key article is "Attention Is All You Need" that (Vaswani et al., 2023)

2.1.4.3 Encoder

Wichtig

positional encoding important (and distinguishes from tf-idf): dog eats cat

sinusoidal positional encoding, which uses sine and cosine functions of varying frequencies to create unique positional vectors, and Rotary Position Embedding (RoPE), which applies a rotation to the token embeddings based on their position (Khowaja, 2025)

2.1.4.4 Decoder

For each generated token, the attention mechanism needs to access the key and value vectors of all preceding tokens in the context window. To avoid recomputing these key and value vectors at each step, they are stored in the KV-cache. (Khowaja, 2025) However, the memory required to store the KV-cache scales linearly with the size of the context window.

Wichtig

Token sampling, temperature 0

2

2.1.4.5 GPT (Generative Pretrained Transformers)

hauptsächlich decoder (generieren)

Wichtig

2.1.4.6 Mixture of Experts

Recent LLM (large language model)s often use a MoE architecture. The models of Llama 4, Qwen3 and GPT-4.1 are prominent examples for this kind of LLMs. D. Zhang et al. (2025) and Cai et al. (2025) give an exhaustive overview of different types of MoE architectures. While D. Zhang et al. (2025) lists also models released this year and shows some applications of MoE, is Cai et al. (2025) discussing different architecture types in more detail. Grootendorst (2024) gives a guid to MoE with many helpful illustrations.

The basic idea of MoE models is to combine multiple smaller, specialized FFN (feed forward network)s to achieve better predictions overall. The MoE "paradigm offers a compelling method to significantly expand model capacity while avoiding a corresponding surge in computational demands during training and inference phases" (Cai et al., 2025, p. 21).

Figure 2.1 shows two main differences in the architecture. One one hand there is the dense (a) architecture. Here, each token is fed into every FFN and all results are pooled. On the other hand, there is the sparse architecture. Here, each token is just fed into a subset of FFNs. Dense MoE models often yield higher prediction accuracy, but also significantly increase the computational overhead (Cai et al., 2025).

The gate (also router) takes care of the distribution of tokens to the FFNs. There is a high diversity of the routing algorithms and its goals are to "ensure expert diversity while minimizing redundant computation" (D. Zhang et al., 2025). There are algorithms that focus on load-balancing, domain specific routing and many more. Traditional MoE assumes homogeneous experts, where load balancing might be the paramount goal. Recent advances explore more heterogeneous sets of experts and flexible routing strategies, that promise more efficiency (D. Zhang et al., 2025).

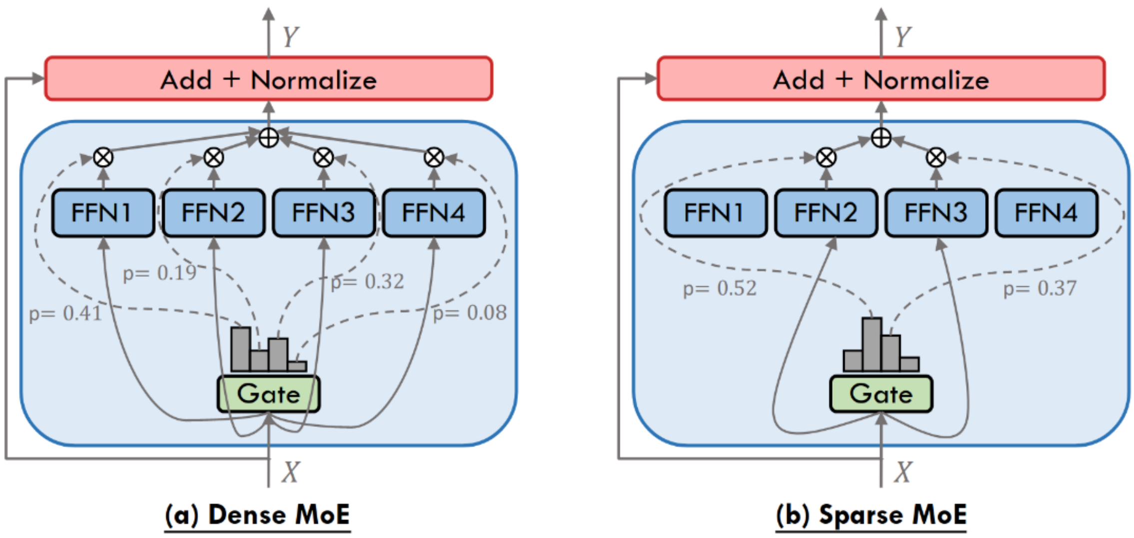


Figure 2.1: Showing schemas of the dense and sparse mixture of experts architecture.

Most of the Qwen3 models have a dense MoE architecture. Only the two models released in July 2025 have a sparse architecture. These models have two parameter specifications. For example Qwen3-235B-A22B is specifying that the model has 235B token in total. But per token processed it uses (activates) just 22B parameters. In their mixture of experts architecture this means that 8 of 128 experts are participating in processing each token.

The Llama 4 models have a shared expert MoE architecture. It combines a shared, fixed expert that processes every token and combines those results with results from a sparse MoE layer.

Googles gemma-3n-E4B uses a selective parameter activation as well. They use the prefix E for effective instead of A for active (Google, n.d.). In gemma-3n there are parameters to handle input of different types - text, vision and audio - and they get loaded and activated as necessary. This allows a multi modal functionality. It additionally caches the PLE (Per-Layer Embedding) in fast storage (RAM) instead of keeping it in the model memory space (VRAM), allowing to run models in low resource environments.

2.1.4.7 Confidence scores

A LLM can return a log probability score together with each token it predicts. In fact, it can return the top k candidates for the next token with the corresponding log probabilities. The sum of individual log probabilities indicates how likely a sequence of tokens is, according to the LLM. This sum can be interpreted as a kind of *confidence* the model has in a prediction (Berkov, 2025).

Boseak (2025) investigate the usage of log probabilities to score multiple choice answers. They show that the sum of log probabilities carries valuable information about model confidence, but its proper use is nuanced. The choice to use the raw sum or a length normalized sum of log probabilities can have a noticeable effect on accuracy and they give no recommandation which approach to use in general. Ma et al. (2025) argue that normalizing the probability scores is a reason why probability-based methods fail to identify reliability.

Kang et al. (2025) compare different potential measures for a LLMs' confidence for Best-of-N Selection tasks. The most simplest of these approaches uses the normalized log probabilities of the returned tokens. The negative exponent of this normalized sum is also called perplexity. It is widely used in LLM evaluations, even though it has been shown to fail in capturing a model's ability to understand long context. Nevertheless, Kang et al. (2025) show that perplexity can perform equally good as more sophisticated measures of confidence up to 16 choices.

Kauf et al. (2024) investigated, if log probabilities can be used to measure semantic plausibility of sentences. They compare pairs of sentences that are similar, but where one is describing a plausible scenario, whereas the other describes a unlikely one. They show, that comparing the sum of returned log probabilities of two sentences yields better results for a semantic plausibility comparison than prompting the model to make this decision explicitly.

2.1.5 Generation enhancing methods with LLMs

2.1.5.1 In-context Learning

Few-shot learning is a prompting strategy that enables LLMs to solve a wide range of tasks, just by presenting task related examples including the solutions. Brown et al. (2020) calls the rapid adaption to the provided task and examples as *in-context learning*. They show that already GPT-3, as general purpose transformer, surpasses the performance of fine-tuned state-of-the-art models in some tasks.

If the performance gain by *in-context learning* is investigated, the terms *zero-shot*, *one-shot* and *few-shot* learning are often used. In zero-shot learning we only state the task without providing any examples. In one-shot learning a single example is provided. In few-shot learning multiple examples are provided.

The classical alternative to in-context learning is model fine-tuning, in which a model's weights are updated via gradient descent, based on a large corpus of examples (Brown et al., 2020). Through fine-tuning, the resulting model can perform the target task independently, without requiring additional context or descriptive tokens at inference time.

In-context learning reduces the need to create large datasets to fine-tune a model on specific tasks. This brings great flexibility and can be resource efficient by sparing the computational costs for adapting the model to new tasks (Dong et al., 2024). On the other hand, it can pay off to fine-tune a model, if a specific task has to be solved regularly, because for *in-context learning* additional tokens have to be processed every time. This can become energy costly and slows down the task processing.

Text classification and information extraction are two tasks that can be solved with few shot-learning (Zhao et al., 2021). Hereby, the order in which the examples are presented is important - at least for GPT-3 (Zhao et al., 2021).

Although tutorials and blogs sometimes suggest that using more examples leads to better results (Kuka", n.d.), Brown et al. (2020) demonstrate that performance may not improve significantly when adding more than a second example.

Furthermore, there is the threat of *context rot* (Kelly Hong & Anton Troynikov, 2025). This means, the performance can decrease noticeably, when the context gets to long. Thus, the number of examples one should provide is not just constrained by the context width. Additionally, there is a threat of over-fitting on patterns that are not present in the actual task but in the examples if they are to homogeneous.

2.1.5.2 RAG

Wichtig

2.1.5.3 Guided and restricted decoding

Guided decoding is a prompting technique, where a LLM is instructed to respond in a specific format, e.g. as JSON formatted string. It often is combined with *in-context* learning, to improve the probability, to get the format requested. One can provide examples of pre-structured templates, too. For JSON one can request certain keys and properties for the corresponding values.

Restricted decoding, on the other hand, is manipulating the

Willard & Louf (2023) are stating, that they use XYZ? to satisfy formatting requirements that are either hard or costly to capture through fine-tuning alone (or through prompting / few-shot learning), summarizing findings of many articles.

“Most implementations of guided generation bias the score values used to determine the probabilities of the tokens in an LLM’s vocabulary. A common and sufficient approach involves repeated evaluations over the entire vocabulary in order to determine which tokens are valid–according to the constraints and previously sampled tokens–and setting the probabilities of invalid tokens to zero. This approach entails a fixed O(N) cost for each token generated, where N is the size of the LLM’s vocabulary.

We propose an approach that uses the finite state machine (FSM) for- mulation of regular expressions to both arbitrarily start and stop guided generation and allow the construction of an index with which the set of non- zero-probability tokens can be obtained efficiently at each step. The result is an algorithm that costs O(1) on average.”

generation template strict (closed) vs open

always selecting the most probable response (temp = 0), so numeric values are correct and classification as well

2.2 General machine learning and statistics

2.2.1 Sample distribution visualization methods

Boxplots Wickham & Stryjewski (2011) describe boxplots as “a compact distributional summary, displaying less detail than a histogram or kernel density, but also taking up less space. Boxplots use robust summary statistics that are always located at actual data points, are quickly computable (originally by hand), and have no tuning parameters. They are particularly useful for comparing distributions across groups.”

Figure 2.2 shows a box and whiskers plot and its components and compares it to a gaussian probability distribution. Half of all observations fall within the box and the median is marked by a thick line. Outliers are defined as observations that are outside the area marked with the (horizontal) lines -called whiskers - that potentially have small bars at their ends. Outliers can be shown by circles or dots.

The median and quartiles are less sensitive to outliers, than the mean and standard deviation of a sample. Thus, they are more suitable for distributions that are asymmetric or irregularly shaped and for samples with extreme outliers (Krzywinski & Altman, 2014). They can be used with five observations and more. But even for large samples ($n \geq 50$), whisker positions can vary greatly.

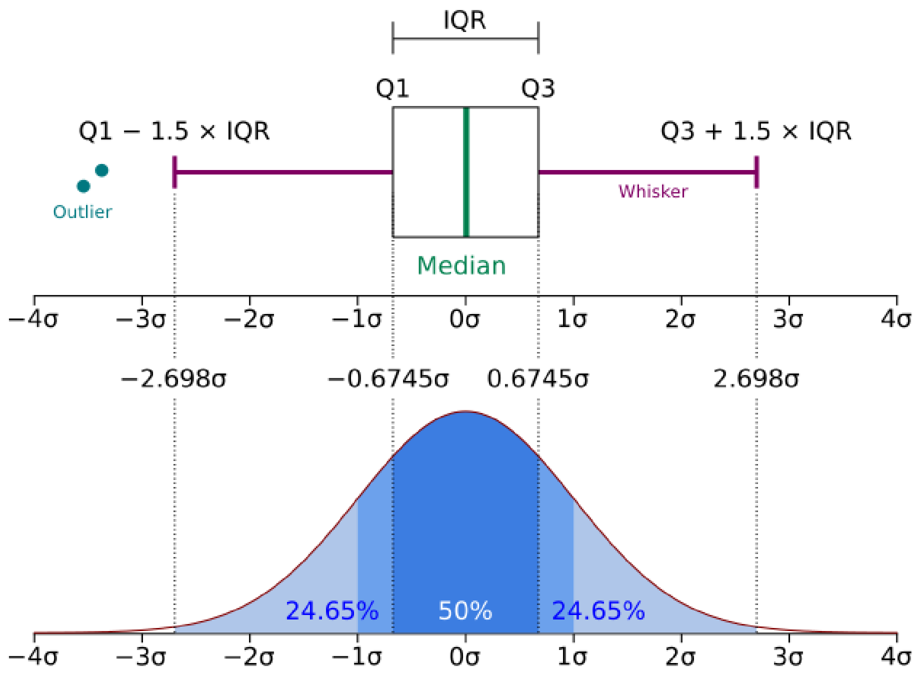


Figure 2.2: Showing a box and whiskers plot with its components - median, quartiles, whiskers and outliers - and compare it with a gaussian probability distribution. Graphic adjusted from Jhguch (2025).

Violin plots There are variations, that try to communicate the sample size of a box plot, either by adjusting the width of the whole box or by introducing notches, that indicate the confidence interval for the median (Wickham & Stryjewski, 2011). Violin plots (Hintze & Nelson, 1998) additionally indicate an density estimate, dropping the strict rectangular shape of the box. Figure 2.3 shows, that the shapes can be necessary to identify multi-modal distributions, that are invisible with regular boxplots (Wickham & Stryjewski, 2011). One can tackle this problem by adding a jitter plot layer to the boxplots. Violin plots can also be used for large datasets, preventing to plot a lot of outliers.

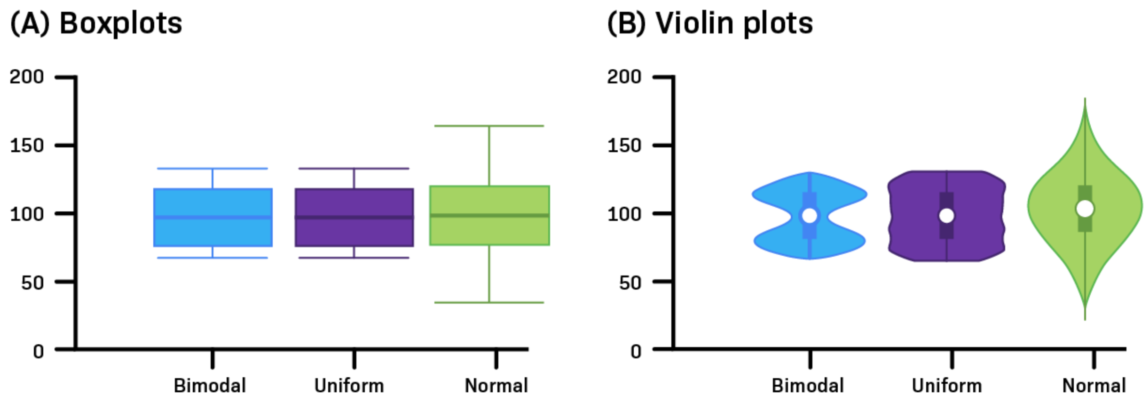


Figure 2.3: Comparing boxplots and violinplots, showing that boxplots can not identify multi-modal distributions on their own. Graphic adjusted from Amgen Scholars Program (n.d.).

2.2.2 Tree based machine learning algorithms

Random forests are an ensemble supervised machine learning technique, composed of multiple decision trees (V. Kulkarni & Sinha, 2013). Mienye & Jere (2024) give a detailed insight into decision trees and their high-performing ensemble algorithms. Tree based machine learning algorithms have gained significant popularity, due to their simplicity and good interpretability (Mienye & Jere, 2024).

Decision tree "The basic idea behind decision tree-based algorithms is that they recursively partition the data into subsets based on the values of different attributes until a stopping criterion is met" (Mienye & Jere, 2024). Figure 2.4 shows this for artificial data of two continuous features. Popular measures to determine how to split a set of observations are the Gini index, information gain or information gain criteria (Mienye & Jere, 2024).

The tree shown is used for a regression task and will predict the average of all values of the corresponding terminal node (leaf). To find out which leaf will be the target terminal node for a given set of features, one simply follows the path from the top node (root) downwards, checking the splitting criteria. Thus, the interpretation of decisions made by a decision tree is straightforward and highly transparent.

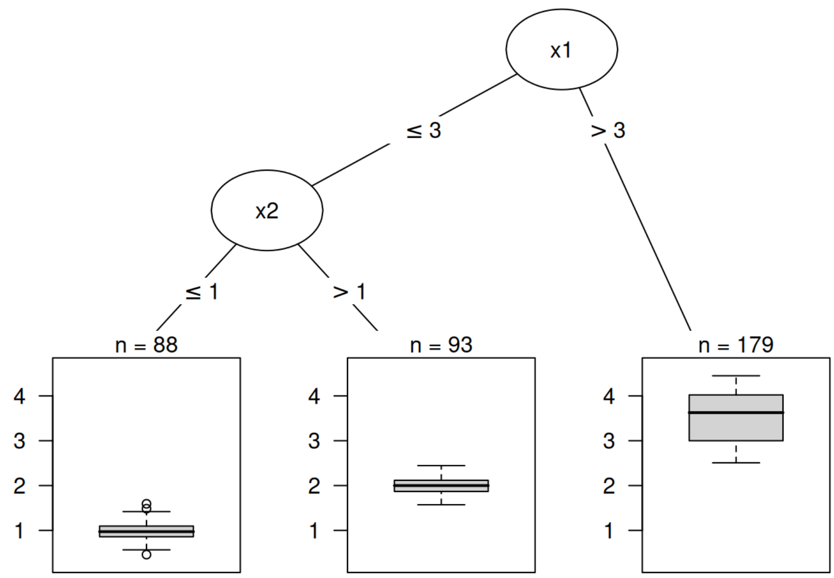


Figure 2.4: Visualizing the of partitioning of a two-dimensional continuous feature space based on multiple splitting criteria for decision tree inducing. Graphic adjusted from Molnar (2025).

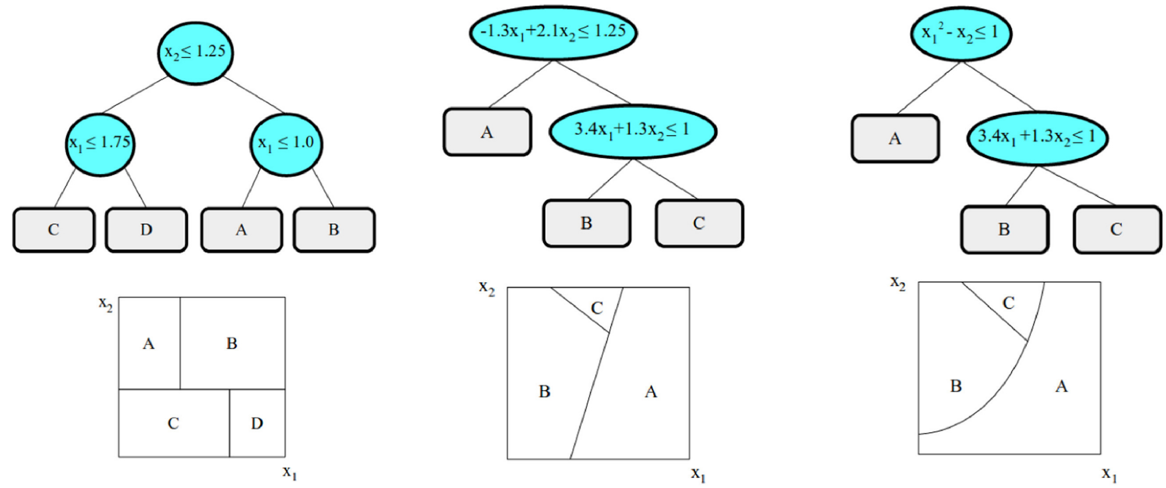
Further benefits of decision trees - besides the good interpretability and computational efficiency - are the native capturing of interactions between features (Molnar, 2025), without modeling this explicitly, as it would for example be necessary in a linear regression. Decision trees can be used for classification and regression. They even can incorporate linear functions as leafs, enabling them to better capture linear relationships (Raymaekers et al., 2024).

Problems of decision trees are, that they lack resilience against data changes and have a tendency to overfitting. A method against overfitting is pruning (Mienye & Jere, 2024). Building an ensemble of decision trees is another possibility, that results in the random forest algorithm, described in the next paragraph .

Rivera-Lopez et al. (2022) focus on decision trees and describe several types, including distinctions based on the splitting procedure (see Figure 2.5). In addition to axis-parallel splitting, they present oblique and non-linear splitting criteria, and provide a state-of-the-art review along with a summary analysis of metaheuristics-based approaches for decision tree induction.

Random forest A random forest is using the principle of bagging and applies it on the level of features and observations. This means, it starts, creating basic decision trees with differing subsets of features and uses bootstrapping to select a randomized set of observations to train the tree with. The final prediction is then determined by voting (for classification) or averaging (for regression) the predictions of all trees in the ensemble.

The induction of the trees can be be parallelized, making it efficient on modern hardware. Random forests can cope with thousands of features and can be applied to large datasets (Breiman, 2001). There are methods that address the problems of imbalanced datasets too. As there are methods to prune a decision tree



(a) Axis-parallel-DT (b) Oblique-DT (c) Non-linear-DT

Figure 2.5: Visualizing the of partitioning of a two-dimensional continuous feature space based on multiple splitting criteria for decision tree inducing. Graphic adjusted from Rivera-Lopez et al. (2022).

to fight overfitting, there are methods to prune a random forest by removing whole trees, to improve the learning and classification performance, too (V. Y. Kulkarni & Sinha, 2012).

Random forests are "powerful learning ensemble[s] given [their] predictive performance, flexibility, and ease of use" (Haddouchi & Berrado, 2024). While it is based on decision trees, that are considered to be *white boxes*, because of their easy interpretability, random forests are seen as *black boxes*. The decision could be tracked without complicated math, but is tedious, because it would require propagating through many decision trees, noting their predictions and then averaging those.

The fact that the random forest model is categorized as a black-box model restricts its deployment in many fields of application (Haddouchi & Berrado, 2024). One feature oriented tool for explainability is the SHAP framework, presented in section 2.2.3. It allows local explanation, a global overview and pattern discovery for random forests (Haddouchi & Berrado, 2024).

Gradient boosted decision trees Another highly effective and widely used advancement to decision trees are gradient boosted decision trees (Chen & Guestrin, 2016). Instead of the principle of bagging it applies the principle of boosting. It is sequentially building decision trees, where the later ones correct the errors in predictions made by the former trees. It uses gradient descent to minimize those errors (Mienye & Jere, 2024).

The XGBoost (Extreme Gradient Boosting) algorithm is a famous member of this family, with "an outstanding performance record" (Burnwal & Jaiswal, 2023). "Among the 29 challenge winning solutions published at Kaggle's blog during 2015, 17 solutions used XGBoost" (Chen & Guestrin, 2016).

In the following we will emphasize some of its benefits as described by Burnwal & Jaiswal (2023):

- XGBoost employs both L1 (Lasso) and L2 (Ridge) regularization in its objective function to penalize model complexity, mitigating overfitting. However, overfitting can occur, especially if hyperparameters are not adjusted properly.
- XGBoost provides feature importance metrics to facilitate model interpretation, facilitating feature selection and improving the understanding of the model's decision-making process.
- XGBoost "runs more than ten times faster than existing popular solutions on a single machine and scales to billions of examples in distributed or memory-limited settings" (Chen & Guestrin, 2016) using parallelization techniques.

But there are some challenges, that are to investigate in future research. E.g. finding methods to handle imbalanced data and automate the hyperparameter tuning process.

2.2.3 Model agnostic explanation models

2

Shapley values Shapley values are introduced by Shapley (2016) originally in 1952 in the field of game theory. He defined three axioms that a fair allocation of rewards must fulfill:

1. Symmetry: If two players contribute the same amount, they are interchangeable and should gain equal reward.
2. Efficiency: The whole value of the game is distributed among the players.
3. Law of aggregation: If a player contributes to multiple independent games, his contribution in total should be the sum of contributions in each game.

From the third axiom a fourth property derives, that is sometimes named independently. If a player is not contributing to a game, he gets no share. O'Sullivan (2023) calls this the *null player* property.

The formula for a single shapley values is given by (S. Lundberg & Lee, 2017)[1]:

$$\phi_i = \sum_{S \subseteq P \setminus \{i\}} \frac{|S|!\,(|P|-|S|-1)!}{|P|!} \left[val\,(S \cup \{i\}) - val\,(S)\right] \tag{2.4}$$

Molnar (2025) bridges the game theory terms to the field of machine learning as follows: "The *game* is the prediction task for a single instance of the dataset. The *gain* is the actual prediction for this instance minus the average prediction for all instances. The *players* are the feature values of the instance that collaborate to receive the gain (= predict a certain value)."

SHAP framework In S. Lundberg & Lee (2017) they present "A Unified Approach to Interpreting Model Predictions" based on shapley values, called SHAP (SHapley Additive exPlanations). It assigns each feature an importance value for every observation. This allows to inspect, why a specific prediction is made and might explain, why a model makes a mistake for specific observations. Inspecting the predictions for all observations can show generalized effects of features.

S. Lundberg & Lee (2017) show that their approach is the only possible explanation model for the class of additive feature attribution methods, that has three desirable characteristics: local accuracy, missingness and consistency. Shapely values can be computed for any machine learning model, but its exact calculation is computationally extremely expensive (Hu & Wang, 2023), since it is of exponential complexity $\mathcal{O}(2^p)$ regarding the number of features (or predictors) $p = |P|$.

Even with the approximation of the shapley values, introduced in S. Lundberg & Lee (2017) as Kernel SHAP, the complexity for tree based algorithms is $\mathcal{O}(MTL2^p)$, with the number of samples M, number of trees T and the number of leaves L. The tree based optimization of the algorithm, TreeSHAP, allows an approximation in $\mathcal{O}(MTLD^2)$ (S. M. Lundberg et al., 2019), with the maximum tree depth D. Depending on the number of observations to calculate shapley values for (M), the Fast TreeSHAP algorithm has a even lower time complexity of $\mathcal{O}(TLD2^D + MTLD)$ (Yang, 2022).

To quantify the overall effect of a feature on the model, we compute the mean of the absolute Shapley values for that feature across all observations. Here is the formula, which we adapted from Molnar (2025) by adjusting the notation and variable names to match Equation (2.4):

$$mean\,(|SHAP|) = \frac{1}{n} \sum_{k=1}^{n} |\phi_i^{(k)}| \tag{2.5}$$

This value is called SHAP feature importance. It can be interpreted similar to standardized beta values for a linear regression. In some cases it would be possible to calculate an effect direction for the feature importance. But it is not common practice. Instead visual representations presented in section @ref() are used for such interpretations.

[1] We replaced F by P to speak in the terms of players instead of features. We also replcaed f by val, because it better fits the story, that this is the value gain in a game, as explained by O'Sullivan (2023).

S. Lundberg & Lee (2017) also showed that SHAP values are more consistent with human intuition than preceding local explainable models. Z. Li et al. (2024) mention, that explainability of machine learning models is not only important for researches but also for practitioners, to demonstrate their reliability to potential users and build trust. Regardless of the popularity of SHAP scores, there are claims that they can be inadequate as a measure of feature importance (Huang & Marques-Silva, 2024). The approximated as well as exact SHAP scores can assign higher value to unimportant features than to important ones.

However, the need for a high explainability of machine learning algorithms is more urgent than ever, since the EU's regulatory ecosystem is emphasizing the importance of XAI (explainable artificial intelligence) (Nannini et al., 2024).

2.3 Summary

This chapter highlighted recent architectures for LLMs, such as Mixture-of-Experts, and provided references for readers to explore foundational concepts and the history of NLP in greater detail. it presented *in-context learning* and retriever augmentation as advanced prompting strategies benchmarked in our experiments.

The review underscored the importance of explainability in machine learning, particularly through frameworks like SHAP, and discussed visualization techniques such as boxplots and violin plots for understanding data distributions.

The discussed methods and concepts directly inform the thesis's approach to extracting financial information from unstructured documents. By presenting advanced LLM architectures and explainability tools, the chapter establishes a foundation for the experimental strategies and evaluation criteria used throughout the thesis.

Building on these insights, the following chapter details the methodology adopted for applying and evaluating these techniques in the context of financial report analysis, setting the stage for the experimental investigations and results.

Chapter 3

Methodology

This chapter describes the research design of this thesis. In the subsequent sections it elaborates

3.1 Problem Definition

This thesis aims to evaluate a framework for information extraction from financial reports using advanced computing algorithms, such as LLMs, presented by H. Li, Gao, et al. (2023). We apply this framework on German annual reports of multiple companies and focus on using open-weight LLMs. This task requires two problems to be solved:

1. The information to extract has to be located in the document.
2. The information has to be extracted correct and a in format that allows further processing in down stream tasks.

We limit the information of interest on the data found in the balance sheet and profit and loss statement. Both are found on separate pages and have a table-like structure. The information extracted should reflect the hierarchy defined in HGB (2025). The information to extract consists of numeric values.

Since the information of interest is placed on separated pages, the first problem is to find the pages that contain the balance sheet and profit and loss statement. We do not attempt to select a specific part of the page, where the data can be found. Thus, this becomes a classification task, if a page contains the information of interest. Spatial information is not processed.

The second problem is an information extraction task. Potential information has to be identified, its entity has to be recognized and finally its numeric value has to be extracted. In this thesis no special techniques specialized on table extraction are used. We just use a plain text extract.

3.2 Research Design & Philosophy

The research design for this thesis is set up, following the guideline found in Wohlin et al. (2024). Figure 3.1 shows the decisions made following this guideline. According to Collis & Hussey (2014) research classification the outcome of this thesis is applied research, focusing on solving a practical problem. Its purpose is evaluation research, comparing different approaches with each other (*benchmarking*). The data collected in our experiments is of quantitative nature and its evaluation uses (semi-)quantitative methods.

3.2.1 Research questions

For this thesis we formulate two main research questions:

Q1 How can we use LLMs effectively to locate specific information in a financial report?

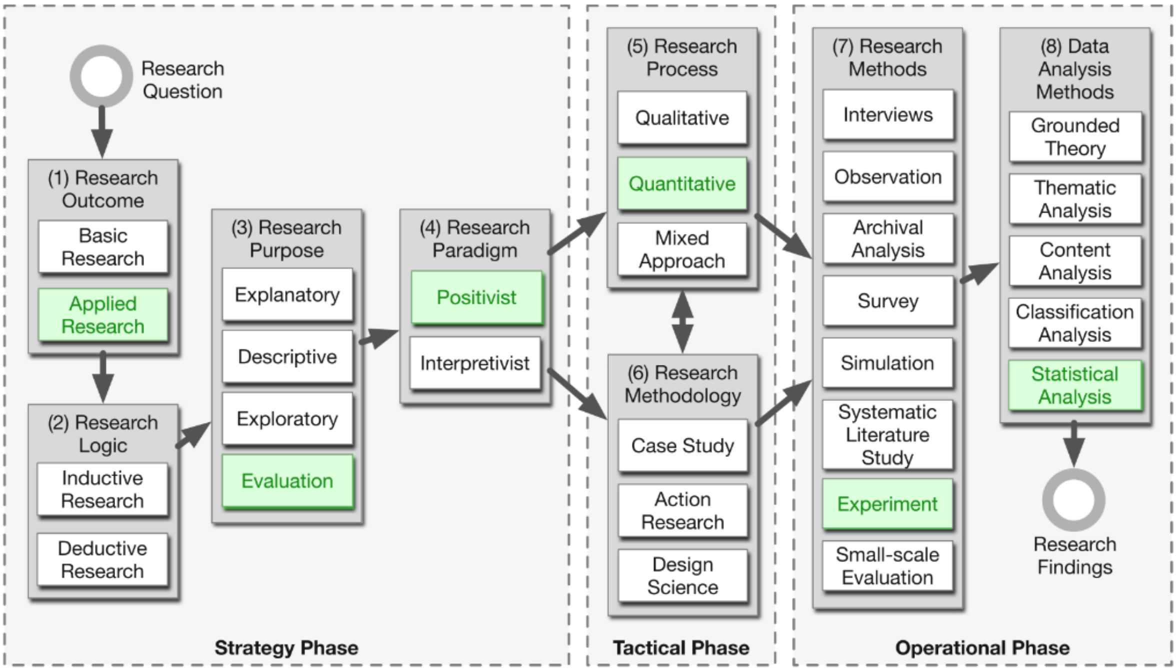


Figure 3.1: Showing the decisions made regarding the research design. (The figure is adapted from Wohlin & Aurum (2015). The copyright for the original figure is held by Springer Science+Business Media New York 2014.)

Q2 How can we use LLMs effectively to extract this information from the document?

Each of this questions is investigated with its own methods and experiments. In the following we will use the term *page identification* to refer to the first research question and *information extraction* to refer to the second.

Additionally, we formulate a UX (user experience) motivated side research question:

Q3 Can we use additional information from the extraction process to guide the user on which values need to be checked and which can be trusted as they are?

The third question is refereed to using the term *error rate guidance*.

3.2.2 Hypotheses

Subsequent we formulate our hypotheses for the research questions. Regarding to the *page identification* task we propose:

H1.1: LLMs can be used to locate specific information in a financial report, achieving a high F1 score.

H1.2: LLMs can be combined with other approaches to reduce the energy consumption, without lowering the systems recall.

For the *information extraction* task we define two groups of hypotheses. The first group proposes, that LLMs can match entities, extract numeric values without mistakes and is not hallucinating if values are not present:

H2.1a: LLMs can be used to correctly extract multiple numeric values from the assets table.

H2.1b: LLMs can match row identifiers and place the numeric values in the correct target row.

H2.1c: LLMs can identify unmatched row identifiers and report, that the value is missing.

The second group of hypotheses proposes, that there are three sources for potential predictors for the information extraction performance: model specific, prompting specific and table specific ones.

H2.2a: Model specific features have an effect on the extraction performance.

H2.2.b: Prompt strategy specific features have an effect on the extraction performance.

H2.2c: Table specific features have an effect on the extraction performance.

Model specific predictors are the model family and the parameter size. Predictors related to the prompting strategies are related to in-context learning decisions, e.g. how many examples should be presented, how should those examples be chosen and is it permitted to use examples from the same company as the current subjects tasks document. Example for table characteristics investigated are the number of columns, visual separation and if there is a currency unit - i.e. *T€* - given.

A list of all investigated predictors and their assumed directions of effect on multiple measures of the information extraction task can be found in the hypotheses evaluation tables in Appendix D.

The hypotheses for our side research question name the *confidence score* as concrete target for our investigations.

H3.1: The confidence score can be used to guide the user on which of the identified pages need to be checked and which can be trusted as they are.

H3.2: The confidence score can be used to guide the user on which of the predicted values in the information extraction task need to be checked and which can be trusted as they are.

3.2.3 Evaluation research

We follow the process of evaluation research, in order to investigate our research questions. We compare different approaches to solve the two tasks, searching for the most effective setup, to solve the problems. A setup is considered effective if it achieves good results while being as computationally efficient as possible.

As a baseline for each task a regex (regular expression) based approach is set up. Regular expressions are chosen as baseline because they are computationally efficient. The results are compared with the authors human performance as well. The results will be used to implement an application that is used by the employees of RHvB in future.

3.3 Evaluation Strategy

This section defines, under which conditions we consider a task to be successfully solved and which metric we use to measure the outcome.

3.3.1 Evaluation framework and metrics

3.3.1.1 Page identification

The page identification task is successful, if a page is correctly classified to contain the information of interest. The balance sheet is composed of the assets (*Aktiva*) and liabilities (*Passiva*) table. Together with the profit an loss statement (*Gewinn- und Verlustrechnung, GuV*) they form the three target classes. The fourth class is called *other*. Subsequently will will use the German terms for the target classes (or table types): **Aktiva**, **Passiva** and **GuV**.

Metrics The distribution of target classes and pages of type *other* is highly imbalanced. At most two pages per target class are found in documents with up to 152 pages. Thus, following Saito & Rehmsmeier (2015) suggestion, we report measures as precision, recall and F1 score instead of accuracy, to describe the approaches performances.

In a HITL application the recall value might be of higher interest than the F1 score. More precisely, in those cases the number of pages to check until the correct page is found is of interest. Thus, the top k recall is reported additionally, if the approach permits to rank the classified pages according to a score.

An alternative measure for the F1 score would be the AUC (area under the curve) score precision-recall curve.

3.3.1.2 Information extraction

The information extraction task if successful, if the correct numeric value is extracted with the correct entity identifier in the correct json (JavaScript Object Notation) format. If a value, defined by the legal text, is not present *null* should be returned with the corresponding entity identifier. The entity identifier can be composed of up to three labels, representing the hierarchy defined in the legal text.

Metrics We use two measures to describe the approaches performances for the information extraction task. First, we check how many of the predicted numeric values are matching the numeric values in the ground truth. The only permitted differences are based on the number of trailing zeros. We do not check for partial correctness, since the real life application requires totally correct extracted numbers.

Second, we report the F1 score for correctly predicting values as missing and thus returning *null*. The distribution of missing values and given numeric values is not imbalanced. Nevertheless, we report the F1 score to establish a comparability with the results of the page identification task.

3.3.1.3 Error rate guidance

An error rate guided result checking process can be implemented, if we can use extraction task related information, to identify a prediction trust worthy. This means, we could white list these values and red flag the remaining ones. Thus, we could guide the users attention in the error checking process on those values, that empirically tend to have a high chance to be faulty.

Metrics In this thesis we focus our attention on a criteria, which we name *confidence*. We calculate the *confidence* score for answers received from LLM based on the non-normalized sum of token log probabilities (Boseak, 2025):

$$confidence = \exp\left(\sum logprob(token_i)\right) \tag{3.1}$$

For the classification tasks this is equal to the normalized (averaged) sum, since the answer is either containing just one token or the subsequent tokens have a log probability of 0, because the answer is fully determined by the first token.

We are using the non-normalized sum of token log probabilities for the information extraction task too, because we want a single uncertain digit to flag the whole numeric value as unreliable. This means, that shorter answers tend to have higher *confidence* scores. This is especially true for predicting *null*. Thus, we investigate the prediction of numeric values and *null* separated.

We group predictions based on their confidence scores in intervals with a range of 0.05 and calculate the empirical error rate for each interval. We hope to find a error rate (close to) zero for the highest confidence intervals and a noticeable proportion of predictions falling into those intervals.

3.3.2 Benchmarking

Comparing the performance of different approaches benchmarked in this thesis is possible, because the approaches within a task are performed on a common document base. The task to solve is the same for each approach. The prompts for the different prompting strategies are build systematically and derive from the base prompt formulated for the *zero shot* strategy. Comparing the runtime or energy consumption gets possible with the GPU (graphics processing unit) benchmark data (see section 4.7)

3.4 Data Strategy

The population of annual reports of interest for the work at the RHvB is composed of all annual reports of companies, where the state of Berlin holds a share. There are often multiple versions of those annual reports: one that is publicly available and targeting share- and stakeholders. The structure and layout of these reports is quite heterogeneous. Often there is a second version that is used internally or for communications with public administrations. They often consist of plain text and tables and show neither diagrams nor photos.

Since the evaluations are run on the BHT (Berliner Hochschule für Technik) cluster and partially in the Azure cloud, we work with the publicly available reports, while at RHvB the internal documents are more common. The annual reports mostly are downloaded from the companies websites. Some documents are accessed via Bundesanzeiger or the digitale Landesbibliothek Berlin.

For the page identification task all kinds of pages from the annual reports are used. For the information extraction only pages with **Aktiva** tables are used. In addition, a set of self-generated synthetic **Aktiva** tables is used for the information extraction task. It is created to systematically investigate potential effects of characteristics financial tables could have.

3.4.1 Sampling methodology

The reports are selected with the goal of reflecting the heterogeneity of the population of documents within the chosen sample.

Page identification For the page identification task companies from different fields of business are selected and all publicly annual reports downloaded. The companies chosen are found in the first row of Figure 1.1.

The assumption is, that each company within a branch can represent the document style for other representants of the same branch and that the position within a column has no implications for the annual reports styles. Realizing that the *degewo AG* reports would require OCR preprocessing we additionally downloaded reports for *GESOBAU AG*. A description of the resulting data set can be found in section @ref().

Information extraction For the information retrieval task the publicly available reports for all companies are downloaded. A amount of documents chosen for each company is more balanced than for the first task. A description of the resulting data set can be found in section @ref().

We excluded all companies, that do not report their assets table in the most detailed fashion defined by the legal text, because these asset tables do not match with the strict schema we use. This decision excludes the reports of well known companies as BVG and BSR.

Error rate guidiance We investigate the third research question on the two datasets described above, instead of creating a separate dataset.

3.4.2 Ground truth creation process

The ground truth for both tasks is created by manual annotation through the authors. The results of early experiments are used to check the ground truth for mistakes or missed items. Found issues get resolved. Details on the ground truth creation processes can be found in section @ref()

3.4.3 Preprocessing

We use plain text extracted from the annual reports for all tasks. We do not extract geometric coordinates for the text. Auer et al. (2024) describes, that available open-source PDF parsing libraries may show issues as poor extraction speed or randomly merged text cells. We tested five PDF extraction libraries, because the results of all subsequent experiments will depend on the text extracts. Section E.3.1 shows the results.

We perform no manual data cleaning, because this will not be done from the employees of RHvB either.

3.4.4 Data splitting

When we train a machine learning model, we split the data into train and a test set. We do not use a validation set, because we do not compare models using an extended hyper-parameter variation strategy. Instead we just report the performance found for the models build with default settings. We build two random forests for the term frequency approach in the page identification task and more random forests for evaluating the hypotheses for the information extraction task.

Building the term frequency random forest, we face a highly imbalanced dataset. We apply undersampling for the training and evaluate the model on the imbalanced test set.

3.5 Experimental Framework

3.5.1 LLM overview

Table 3.1 gives an overview on all LLMs used for the tasks in this thesis. It shows the passive parameter count in billions for each LLM and shows in which specific approach it is used with a tick. Overall 37 models from 10 model families are used. If available, we use a instruction fine tuned version of the models.

3.5.2 Approaches

3.5.2.1 Page identification

Regular expressions We develop multiple sets of regular expressions and filter out all pages that do not fulfill all regular expressions of a given set. There are different sets for each target type, **Aktiva**, **Passiva** and **GuV**. The sets also differ in how versatile they can cope with additional white space introduced by a imperfect text extraction and how many different words for a given term are accepted. Figure 3.2 shows an example for two sets of regular expressions to identify a **Aktiva** page.

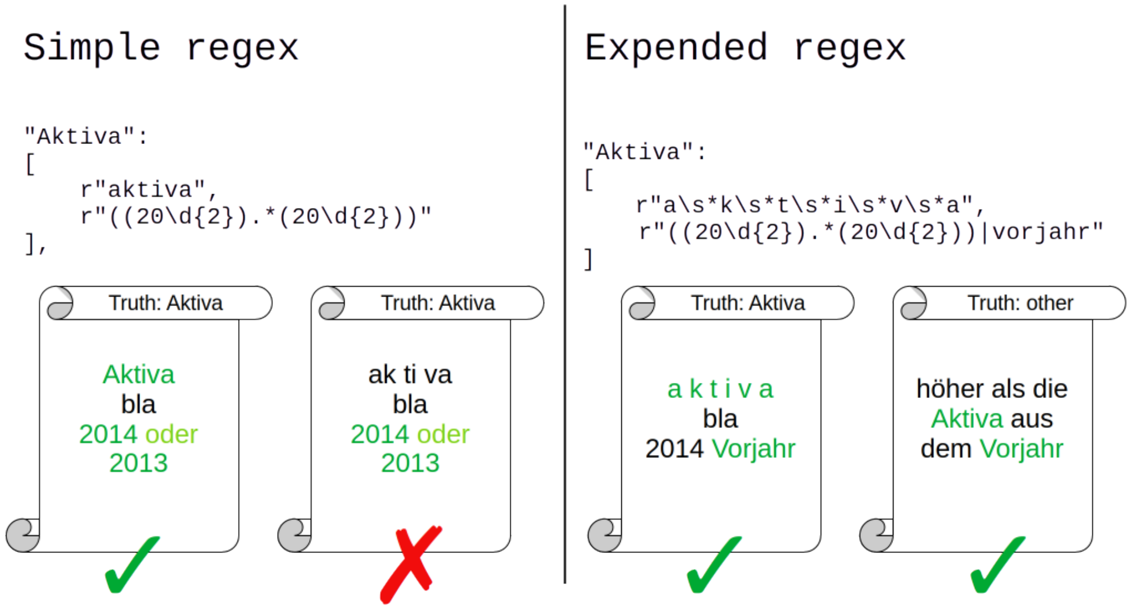


Figure 3.2: Comparing the prediction of two different sets of regular expressions on dummy pages. The simple one has a lower recall, while the expended one has a lower precision.

Table 3.1: Overview of benchmarked LLMs for all tasks. Parameter shows passive parametercount in billions.

		information extraction			page identification	
model	parameter	real tables	synth tables	hybrid	binary	multiclass
chat-gpt						
gpt-4.1	NA	✓				
gpt-4.1-mini	NA	✓				
gpt-4.1-nano	NA	✓				
gpt-5-mini	NA	✓				
gpt-5-nano	NA	✓				
google						
gemma-3-4b-it	4	✓			✓	✓
gemma-3n-E4B-it	4	✓			✓	✓
gemma-3-12b-it	12	✓	✓	✓	✓	✓
gemma-3-27b-it	27	✓	✓	✓	✓	✓
Llama-3						
Llama-3.1-8B-Instruct	8	✓	✓	✓	✓	✓
Llama-3.1-70B-Instruct	70	✓	✓		✓	✓
Llama-3.3-70B-Instruct	70	✓	✓		✓	✓
Llama-4						
Llama-4-Scout-17B-16E-Instruct	109	✓	✓	✓	✓	✓
Llama-4-Maverick-17B-128E-Instruct-FP8	402	✓	✓		✓	✓
microsoft						
phi-4	15	✓			✓	✓
mistralai						
Ministral-8B-Instruct-2410	8	✓	✓	✓	✓	✓
Mistral-Small-3.1-24B-Instruct-2503	24	✓	✓		✓	✓
Mistral-Large-Instruct-2411	124	✓	✓	✓	✓	✓
openai						
gpt-oss-20b	20	✓				
gpt-oss-120b	120	✓				
Qwen 2.5						
Qwen2.5-0.5B-Instruct	0.5	✓			✓	✓
Qwen2.5-1.5B-Instruct	1.5	✓			✓	✓
Qwen2.5-3B-Instruct	3	✓			✓	✓
Qwen2.5-7B-Instruct	7	✓	✓		✓	✓
Qwen2.5-14B-Instruct	14	✓			✓	✓
Qwen2.5-32B-Instruct	32	✓			✓	✓
Qwen2.5-72B-Instruct	72	✓	✓		✓	✓
Qwen 3						
Qwen3-0.6B	0.6	✓				
Qwen3-1.7B	1.7	✓				
Qwen3-4B	4	✓				
Qwen3-8B	8	✓	✓	✓	✓	✓
Qwen3-14B	14	✓				
Qwen3-30B-A3B-Instruct-2507	30	✓	✓		✓	✓
Qwen3-32B	32	✓	✓		✓	✓
Qwen3-235B-A22B-Instruct-2507-FP8	235	✓	✓	✓		
Qwen3-235B-A22B-Instruct-2507	235		✓		✓	✓
tiiuae						
Falcon3-10B-Instruct	10	✓			✓	✓

Table of Contents Understanding We use a LLM to extract the TOC from the first pages from a document or use the embedded TOC and prompt a LLM to identify the pages where the **Aktiva**, **Passiva** and **GuV** are located. Figure 3.3 shows a screenshot of a annual report with an embedded TOC and its TOC in text form.

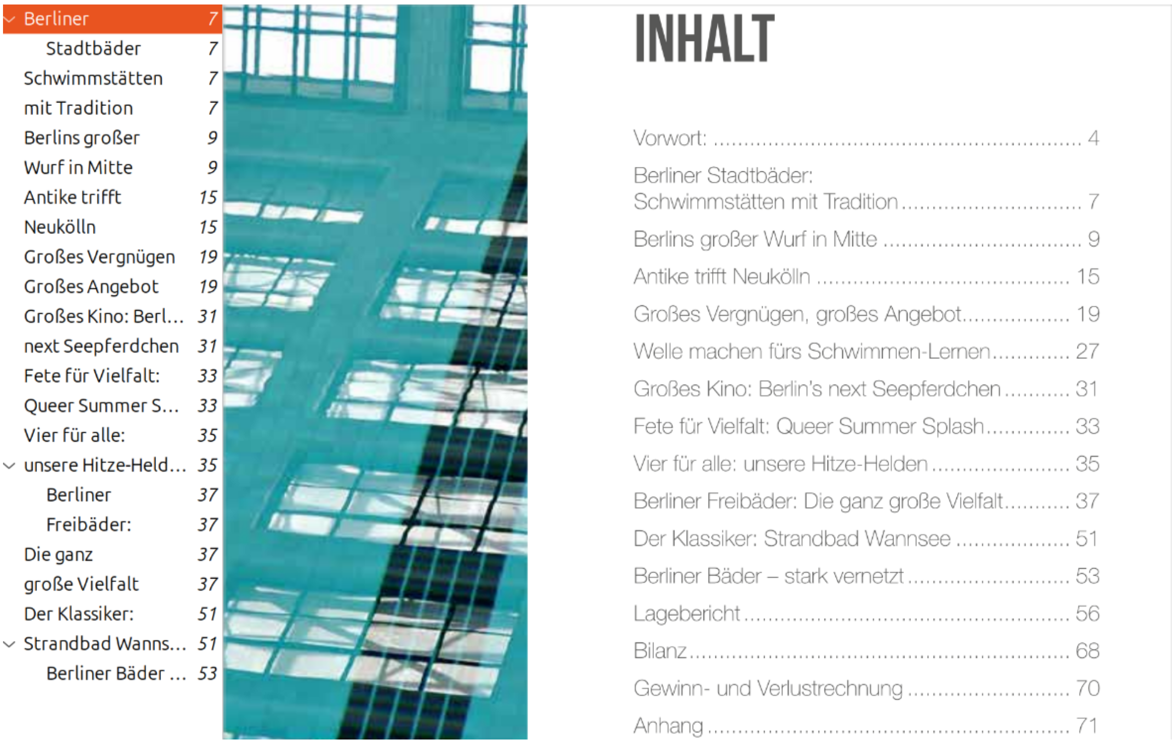


Figure 3.3: Showing a screenshot of a annual report with an embedded TOC (left) and its TOC in text form (right). The embeded TOC is not listing all entries from the TOC in text form.

Large Language Model Classification We use LLMs to classify if the text extract of a given page is containing a **Aktiva**, **Passiva** or **GuV** table or something else. We test binary classification and a multi-classification approach. The reported confidence scores can be used to form a ranking, which text extract might be most similar to the target type. We use a sampling temperature of zero, because we neither want creative nor variable answers, but just the most probable class and exact copied numbers.

We test a wide range of open-weight models and compare different prompting techniques. Figure 3.4 shows, how the prompts are composed for the different strategies. Besides a zero shot approach we test few-shot in-context learning with examples that are either chosen randomly or retrieved based on their vector similarity. Finally, we test passing the legal text instead of examples from a annual report.

Term frequency Ranking We use normalized term frequencies and normalized float frequency to as features for a classification using a random forest. The predicted scores are used to build a ranking, which page most probably contains the target pages. Undersampling is used during training, to handle the unbalanced data. Figure 3.5 visualizes, how the prediction works in this approach.

3.5.2.2 Information extraction

Regular expressions We use regular expressions to extract the numeric values for matching row identifiers. The regular expressions handle line breaks between words in the row identifiers, but not within a word. They can handle multiple signs of white space. Besides that, they try to fully match the labels from the legal text with the text extract, ignoring upper case. They extract numbers with "." as thousands separator. Figure 3.6 is visualizing those capabilities.

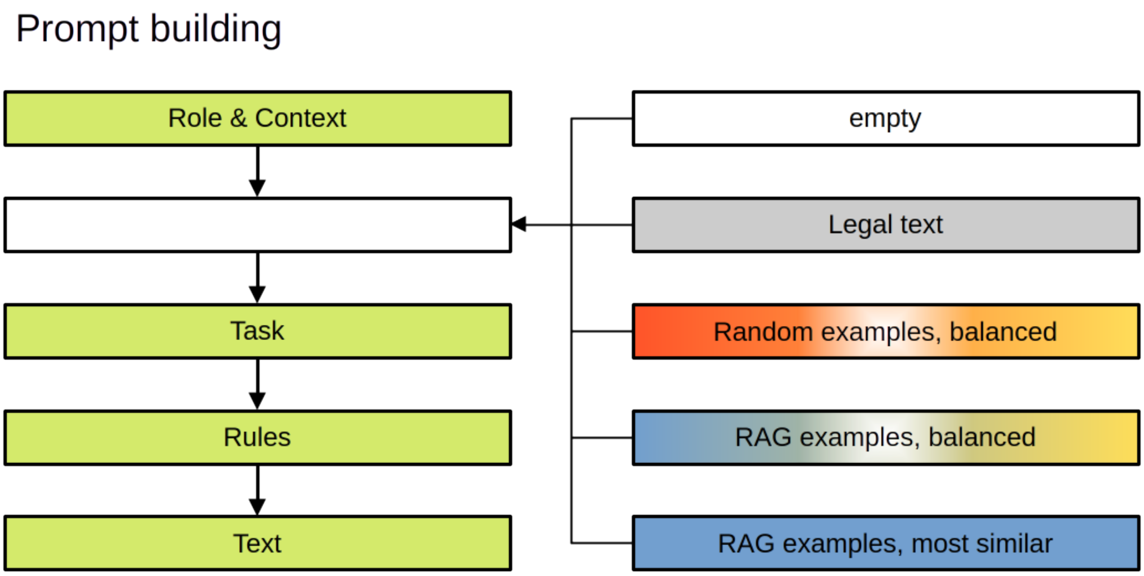


Figure 3.4: Showing the basic structure of the prompts and which strategies are used to pass additional information to the LLM.

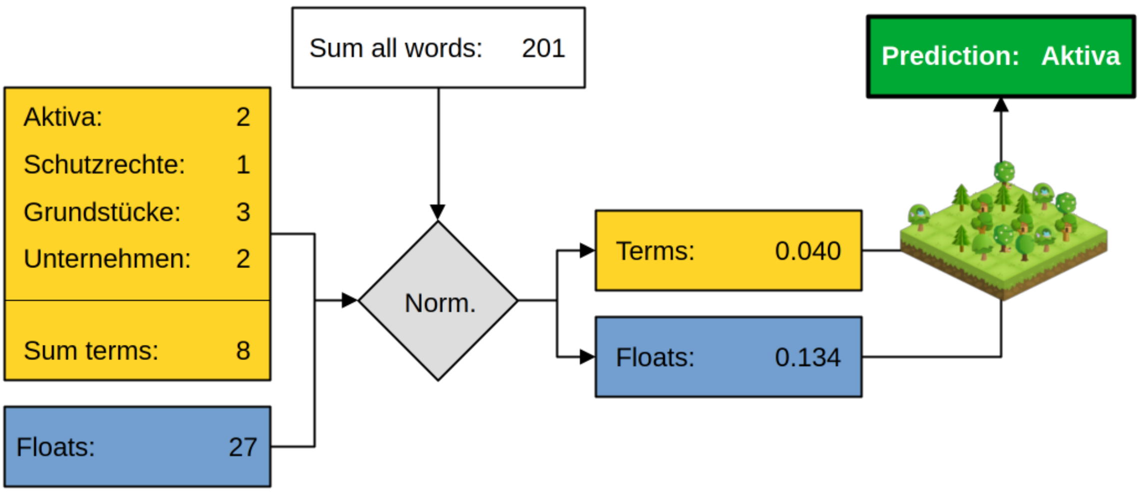


Figure 3.5: Visualizing, how term and float frequency get calculated and used to predict, if a page is of the target class.

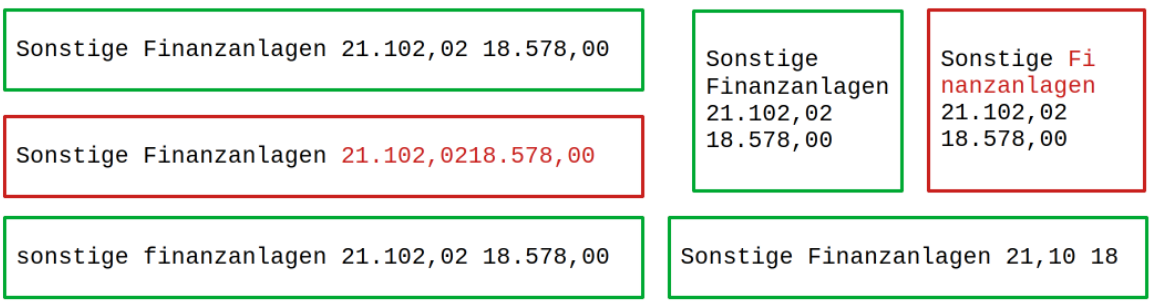


Figure 3.6: Visualizing the extraction results for different text examples. Texts in green boxes are matching our regular expression. Texts in red boxes do not, because of the red text part.

3

Real tables We use LLMs to extract the numeric values of real **Aktiva** tables with restricted generation. We use a sampling temperature of zero, because we neither want creative nor variable answers, but just the most probable class and exact copied numbers.

The LLM has to group row identifiers and corresponding numeric values and match the row identifier with the labels of the schema. If a row identifier is unknown, the values have to be discarded. If a label is not present among the row identifiers, the model predicts *null*. All values are extracted in one pass. We do not include any instruction, how to proceed with currency units, that might be given for certain columns.

We test a wide range of open-weight models and compare different prompting techniques. Figure 3.7 shows, how the prompts are composed for the different strategies. Besides a zero shot approach we test few-shot in-context learning with examples that are either chosen randomly or retrieved based on their vector similarity. Finally, we test passing a synthetic **Aktiva** table as example. We test models from OpenAIs GPT family in addition to the open-weight models.

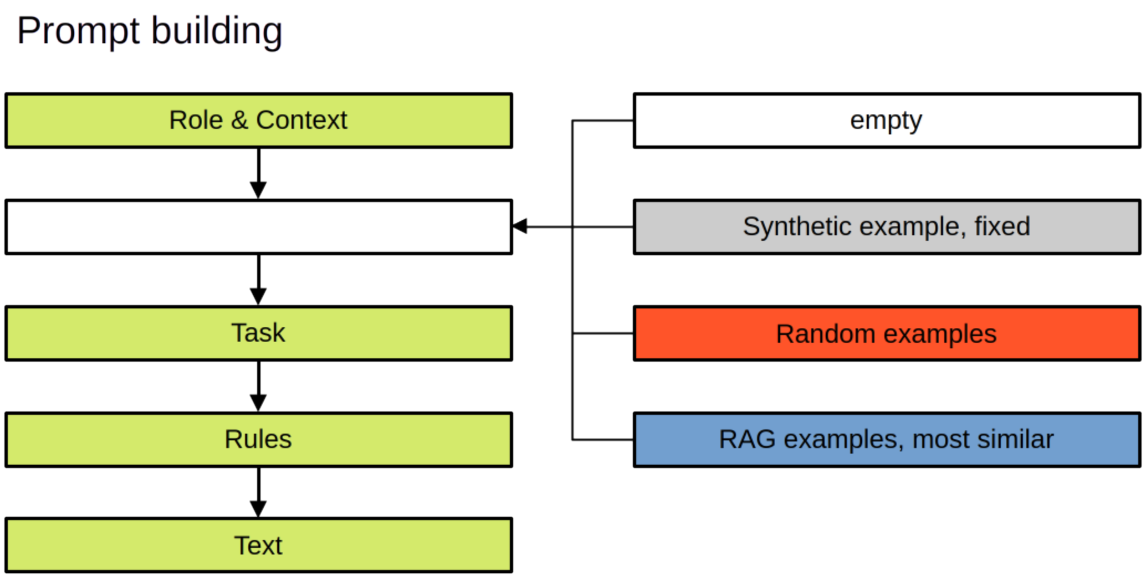


Figure 3.7: Showing the basic structure of the prompts and which strategies are used to pass additional information to the LLM for the information extraction task.

Synthetic tables We use LLMs to extract the numeric values of synthetic **Aktiva** tables with restricted generation. The procedure is identical as with the real **Aktiva** tables. We extract all values with and without an explicit instruction on how to proceed with currency units. We limit our test on the open-weight models.

Hybrid approach We use LLMs to extract the numeric values of real **Aktiva** tables with restricted generation, providing examples from synthetic **Aktiva** tables. The procedure is identical as with the real **Aktiva** tables. We extract all values with and without an explicit instruction on how to proceed with currency units. We limit our test on the open-weight models.

3.5.3 Error analysis

To better understand the limitations of the evaluated models and find ways to further improve the system, we will conduct a detailed error analysis. Representative error cases will be documented to illustrate typical failure modes and to inform potential improvements for future work.

3.5.3.1 Page identificaion

Content rot: We will first quantify the types of misclassifications using confusion matrices and error rate statistics.

TF: Additionally, we will manually inspect a sample of erroneous predictions to identify common causes, such as ambiguous table layouts, or model misinterpretations.

(Grandini et al., 2020)

3.5.3.2 Inforamtion extraction

Under construction!

99.5 % or 96 % accuracy for extracting financial data from Annual Comprehensive Financial Reports (H. Li, Gao, et al., 2023) In the untabulated test, GPT-4 achieved an average accuracy rate of 96.8%, and Claude 2 achieved 93.7%. Gemini had the lowest accuracy rate at 69%. (ebd.)

found error types: including omissions when the LLM was instructed to extract a list of line items, misjudgment of units (such as thousands or millions), and incorrect identification of rows and columns

failed to extract all of list, Too many hallucinated values when it was NA instead (Gougherty & Clipp, 2024)

For the information extraction task, we expect to find issues with wrong extracted numeric values due to disrespecting currency units or hallucinated numbers, if a value is absent. We further expect wrong entity recognition and thus wrong row identification matching.

Error rates will be compared among different experimental setups to reveal systematic weaknesses in a stratified analysis.

Finally, we investigate some of the erroneous extracted examples manually, and try to identify the underlying issues.

Tools and criteria

Reporting

Example:

Numeric values are difficult to handle for langauge models in specific tasks. Copying numbers seems not to be a recent problem. How about transforming by multiplication with 1000?

3.5.4 Evaluation methods

We use a lot of visual representations for the interpretation of the results. Many of those can be found in the technical report sections of the appendix. Bigger graphics with a lot of small multiples can be found in the appendix G. These visuals have a high information density and enable us to compare a lot of results - including distributional information - at once[1].

We prefer this process over comparing many rows with measures of location (e.g. mean and median) and measures of distribution shapes (e.g. standard deviation, median absolute deviation, skewness). For presenting the results in chapter 5 we compress a selection of results in tabular form, because it is easier to grasp. Some interesting observations are discussed in chapter 6. But a lot of information is still kept in the visuals.

Boxplots We use a lot of boxplots to compare the distributions of sample of results. We also inspect violin plots or add a point jitter to the graphics during the inspection, to check for multi-modality. If we find intersting details, we keep those for the presented graphics. If not, we just present the boxplots.

[1] A big screen with high resolution is helpful for this process.

SHAP For interpreting the SHAP values, we use bar plots showing the mean absolute shapley values as feature importance, which can be interpreted as an indicator of effect strength. Furthermore, we use bee swarm plots (also called summary plot) and dependency plots, to investigate if we can name a direction for the effect and identify interesting interaction patterns. Further information can be found in chapter 18 of the book "Interpretable Machine Learning" (Molnar, 2025).

random forest We build a random forest classifier in the term frequency approach for the page identification task, to evaluate our hypotheses **H2.2**. We perform no hyper parameter optimization. We want to get a first glimpse on what might influence the extraction instead of getting the best possible performance. We are not modelling causal relationships at all.

We do not use XGBoost for the final analysis, because calculating the SHAP values for XGBoost model took to long. Linear and logistic regression models are fitted as well, but not used for the final analysis.

Chapter 4

Implementation

(max 5p)

4.1 Environments

The computations for this thesis are performed in two environments. Tasks that do not require a GPU are run locally. You can find the specifications of the local device in section E.2. Other tasks are run on the Datexis Kubernetes cluster.

4.2 Software Packages

Python We use vLLM (Virtual Large Language Model) and *chromadb* to set up our in-context learning LLM pipeline. vLLM is a fast and easy to use framework to provide LLMs locally or as service. *chromadb* is a vector database and key component of our RAG architecture. It provides the LLM with examples that are most similar regarding their vector representation.

For data wrangling we use *pandas*. For transferring data and saving results we use the *json* library. We extract texts from PDFs mainly with *pypdfium2* and use *sklearn* for building random forest classifiers.

R We use the *tidyverse* library for data wrangling and creating our visualizations with *ggplot*. We use *h2o* for building random forest models on the Datexis cluster and *shapviz* to interpret these models. *knitr* is used to create this thesis document.

4.3 Ground truth datasets

4.3.1 Ground truth creation

This section describes the ground truth creation process in detail. It describes our initial annotation process and what improvements are made, based on later iterations and findings from the experiments.

Page identification For the page identification task the chosen documents are searched for the target pages either by using the search functionality, TOC or scrolling through all pages. For each target page the filepath, page and type is listed in a csv file. For some reports there are multiple pages present for a single target type. In this case, both pages are added to the ground truth. Sometimes the **Aktiva** and **Passiva** page are on a single page. In this case a single entry is made and its type is *Aktiva&Passiva*. If a table spans two pages, both pages are recorded. Excluding pages that need OCR processing we created 252 entries.

For double checking all identified pages are extracted from their original PDF files and combined in a single file. Thus, problems with the numbers shown in the PDF viewer and the actual page number in the file are identified and resolved. After the first experiments pages, that have been classified as a target by multiple models, are checked. Thus, some additional target tables, that span two pages, are identified.

Additionally we identified all pages, that contain a table of any type in a earlier created sample of pages. This smaller sample is based on all pages that are classified as holding a table of type **Aktiva**, **Passiva** or **GuV** by the regex approach. The true and false positive classified ones. This ground truth is used to test the capabilities of various Python libraries, LLMs and visual models to detect tables. Results can be found in the appendix in section E.3.2.

Information extraction For the information extraction task we copy the numeric values from the annual reports into csv files, replace the thousands separators and floating point delimiters and multiply those values by 1_000, if a currency unit is given for the column the value comes from. The csv files are already prefilled with all entities defined in the legal text, identified by their full hierarchy. Thus, we choose which line to put the value in, if the description in the annual report is different.

There are cases, where a single line defined in the legal text is split up into multiple lines in the annual reports. In those cases we enter the sum into the according row in the csv file. If entities are found, that do not fit any entity given in the legal text, this entry is dropped. For the first iteration the csv files just contained the entities and column names but no values.

In the second iteration we use the predictions of Qwen3-235B, check the values and mark mistakes, correct the values and log all mistakes found. In this iteration we check the ground truth created in the first iteration as well and correct mistakes made earlier.

Additionally, we implemented an algorithm to generate synthetic **Aktiva** tables with systematically varied characteristics to enable experiments that focus those characteristics as possible predictors for the prediction performance. Additionally, this enables us to estimate an upper bound for the extraction task with real **Aktiva** tables.

Error rate guidiance There is not ground truth for the calculated *confidence* scores. We will simply check, if the true and false predictions can be separated based on the returned *confidence scores*.

4.3.2 Ground truth database composition

This section describes the resulting ground truth database in detail separate for all our tasks. It shows how many documents from which companies are included and how the synthetic dataset is created.

Page identification Figure 4.1 shows how the document base for the tasks in this section is composed. Overall 74 annual reports from 7 companies are used. For this thesis the tables of interest are those that show **Aktiva**, **Passiva** and **GuV**. Among the 4981 pages 265 tables have to be identified on 251 pages. Figure 4.1 also gives an impression on how many pages the documents have. The documents of *IBB* tend to be longer. The documents of *Amt für Statistik Berlin-Brandenburg* tend to be shorter.

Table 4.1 shows how many documents have multiple target tables per type and how many target tables span two pages. In total 21 tables are distributed on two pages. In 8 documents there are multiple tables per type of interest. There are 14 pages with two target tables (**Aktiva** and **Passiva**) on it.

Table 4.1: Showing the number of documents with multiple target tables per type and the number of target tables that span two pages.

type	multiple targets in document	target two pages long
Aktiva	7	1
GuV	8	20
Passiva	7	0

4

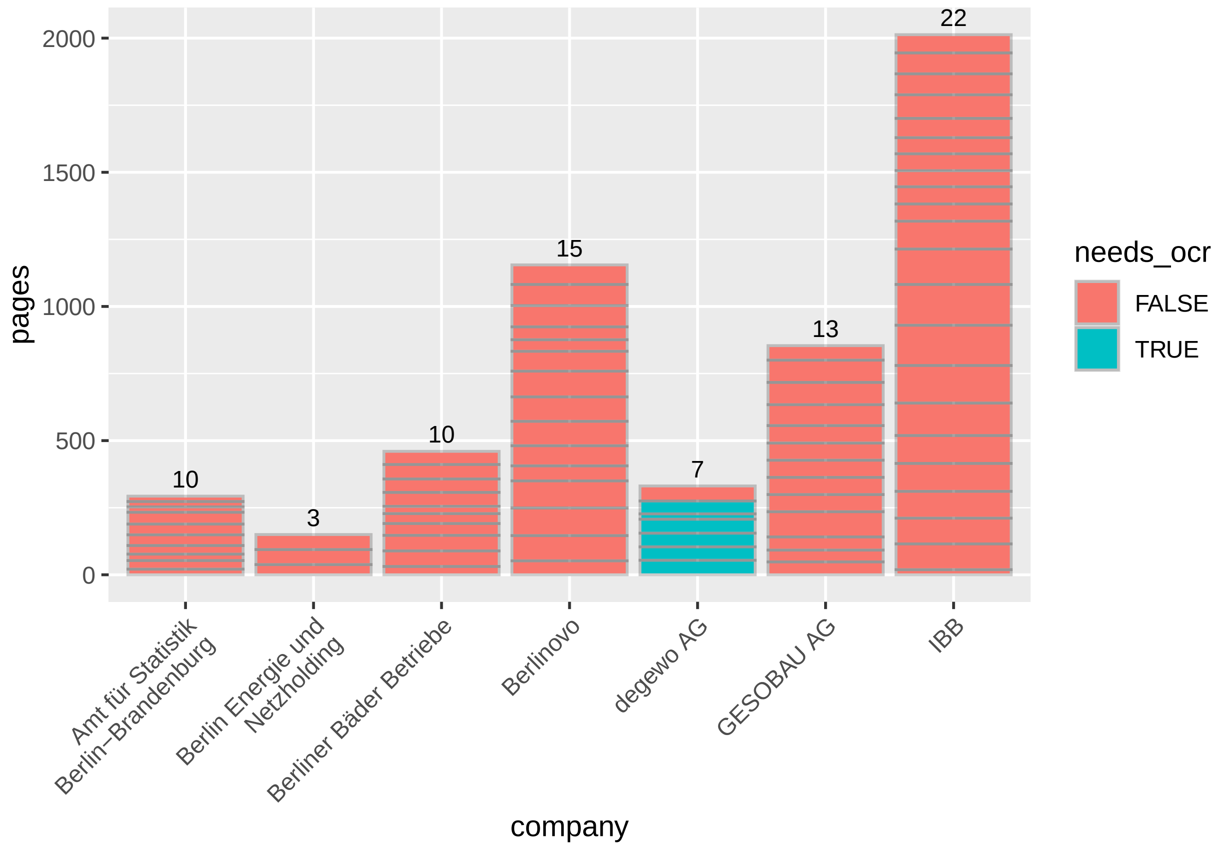


Figure 4.1: Showing the number of pages (bar height) and number of documents (number above the bar) per company for the data used for the page identification task. Some documents would require ocr before being processed and were not used.

Information extraction For the manual information extraction we need up to 12 minutes per table. The maximum amount of values to copy and format (or type manually) among the tables used is 40. In addition to this manual process conceptional process can be necessary, because the values have to matched to the strict grammar. Sometimes we have to decide that there is no row a value fits in or there are multiple values that have to get summed up in order to calculate the value that fits in the predefined schema.

This manual work was done for 36 documents. For every company that published the detailed form of balance sheets a single document was included. Additionally documents were included for *Amt für Statistik und Brandenburg* to check, if a context learning approach is benefiting from documents from the same company.

Later, the predictions of the LLMs were used, to create additional 106 ground truth tables. The old ground truth tables were checked in this this iteration and an error rate of 2.4 % was detected. Thus, the human reference score for percentage of correct predictions is 0.976. Figure 4.2 shows how many **Aktiva** tables are used for all tasks in this subsection, that use real data instead of synthetic data.

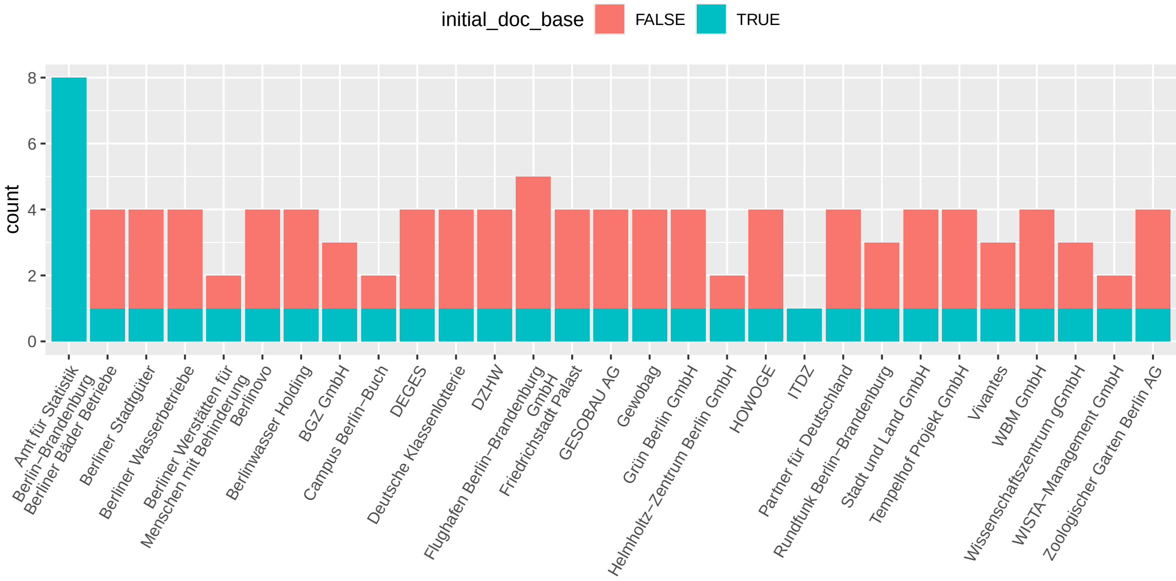


Figure 4.2: Showing the number of documents used for the table extraction task. The number of Aktiva tables is equal to the number documents.

To overcome the limited amount of real data and to allow the systematic investigation of potential predictors for the extraction performance, even if their occurrence is very unbalanced within the real data, synthetic **Aktiva** tables were created. Subsection B.2.2 gives a detailed description, which table features are varied systematically, resulting in 16_504 tables in three formats each.

4.4 Data processing

This section is describing the data flow for our LLM using approaches. Figure 4.3 is visualizing the different processing steps. On the blue area the data flow of the benchmark is presented, starting with the PDF of an annual report as input and the benchmark results as output in json format. The green area shows how R (Markdown) is used to visualize the results for analysis and interpretation as well as to document the results in this thesis.

We start by extracting the text for each page from the annual reports, using PDF extraction libraries like *pdfium*, *pymupdf* or *pdfplumber*. We do this once for each document and save all text extracts in a json file together with the original file path and pagenumber.

For LLM approaches we embed the texts in the prompt template. The (merged) texts are then processed by the algorithm, predicting, if the text is including a specific type of information or not. The predictions are compared with ground truth.

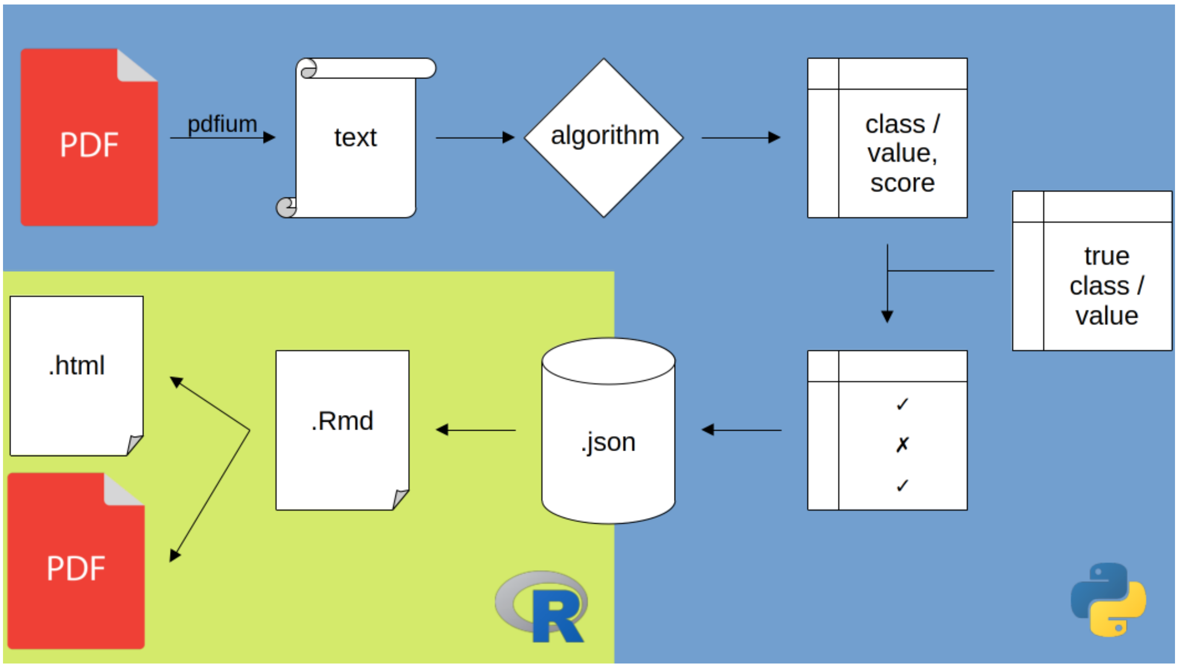


Figure 4.3: Showing the processing steps from input data to the results in this thesis.

The results of this check are saved as json. We save the result for every individual check, as well as calculated performance metrics and the runtime needed to process all texts as batch. This allows us to reevaluate or exclude single results and recalculate the aggregated metrics later. All steps from the text extraction to the result saving are implemented in *Python*.

The process for the information extraction is similar. One difference is, that the algorithm makes multiple predictions per text. Thus, we save not a single prediction and evaluation but a data frame with all predictions and all ground truth values per text. Saving the ground truth values in this data frame is not necessary but allows us a more convenient reevaluation.

4.5 Evaluation and Reporting

We mostly use *R* for evaluation, visualization and reporting. We use the *bookdown* library[1], to create a report, that is linked to our data and thus automatically includes new results and updates all figures and tables. This allows us to run small additional experiments until the very end of thesis writing.

Furthermore, this allows us to create a PDF version as well as a HTML (hyper text markup language) version of the thesis at the same time with low additional effort. The HTML version allows to use some helpful interactive elements as paginated tables with search and sorting capabilities, image light boxes and image sliders. Thus , a lot of information can be offered without occupying pages over pages with tables and figures. The HTML version is more machine readable as well.

4.6 Speedup with vLLM and batching

We run our final experiments with the vLLM library on Python, using its batch processing capabilities. Our first test used the *transformers* library directly and did not use batch processing. Section E.3.3 shows the runtime reduction that is achieved with the final setup.

[1] For the next project we probably will start using *Quarto* instead of *bookdown*. This is the new reporting framework of Posit, we became aware of too late.

4.7 Hardware normalization

To make the runtime of different LLMs running on different amounts and types of GPUs comparable, we conducted a benchmark running the models Qwen2.5-7B and Qwen2.5-32B with different hardware compositions on the Datexis cluster. Figure 4.4 shows the runtime for classifying 100 pages with the multi-class approach, providing three random examples for the in-context learning.

The classification time with Qwen2.5-32B on GPUs of type B200 is a little faster than running Qwen2.5-7B on the same amount of A100 GPUs. We calculate normalized runtimes for our experiments, based on these runtime measures for small and larger LLMs on different types and numbers of GPUs. A minute of computation on a single B200 is comparable to 4:30 minutes of computation on a single A100.

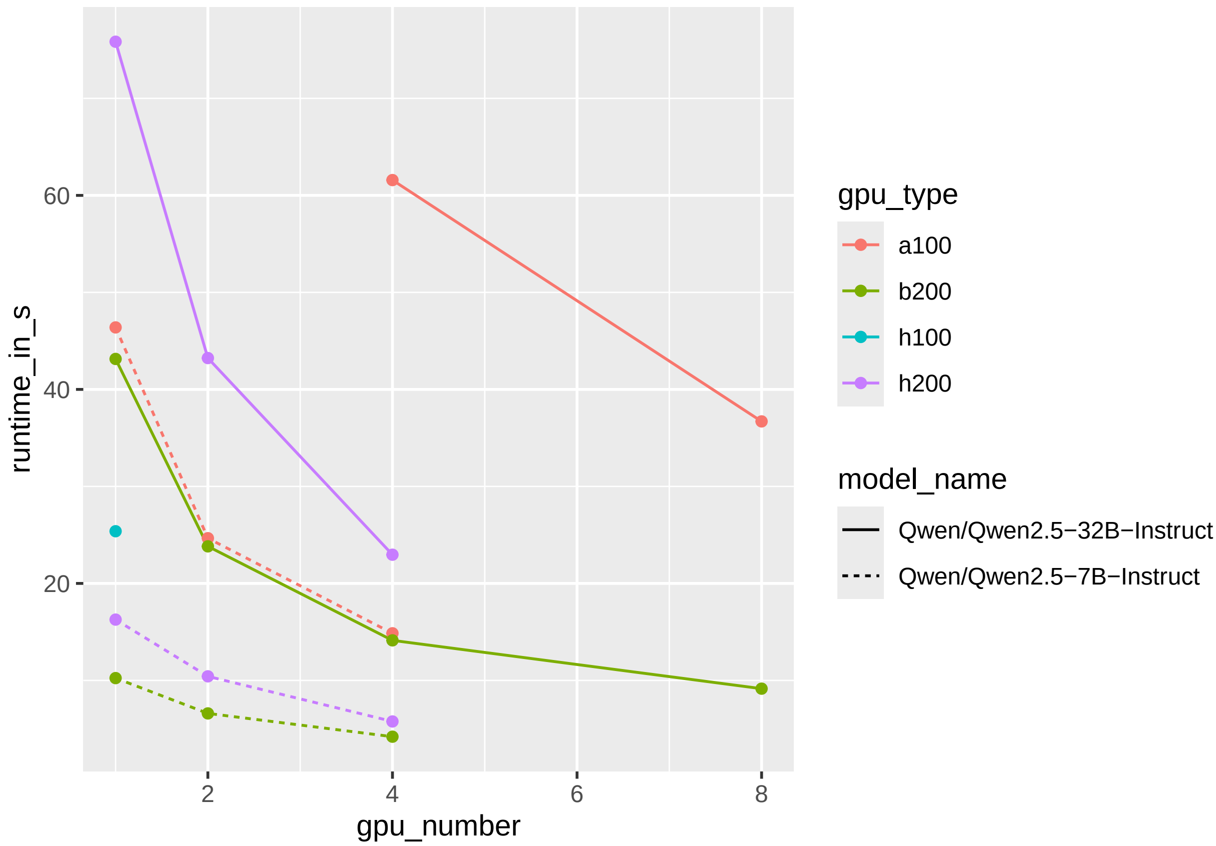


Figure 4.4: Showing the runtime to classify 100 pages with the multi-class approach, providing three random examples for the in-context learning.

4.8 Summary

Chapter 5

Results

This chapter presents the main results for the two main and our side research questions of this thesis:

Q1 How can we use LLMs effectively to locate specific information in a financial report?

Q2 How can we use LLMs effectively to extract this information from the document?

Q3 Can we use additional information from the extraction process to guide the user on which values need to be checked and which can be trusted as they are?

Section 5.1 presents the results for the first research question. Section 5.2 presents the results for the second question. Section 5.3 shows the results for the side research question. Finally, we will summarize all results in Section 5.4.

Each section will start with an overview about the specific sub tasks as well as about the models, methods and data used to investigate the research question.

This chapter focuses on the presentation of results, that answer the question, if an approach can be used to solve a task. We discuss additional findings in Chapter 6. The presentation of those additional results can be found in the appendix. We refer to the chapters of the appendix, that document our investigation and the results of the sub tasks in detail, at the beginning of each of the following sections.

5.1 Page identification

The first research question asks, how LLMs can be used, to effectively locate specific information in a financial report. The task for this thesis is to identify the pages where the balance sheet (*Bilanz*) and the profit-and-loss-and-statement (*Gewinn- und Verlustrechnung, GuV*) are located. The balance sheet is composed of two tables showing the assets (*Aktiva*) and liabilities (*Passiva*) of a company. Often, these two tables are on separate pages. Hereafter, the German terms **Aktiva**, **Passiva** and **GuV** will be used.

H. Li, Gao, et al. (2023) describes two ways to identify the relevant pages (see Figure 1.2). For longer documents they propose to use the TOC to determine a page range that includes the information of interest. In addition, they develop target specific regular expressions and rules to filter out irrelevant pages[1]. The result of this “Page Range Refinement” is then passed to the LLM to extract information from.

This section presents the results of four approaches to identify the page[2] of interest. The detailed report on all experiments conducted and their results can be found in the appendix:

[1]Our personal opinion: Developing well performing regular expressions can be a very tedious and setting appropriate rules requires some domain knowledge. It can be worth the effort if there are a lot of documents with similar information to extract. For this thesis it took multiple months. At least, now there is kind of a pipeline one can reuse, exchanging the rules and key word lists. Thus the next similar task should be solved faster.

[2]In some cases the information of interest is spanning two pages. These rare cases are not covered from the approaches presented here, yet.

- Subsection A.1 presents the findings of a page range refinement using a list of key words with a regular expression.
- Subsection A.2 presents the findings of a TOC understanding approach
- Subsection A.3 presents the findings of a text classification approach using LLMs.
- Subsection A.4 presents the findings of a term-frequency approach.

We compare and summarize the results in Subsection 5.1.2. Subsection 6.3.1.5 proposes an efficient combination of approaches to solve the task of this thesis and discusses its limitations.

Dataset describtion For the page identification task, we selected companies (mainly) from the first row of Figure 1.1, to build the ground truth from. The main motivation is that the documents of a companies within a category are more similar to each other, than to documents of companies of other categories. For the chosen companies all available annual reports are selected. Since one of the companies mainly published documents that require OCR preprocessing, we include the documents of a second company for this category.

5

5.1.1 Approaches

For all approaches presented in this section except the TOC understanding approach, the page identification task is formulated as a classification problem. This subsection briefly describes our approaches. A detailed report can be found in teh Appendix A.

Regular expressions We develop multiple sets of regular expressions and filter out all pages that do not fulfill all regular expressions of a given set. There are different sets for each target type, **Aktiva**, **Passiva** and **GuV**. The sets also differ in how versatile they can cope with additional white space introduced by a imperfect text extraction and how many different words for a given term are accepted.

Table of Contents Understanding We use a LLM to extract the TOC from the first pages from a document or use the embedded TOC and prompt a LLM to identify the pages where the **Aktiva**, **Passiva** and **GuV** are located.

Large Language Model Classification We use LLMs to classify if the text extract of a given page contains a **Aktiva**, **Passiva** or **GuV** table or something else. We test binary classification and a multi-classification approach. The reported confidence scores can be used to form a ranking, which text extract might be most similar to the target type.

We test a wide range of open-weight models and compare different prompting techniques. Besides a zero shot approach, we test few-shot in-context learning with examples that are either chosen randomly or retrieved based on their vector similarity. Finally, we test passing the legal text instead of examples from a annual report.

Term Frequency Ranking We use normalized term frequencies and normalized float frequency as features for a classification with a random forest. The predicted scores are used to build a ranking, which page most probably contains the target pages. Undersampling is used during training, to handle the unbalanced data.

5.1.2 Comparison

This subsection presents the performance and efficiency for all four presented approaches and compares it with the the results a human achieves manually

Prediction performance Table 5.1 shows the best performance achieved by the four presented approaches regarding precision, recall and F1 score.

Llama 4 Scout reaches the best F1 score for the target types **Akiva** and **Passiva** in the multi-class classification approach. For **GuV** the best F1 score (0.985) is found with Ministral-8B-Instruct in the binary classification approach. Llama 4 Scout reaches a F1 score of 0.971 for target type **GuV** and multi-class classification.

In the dataset preparation for the table extraction task (see section 4.3.2) 107 **Aktiva** pages have been selected. In this manual process we made two mistakes, accidently selecting one **Passiva** and one **GuV** page. Thus the human baseline to compete with is 0.981. Thus, Llama 4 Scout is more precise than us.

Furthermore, Llama 4 Scout reaches a recall of 1.0 for all target types. This means, the results can be used downstream, even though the precision is not always perfect. The pages classified as target can be double checked by a human, without missing any page.

The performance of approaches that do not utilize LLMs is substantially lower. Only the term-frequency approach yields results that are suitable for downstream use, as it achieves a recall of 1.0. Table 5.2 shows the results of the top k recall for the term-frequency and LLM approaches. For the term-frequency approach the ranking is based on the scores of the random forest classifier. For the LLMs the ranking is based on the calculated confidence scores.

5

The LLMs always rate the correct **GuV** page highest. With Llama Scout 4 we find all target pages within the first two ranked pages. For the term-frequency approach a human sometimes has to check up to five pages.

Energy usage and runtime Table 5.3 shows the runtime in seconds per document, estimated energy consumption in Joule per document and costs in **CENTS per 1000 documents**. The runtime for the LLMs was normalized to seconds on a NVIDIA B200 and thus the TDP (thermal design power) of 700 W is used to calculate the energy consumption. For the other approaches, running on a laptop (see section E.2) a TDP of 28 Watts is used. For manual work by a human additional 60 W are added for the screen used. It is assumed that the LLM is hosted locally.

Table 5.3 shows, that the regular expression approach is fastest and consumes least energy. Nevertheless, since the results are not sufficient, another approach has to be chosen, if the amount of manual labor should be reduced for the human inn the loop.

Second place, regarding all these criteria, is the term-frequency approach, which guarantees a perfect recall, while reducing the number of pages to investigate to five per target type. This is similar to the number of pages a human has to investigate to find the TOC of the document. And it is a reduction to 7.4 % of the average 67 pages the documents in this dataset have. The costs are still negligible.

The LLM approaches have the highest runtime and energy consumption. This is the case, because they process every page with very computational demanding algorithms. For the TOC approach LLMs are used as well, but they process far less of the documents pages. Thus, their energy consumption is lower.

Since all approaches but the manual identification need the text extract, this runtime and energy consumption are also not listed (but low).

5.2 Information extraction

The second research question asks, how LLMs can be used, to effectively extract specific information from a financial report. The task for this thesis is to extract the numeric values for the assets (*Aktiva*) table, which is part of the balance sheet (*Bilanz*). Hereafter, the German term **Aktiva** will be used. We are limiting the scope even further than in Subsection 5.1, because it takes more time to manually create the first reference dataset.

Structured output We use a strict schema for the extraction process that is derived from the legal text (HGB, 2025, Section 266). Actually, there are three types of verbosity, that are defined in the law. Smaller

5

Table 5.1: Comparing page identification perfromance among all four approaches.

approach	strategy	precision	recall	F1
Aktiva				
Regex	exhaustive	0.132	0.997	0.233
TOC	machine readable	0.6	0.256	0.359
TOC	combi	0.338	0.268	0.299
LLM binary	Ministral-8B-Instruct-2410, 3_rag_examples	0.906	0.939	0.922
LLM multi-class	Ministral-8B-Instruct-2410, 3_rag_examples	0.987	0.937	0.961
LLM multi-class	Llama-4-Scout-17B-16E-Instruct, 3_rag_examples	**1**	**1**	**1**
TF	high recall	0.193	**1**	0.324
human	manual	NA	NA	0.981
GuV				
Regex	exhaustive restricted	0.21	**1**	0.35
TOC	machine readable	0.446	0.245	0.316
TOC	combi	0.378	0.363	0.37
LLM binary	Ministral-8B-Instruct-2410, 3_rag_examples	**0.981**	0.99	**0.985**
LLM multi-class	Ministral-8B-Instruct-2410, 3_rag_examples	0.903	**1**	0.949
LLM multi-class	Llama-4-Scout-17B-16E-Instruct, 3_rag_examples	0.944	**1**	0.971
TF	high recall	0.131	**1**	0.232
Passiva				
Regex	exhaustive	0.13	0.993	0.23
TOC	machine readable	0.5	0.21	0.296
TOC	combi	0.281	0.222	0.248
LLM binary	Ministral-8B-Instruct-2410, 3_rag_examples	0.937	0.914	0.925
LLM multi-class	Ministral-8B-Instruct-2410, 3_rag_examples	**1**	0.761	0.864
LLM multi-class	Llama-4-Scout-17B-16E-Instruct, 3_rag_examples	0.985	**1**	**0.993**
TF	high recall	0.21	**1**	0.347

Table 5.2: Comparing the top k recall for the termfrequency and LLM approaches.

approach	strategy	top 1 recall	k for full recall
Aktiva			
LLM binary	Ministral-8B-Instruct-2410, 3_rag_examples	0.959	2
LLM multi-class	Ministral-8B-Instruct-2410, 3_rag_examples	0.932	3
LLM multi-class	Llama-4-Scout-17B-16E-Instruct, 3_rag_examples	**1**	**1**
TF	high recall	0.826	5
GuV			
LLM binary	Ministral-8B-Instruct-2410, 3_rag_examples	**1**	**1**
LLM multi-class	Ministral-8B-Instruct-2410, 3_rag_examples	**1**	**1**
LLM multi-class	Llama-4-Scout-17B-16E-Instruct, 3_rag_examples	**1**	**1**
TF	high recall	0.7	5
Passiva			
LLM binary	Ministral-8B-Instruct-2410, 3_rag_examples	0.932	**2**
LLM multi-class	Ministral-8B-Instruct-2410, 3_rag_examples	0.824	3
LLM multi-class	Llama-4-Scout-17B-16E-Instruct, 3_rag_examples	**0.973**	**2**
TF	high recall	0.808	5

Table 5.3: Comparing page identification efficiency among all four approaches.

approach	strategy	runtime per document in s	energy in J	costs in CENTS per 1000 documents
Regex	exhaustive	**0.005**	**0.151**	**0.001**
TOC	machine readable	0.202	141.580	1.062
TOC	text based	1.939	1357.534	10.182
LLM binary	Ministral-8B-Instruct-2410, 3_rag_examples	35.851	25095.946	188.220
LLM multi-class	Ministral-8B-Instruct-2410, 3_rag_examples	18.905	13233.784	99.253
LLM multi-class	Llama-4-Scout-17B-16E-Instruct, 3_rag_examples	60.149	42104.054	315.780
TF	high recall	0.138	3.859	0.029
human	manual	61.000	5368.000	40.260

companies are permitted to create less detailed balance sheets. Our schema is created based on the most detailed level. This is the form most often found in the document base[3].

Using a strict schema has advantages for processing the results in downstream tasks - i.e. for adding the results to a relational database. It is also easier to compare the results with a ground truth, if the names of all rows and their order is fixed. The schema is defined as ebnf (extended Backus–Naur form) grammar and passed as an argument to vLLM.

Gound truth dataset For the information extraction we use two datasets. First, a collection of 107 real **Aktiva** tables is created, going through two sampling iterations. In the first iteration a single report is selected for each company. In addition, all available reports from the first listed company are chosen, to test an in-company learning approach. In the second iteration more reports of the other companies are added, to increase the ground truth size and to allow testing the in-company approach for all companies.

Second, a dataset of 16_504 synthetic **Aktiva** tables is created. We generate these tables based on the extraction schema and fill them with random numeric values. Different table characteristics are systematically combined, to investigate potential effects of these features on the extraction performance. The tables are created as PDF, HTML and Markdown files each. This dataset allows to estimate the extraction performance, if there are no unknown row identifiers present.

5.2.1 Approaches

Regular expressions We use regular expressions to extract the numeric values for matching row identifiers. The regular expressions handle line breaks between words in the row identifiers, but not within a word. They can handle multiple signs of white space. Besides that, they try to fully match the labels from the legal text with the text extract, ignoring upper case. They extract numbers with "." as thousands separator.

Real tables We use LLMs to extract the numeric values of real **Aktiva** tables with restricted generation. The LLM has to group row identifiers and corresponding numeric values and match the row identifier with the labels of the schema. If a row identifier is unknown, the values have to be discarded. If a label is not present among the row identifiers, the model predicts *null*. All values are extracted in one pass. We do not include any instruction, how to proceed with currency units, that might be given for certain columns.

We test a wide range of open-weight models and compare different prompting techniques. Besides a zero shot approach we test few-shot in-context learning with examples that are either chosen randomly or retrieved based on their vector similarity. Finally, we test passing a synthetic **Aktiva** table as example. We test models from OpenAIs GPT family in addition to the open-weight models.

[3]Unfortunately, well known companies as BVG and BSR publish a less detailed form. Thus, their documents are not included in the document base for this task.

Synthetic tables We use LLMs to extract the numeric values of synthetic **Aktiva** tables with restricted generation. The procedure is identical as with the real **Aktiva** tables. We extract all values with and without an explicit instruction on how to proceed with currency units. We limit our test on the open-weight models.

Hybrid approach We use LLMs to extract the numeric values of real **Aktiva** tables with restricted generation, providing examples from synthetic **Aktiva** tables. The procedure is identical as with the real **Aktiva** tables. We extract all values with and without an explicit instruction on how to proceed with currency units. We limit our test on the open-weight models.

5.2.2 Comparison

5

This subsection compares the results for the table extraction tasks. It will discuss the findings about performance and runtime and compare it with the results a human may achieve wit manual labor. The detailed report on all experiments conducted in their results can be found in the Appendix B. The evaluation of feature importance can be found in Appendix D.

Performance Table 5.4 summarizes the mean percentage of correct predictions total for all approaches and both types of **Aktiva** tables. The highest baseline for the extraction tasks is set by our own manual performance. We achieve 97.6 % correct extracted values on the real **Aktiva** tables. The regex performance on the synthetic **Aktiva** tables comes close but on real **Aktiva** tables it is far off.

The mean performance of Qwen3-235B does not match our baseline on the real **Akiva** tables. But its median performance already is 100 %.

On synthetic tables its mean performance is almost perfect, if currency units get respected. With HTML documents we find 100 % correct predictions. With Markdown documents we find 99.9 % correct predictions as well. Figure G.9 shows, that the better performance on the synthetic tables is found for almost all models.

Qwen3-8B performed best among the small models LLMs but shows over 4 % more wrong predictions than Qwen3-235B.

Using synthetic examples results in worse performance. But it can be used to show how to handle currency units.

Runtime Extracting the values from all 106 tables took Qwen3-235B around six minutes. Thus, excluding the setup time for the LLM, Qwen3-235B-A22B-Instruct is around 100 times faster than a human.

Table 5.4: Comparing the mean percentage of correct predictions total among all approaches and table types.

approach	strategy	mean_percentage_correct_total
real		
human	manual	**97.6**
regex		68.6
llm	Qwen3-235B, top_5_rag_examples	97
llm	Qwen3-8B, top_5_rag_examples	92.7
llm	Qwen3-235B, top_5_rag_examples, synth examples	91.8
synth		
regex		96.9
llm	Qwen3-235B, top_5_rag_examples, respect_units	**99.9**
llm	Qwen3-8B, top_5_rag_examples	94.6

Hypotheses The predictor that shows a strong effect in all approaches is currency unit. Reflecting this in the table extraction is a key factor to optimize the performance. For the approaches that use LLMs most of the model and method related variables showed a strong effect. Using a versatile model and providing good learning examples is mandatory.

The findings from approaches utilizing synthetic tables suggest that the input format can have a substantial influence on extraction outcomes. It appears important to prevent errors during text extraction, and converting the extracted text to HTML may help resolve remaining ambiguities. However, it remains unclear whether a perfect text extract would perform as well as HTML or Markdown formats.

5.3 Error rate guidance

The side research question asks, if it is possible to guide the users attention to predictions that have a higher empirical rate of errors. In this thesis we focus on the confidence score calculated from the reported log probabilities.

Subsection 5.3.1 presents the results found regarding our side research question for the page identification task. Subsection 5.3.2 presents the results found regarding our side research question for the information extraction task. The detailed report can be found in the Appendix C.

5.3.1 Page identification

We find, that the confidence score can be used in the page identification task, to identify confidence intervals, that contain no or only a few errors. The amount of predictions in these intervals varies among models, classification task and target class.

Binary classification Distinguishing correct and wrong classifications based on the confidence score is working well for the responses of Ministral-8B-Instruct. But for other models, e.g. from the Qwen family, it works worse. This is possible, because Ministral-8B-Instruct reports confidence scores over a wide range, while models from the Qwen 2.5 family report always high confidence.

Thus, it is possible to define a wide range of confidence intervals, where we find a empirical error rate of zero for Ministral. For Qwen 2.5 32B we find the highest confidence interval containing some mistakes, But it is still good with less 1 %. Most predictions are in the highest confidence score interval for both models.

Multi-class classification Distinguishing correct and wrong classifications based on the confidence score is still working well for the responses of Ministral-8B-Instruct, predicting **Aktiva** and **GuV**. For **Passiva** we find a single wrong prediciton in the highest confidence interval. For Llama 4 Scout it is working for the target classes **Aktiva** and **Passiva**. For Qwen 2.5 32B it works worst among those three - still well performing - models.

For Ministral we find a empirical error rate of zero for the highest confidence interval, except for **Passiva**. For Llama we find an error rate lesst than 1 % for all classes. For Qwen 2.5 32B only **Passiva** has a high confidence interval with less than 1 % error rate. Most predictions are in the highest confidence score interval for all models again.

5.3.2 Information extraction

We find, that the confidence score can not be used alone in the information extraction task, to identify confidence intervals, that contain no or only a few errors. There are only few exceptions, where we achieve an error rate of under 1 % over all annual reports. We just report the results of the best performing model here.

Real tables For the best predicting model Qwen3-235B-A22B-Instruct, we find an empirical error rate of 1.3 % for predicting a missing value and 3.3 % for predicting numeric values. Almost all predictions fall in the highest confidence interval. Making the interval width smaller, does not result in intervals with lower error rate.

Synthetic tables For the best predicting model Qwen3-235B-A22B-Instruct, we find an empirical error rate below 1 % for predicting a missing value and numeric values, if we explicitly instruct the model to handle currency units. If the input format is not a text extracted from a PDF file, but perfect HTML code the error rate gets 0 %. With perfect Markdown code the error rate is above 0 % but below 1 %.

Hybrid approach For the best predicting model Qwen3-235B-A22B-Instruct, we find an empirical error rate below 4 % for predicting a missing value and around 20 % to 26 % for predicting numeric values. The values is lower, if we explicitly instruct the model to handle currency units.

5.4 Summary

In summary, the results presented in this chapter demonstrate the effectiveness of LLMs for both page identification and information extraction tasks in financial reports. LLM-based approaches consistently outperform traditional methods, achieving high recall and precision, and enabling efficient automation of information extraction. The experiments further highlight the importance of input format and model selection, as well as the potential for guiding user attention through confidence scores.

These findings provide a solid foundation for the subsequent discussion, where we interpret the results in the context of our research questions and hypotheses, analyze sources of error, and explore practical implications for future applications.

Chapter 6

Discussion

This chapter consits of the main sections as follows:

Section 6.1 interprets the results found in relation to our research questions and hypotheses. It compares the findings with findings in previous work and names unexpected results.

Section 6.2 contains our error analysis and discusses interesting findings under a researchers perspective.

Section 6.3 describes practical implications and limitations for the planned HITL application derived from the error analysis. It interprets the results from an engineers point of view.

Subsequent, Section 6.4 describes, what we have not investigated or implemented in this thesis. Section 6.5 gives an outlook on the challenges for implementing the system at RHvB. Finally, ethical and practical consideration get presented in Section 6.6.

6.1 Research questions

6.1.1 Page identification

We posed the following research question and hypotheses for the page identification task:

Q1 How can we use LLMs effectively to locate specific information in a financial report?

H1.1: LLMs can be used to locate specific information in a financial report, achieving a high F1 score.

H1.2: LLMs can be combined with other approaches to reduce the energy consumption, without lowering the systems recall.

Results The page identification task is solved by the LLM with higher F1 scores for every target class than the human reference F1 score. It is solved completely on the created dataset for predicting the class **Aktiva**. In two cases the multi-class classification with Llama 4 Scout is best. For classifying **GuV** the binary classification with Ministral works best.

The prompting strategy *n_rag_examples* often yields best results, especially if in-context learning examples from the same company as the subject of identification company are used. Section 6.2.2.2 is discussing this finding for both tasks.

The term frequency approach reaches a top k recall of 1.0 for a small $k = 5$. It runs 100 times faster than the LLM approach and reduces the average number of pages to classify by 92.6 %. We discuss how to combine the two approaches in Section 6.3.1.5.

Interpretations Both of our hypotheses find support. Thus, we see our research question as answered with the following statement: It is possible to locate specific information in a financial report using LLMs and it can be done more efficient if a term frequency based approach is used for page range refinement in advance.

Comparison with previous work We are able to narrow down the page range to five pages without using a LLM. With the LLM we are guaranteed, to find the correct pages within a range of two pages. Most of the time, the page, ranked highest in the confidence score ranking, is the correct one. H. Li, Gao, et al. (2023) do not present a concrete number of pages, they have to process after page refinement.

The TOC does not work as well as expected. H. Li, Gao, et al. (2023) do not report any issues or a low recall value for their TOC based approach. Thus, we assumed to find a recall value close to 1.0 for the TOC based approach too. Instead we found a recall of 0.27. We discuss difference in the two approaches in Section 6.3.1.3.

Unexpected results We are surprised by the strong performance of Ministral-8B. It performs better, than other models of the Mistral family, that have been released more recent and have a larger parameter count. It shows one of the strongest performances with the *zero_shot* and *law_context*, too.

We are also positive surprised of the strong performance of the recently released members of the Qwen3 family. The older Qwen3 models could not handle the classification tasks at all - with or without thinking mode[1] - while the models of Qwen2.5 performed well. We identified the implementation of dense MoE layers as an architectural difference between the more recent and older Qwen3 models.

But we assume the difference is a different fine tuning process. While the original released models already claimed to deliver "groundbreaking advancements in [...] instruction-following" (*Qwen/Qwen3-4B · Hugging Face*, 2025) we now find a newer version of the Qwen3-4B model that explicitly is tagged as Qwen3-4B-Instruction. It would be interesting, to add this model to the benchmark in future work.

Conclusion

6.1.2 Information extraction

We posed the following research question and hypotheses for the information extraction task:

Q2 How can we use LLMs effectively to extract this information from the document?

We will start discussing the hypotheses group **H2.1** in subsection 6.1.2.1, before we will look at the hypotheses group **H2.2** in subsection 6.1.2.2. The hypotheses of group **H2.1** are:

H2.1a: LLMs can be used to correctly extract multiple numeric values from the assets table.

H2.1b: LLMs can match row identifiers and place the numeric values in the correct target row.

H2.1c: LLMs can identify unmatched row identifiers and report, that the value is missing.

6.1.2.1 Core capabilities

Results The best performing model - Qwen3-235B-A22B-Instruct - almost reaches human performance on real **Aktiva** tables, but is much faster. Both measures, percentage of correct numeric predictions (98.0 %) and F1 score (98.1 %), are not perfect yet and could be improved. It achieves perfect results on synthetic tables provided in HTML format.

The prompting strategy *n_rag_examples* often works best, especially if in-context learning examples from the same company as the subjects of extraction company are used. Section 6.2.2.2 is discussing this finding for both tasks.

[1] The Qwen3 models support two operating modes: A thining mode and a non-thinking mode. The thinking mode should yield better answers in complex tasks and the additional amount of processing can be controlled by setting a thinking budget (Qwen Team, 2025). This thinking budget can be seen as the amount of tokens used for a step wise solution.

Interpretations The high percentage of correct extracted numeric values, shows that Qwen3-235B is not only able to exactly copy their values and to match the row identifiers, but also to perform numeric transformations, respecting the currency units, in many cases. If no numeric transformations are performed, the upper limit for correct numeric extraction is 80.8 %, since 19.2 % of all numeric values have *T€* as currency unit. We discuss this in Section 6.2.2.4.

Furthermore, we can see for the extraction from synthetic **Aktiva** tables, that it is possible to achieve perfect results, if the input is perfectly structured and without unknown row identifiers. In analogy to the concept of *garbage in, garbage out*, we could phrase it as *perfect in, perfect out*. Thus, we show that there is no LLM approach inherent mechanism, that prevents perfect information extraction of numeric values. @ref()

The high F1 score shows, that Qwen3-235B also is able to identify missing row identifiers and correctly returns *null* instead of hallucinating numeric values. We discuss the matter of label matching further in Section 6.2.3.

Thus, we conclude, that all hypotheses of group **H2.1** get supported and the research question can be partially answered with the following statement: LLMs can be used effectively to extract information from the financial reports using in-context learning and a strict schema.

Comparison with previous work H. Li, Gao, et al. (2023) achieve a perfect extraction result, after refining their approach, extracting a total of 152 data points from 8 ACFR (Annual Comprehensive Financial Report)s reports. Before the prompt refining they find 96.1 % correct extracted datapoints. Applying the refined approach on a more heterogeneous sample of 4_000 county year ACFRs, they achieve a performance of 96 % on 80_000 data points.

Thus, our found performance is higher as their initial in sample and final out of sample performance. Since we did not adjust our prompting strategy before the second iteration of the ground truth creation, we argue that our performance should be compared to their out of sample performance.

There are some differences in our prompting approaches, that may have a minor influence on the performance. They include explicit instructions for handling currency units, which we do not for the extraction task with real **Aktiva** tables. But we investigated this with the synthetic and hybrid approach and discuss the findings in Section 6.2.2.4.

H. Li, Gao, et al. (2023) extract three values per call, while we expect 10 to 40 numeric and 18 to 38 *null* predictions per call. They explicitly include an instruction on how to handle missing values and do not report any remarkably higher error rates with missing values. Therefore, we can not compare our F1 score with their results.

Unexpected results We did not expect the LLM to perform any currency units related numeric transformations, without being explicitly instructed, (how) to do this. But it is plausible, that this pattern can be learned during in-context learning. This is especially true, if the examples are chosen according to vector similarity and if documents from the same company can be used. We describe this in more detail in Section 6.2.2.2.

Conclusion

6.1.2.2 Feature effects

The hypotheses of group **H2.2** are:

H2.2a: Model specific features have an effect on the extraction performance.

H2.2.b: Prompt strategy specific features have an effect on the extraction performance.

H2.2c: Table specific features have an effect on the extraction performance.

Results The features, that show an noticeable importance value most often, are:

1. *model_family*
2. *parameter_count*
3. *method_family*
4. *n_examples*
5. *T_in_previous_year*

We can determine an effect direction for the features *parameter_count*, *n_examples* and *T_in_previous_year*. More parameters and more examples are beneficial. The only exception for this is found for the model Llama 4 Maverick. We discuss this case in section 6.2.4. The performance gain gets smaller comparing experiments with three and more provided examples. The prompting strategies *n_random_examples* and *top_n_rag_examples* perform best.

The method specific feature *respect_units* shows meaningful importance for evaluating synthetic tables. Its effect has opposite direction to our assumption. For the hybrid approach it has a small importance, but with an effect direction matching our assumption. We discuss this in Section @ref().

With the synthetic tables we also find that HTML and Markdown are a *input_format* that yield better results. We show in section 6.3.2, that we do not find these benefits for processing real tables from (imperfect) Markdown input.

Interpretations All model and method specific features are meaningful predictors for the information extraction task. Thus, we argue, that **H2.2a** and **H2.2b** find total support. The features *model_family* importance could be an artifact of Llama 4 Mavericks bad performance with multiple in-context learning examples.

Of all the table characteristics only *T_in_previous_year* shows a higher importance. Thus, we argue, that **H2.2c** only gets support for a single feature, where we proposed an effect. There are more features, where we proposed no effect and where no effect is found. These hypotheses also get supported. We assume, that *T_in_year* would get a higher importance, too, if the number of tables, that posses this characteristic, in our sample would be higher.

Compare with previous work H. Li, Gao, et al. (2023) report, that the most frequent errors for the extraction of values from ESG (environmental, social, and governance) reports are based on not correctly handled units. This aligns well with our finding of a high importance value for *T_in_previous_year*. Our results also align with Brown et al. (2020) findings, that the performance might not increase much more by adding additional examples after the second one.

Limitations We failed to reflect the *header_span* and *text_around* characteristics, during the creation of the synthetic HTML and Markdown tables. Thus, we can not evaluate the interaction effect hypotheses with *input_format* for those features. But we found no big importance for *header_span* in any experiment.

We also did not include the fact, if in-context learning examples are from the same company as the subject of extraction, as a method related feature. We show, that this matters in section 6.2.2.2.

Unexpected results We did not expect to find such a drastic example for content rot as we do for Llama 4 Maverick. We are surprised that it is not the same for Llama 4 Scout. Since it has an even larger context window, we wonder, if we will observe context rot, if we increase the number of examples proportionally.

Recommendations We recommend to evaluate the hypotheses for the visual table characteristics once more, when more advanced document parsing methods are used, that process visual information. The method for document parsing should be included as a feature in this evaluation as well.

Conclusion In contrast to the classification task, we find no model family that performs worse in general. More parameters are helpful, but it is more important, that at least one or two examples are provided for in-context learning.

6.1.2.3 Error rate guidance

We posed the following UX inspired side research question and hypotheses:

Q3 Can we use additional information from the extraction process to guide the user on which values need to be checked and which can be trusted as they are?

H3.1: The confidence score can be used to guide the user on which of the identified pages need to be checked and which can be trusted as they are.

H3.2: The confidence score can be used to guide the user on which of the predicted values in the information extraction task need to be checked and which can be trusted as they are.

Results For the page identification task, we find high confidence intervals with (near) zero empirical error rate for well performing models. Ministral-8B shows a wide spread of confidence scores for the page identification task, that can be used to distinguish correct and wrong classifications.

We do not find high confidence intervals with a error rate below 1 % for the information extraction task on real **Aktiva** tables. Explicitly instructing to respect currency units, reduces the empirical error rate (see Appendix C.2.3). But we can find confidence intervals with zero error rate for the information extraction task on synthetic data.

6

Interpretations We find support for the hypothesis **H3.1** but not for hypothesis **H3.2**. Thus we answer the research question as follows: The confidence score can be used to guide users attention for the page identification task, but hardly for the information extraction task.

Compare with previous work The classification task can be seen as a best-of-N selection task with two or four choices. Here, our simple measure for confidence seems to be sufficient.

Generating the information extraction response, could be seen as a repetition of many best-of-N selections tasks. If every digit, the floating point delimiter, and ending the sequence is counted as individual choice, there are 12 possibilities. Choosing, if a number or *null* should be returned is another task with $N = 2$. Even though the number of choices is limited for each decision, chaining those choices seems to yield a less meaningful measure of confidence.

Unexpected results Most models predict high confidence for all most of their predictions, even for the wrong ones.

Conclusion

6.2 Error analysis

6.2.1 Page identification

This section is presenting a brief error analysis for the LLM and term frequency approach. We discuss errors related to context rot in Section 6.2.4, because they occurred in both tasks.

6.2.1.1 LLM approach

The seven wrong classified pages by Llama Scout 4 all contain a table, that is filling at least half a page containing one column with text and many columns with numeric values. Six out of those seven are classified as **GuV** and one as **Passiva**. The page, classified as **Passiva** contains the term *Passiva*.

From our point of view it would be more plausible, if page 63 of '/pvc/Geschaeftsberichte/Berlinovo/ berlinovo_Finanzbericht_2021.pdf' is classified as **Aktiva** because there are a lot of the same row identifiers present. The term frequency approach correctly classifies this page as *other* and ranks it on 8th place for **GuV** and **Passiva** and only on 14th place for **Aktiva**. Thus, our intuition what kind of terms are in the table, is not correct.

To prevent those few incorrect classification, the system may need more examples for pages, that are similar as the target classes but not of that type.

6.2.1.2 Term frequency approach

Inspecting some of the pages that are ranked high and wrongly classified as **Aktiva** by the term frequency based random forest, we find, that the normalized sum of term occurrences, can lead to missclassification. For example, we find a page identified as **Aktiva**, that contains no single table but a long text. The term counts based on the **Aktiva** term list can be found in teh Appendix E.8. The term 'Geschäfts' is present 19 times. The term 'Unternehmen' is present 10 times. :::

Those terms are probably present very often all over the document. Thus, they should get less weight. Additionally it might be better to count, how many of the terms are present in a boolean manner, instead of counting them. Kong et al. (2024) shows, that BM25 can be used effectively to identify relevant tables.

6

We also find pages that are missclassified, because they have a high density of floats. An example for a page with high float density that is not containing an **Aktiva** table can be found in the Appendix in Figure E.4.

We formulate our practical recommendation in Section 6.3.1.3.

6.2.2 Information extraction

- check for halluzination vs wrong placed / repeated numbers
- new lines / splitted lines
- test synthetic hypothesis with pymupdf extract
- 2.4 % wrong gold standard creation
- errors from wrong formatted numbers
- errors from wrong / unclear entity mapping
- OpenAI not followed the schema strictly
- Umlaute (ggole models?)

6.2.2.1 Real tables

6.2.2.2 Company specific results

Figure 6.1 shows, the precision and recall values for predicting a missing value and the percentage of correct numeric predictions for Qwen3-235B for each company. The number after the company name, as well as the color of the boxes indicate, how many of the numeric columns have *T€* as currency unit. The crosses indicate the individual scores per document. The teal crosses represent predictions, if examples from the same company are used for the *top n rag* prompting strategy. Red ones represent predictions, where this is not the case.

One can see, that Qwen3-235B yields perfect predictions for the majority of the companies. This is especially true, if we only consider the teal crosses. The predictions improve for most companies, if examples from the same company are used for in-context learning. It is especially helpful for handling the single numeric column with *T€* for *Deutsche Klassenlotterie*. It also helps with the two columns with *T€* for *Gewobag*, even though the other examples have not *T€* present.

Figure 6.1: Comparing the F1 score for predicting the missingness of a value for OpenAi's LLMs with some Qwen 3 models. The green crosses indicate results where a model has predicted only numeric values even though there have been missing values.

It seems a little harmful for *WBM GmbH* and can not solve the problems for numeric prediction for *Helmholtz Zentrum GmbH* and *Berliner Stadtgüter*. It improved the precision for *Rundfunk Berlin-Brandenburg* and the recall of *Stadt und Land GmbH* and *Partner für Deutschland*.

Table 6.1 shows the performance of Qwen3-235B for the *top n rag* and *n random* example strategies and distinguishes based on the fact, if examples from the same company can be used for in-context learning. The achieved percentage of correct predictions total is highest, if examples form the same company are used. It is even higher than the human reference score.

If using examples form the same company is not allowed, it seems better to use random examples. This is probably the case, because this comes with a higher heterogeneity. High homogeneity among the learning examples from other companies might demonstrate patterns, that are not correct for the company the target document comes from. The pattern, that the same company in-context learning yields best results is true for all models. The order of the results with random examples or examples from other companies varies among the models.

6.2.2.3 Synthetic tables

The 0.1 % incorrect predictions on synthetic tables from the PDF documents could be caused by faulty text extracts by *pdfium*. But the Markdown input is without any flaws and resulted in 0.1 % errors as well.

6.2.2.4 Numeric transformations

synth and hybrid

This section discusses the surprisingly high percentage of correct extracted numeric values. If no numeric transformation is performed, 80.8 % is the upper limit. We assume, that the behavior to transform the numeric values without being instructed to do so, is nothing learned during fine tuning, because the percentage of correct extracted numeric values for the *zero_shot* performance (0.763) and the *static_example* performance (0.803) are below the expected limit. The static example demonstrates no numeric transformation and the observed performance is found to be almost as high as the expected upper limit.

Table 6.1 shows, that this limit is overcome with all other prompting strategies. It is plausible for the *top_n_rag_examples* strategy, if examples from the same company are used. But it implies, that almost always there is at least one example, among the five random examples presented, that shows this transformation. And it implies further, that the model recognizes, that the pattern demonstrated by a minority of examples is to apply for the subject of the current extraction task. For future work the provided examples should be logged as well, to investigate this implication.

We tested the influence of providing examples, that show how to perform the numeric transformation, together with an explicit instruction to do so, in the hybrid approach (see section B.2.3). The hybrid approach uses synthetic tables as examples for extracting information from real examples. We show, that Qwen3-235B is capable to learn how to extract numeric values respecting the currency unit and is able to transfer this knowledge from examples with two columns with *T€* to tables that only have one column with *T€*. This is not possible for all models, e.g. not for Ministral-8B. We also find models that over generalize this pattern and achieve worse performance for extraction subjects without *T€* as currency unit - i.e. Qwen3-8B.

Table 6.1: Comparing the performance of Qwen3235B for the best approaches depending on the circumstance if examples from the same company can be used for learning.

model	method	same company	mean numeric	mean F1	mean total
Qwen3-235B-A22B-Instruct-2507-FP8	top_3_rag_examples	FALSE	0.972	0.972	0.959
Qwen3-235B-A22B-Instruct-2507-FP8	top_5_rag_examples	TRUE	**0.990**	**0.987**	**0.982**
Qwen3-235B-A22B-Instruct-2507-FP8	5_random_examples	NA	0.982	0.975	0.966

We also see in our hybrid approach experiments, that the LLM benefit more from the real examples for extracting the numeric values than for label matching. Comparing Figure G.28 and Figure G.29 shows that LLMs yield more consistently good results for predicting *null* learning from synthetic examples than extracting the correct numeric value.

In general the approach seems not be prone to hallucinations. When we accidentally tried to extract values from a **Passiva** and **GuV** table, no prediction was made, because non of the row identifiers matches our strict schema.

We have not specifically investigated, if an annual report of the following or previous year is more helpful than years further away. This would be plausible, because it serves the correct numeric and missing values for half of the information extract perfectly formatted as json. Theoretically it should become trivial to extract all values correct and transform it accordingly to the currency units, if the previous and following years annual reports are presented as examples for the in-context learning. This should be investigated in future work.

predictions for barrierefreie documents of WBM empty

An odd text marking order in a PDF by dragging the mouse is no indicator for a bad text extract.

6.2.2.5 Regular expression approach

We find, that the regular expression approaches performance on the synthetic dataset is highly influenced by the extraction library used. For the real data we find no difference. Figure B.1 B shows, that the regex approach on text extracted with *pdfium* especially has a wider precision range. The number of wrong extracted numeric values is a little as well.

A reason for this might be incorrect extracted texts. We find, that there are issues with missing (or additional) white space, misplaced line breaks and an extraction of the text column first, followed by the numeric columns. You can find examples for those types of incorrect extracted texts by three different PDF extraction libraries in section E.7. One of the examples is found with the real table dataset, while two are found with the synthetic tables.

The random white space and line breaks could be handles by adjusting the regular expression for the label matching. Missing white space between numeric values could also be handled by adjusting the regex. The misplaced columns would need more advanced reprocessing strategies. But all those error ypes shouel influence the recall and not the precision.

A possible explanation for a small spread in precision with both PDF extraction backends could be the duplicated row identifier *Geleistete Anzahlungen*. Our simple approach matches this row always with the first occurrence. This means, if the second occurrence is given in the ground truth, but not the first one, our approach would create a false positive result. The percentage of false positive example is then determining the precision value and is linked to the number of total rows in the ground truth. But we do not see sucha pattern for the *pymupdf* results at all.

A reason for wrong numeric values are rows, where the summed value is given in the next column, but same row, as the individual values. In this case the approach selects one individual value and a sum, instead of two individual values.

- synthetic tables have been generated with cell lines because this should have improved the performance of a table extraction approach (not conducted)- maybe this is confusing pdfium? Or the zoom level?

6.2.2.6 OpenAI models

We could not use a strict schema neither with OpenAIs closed-weight GPT models nor with the new open-weight OSS models. Figure B.3 shows, that this results in many predictions with no or few predictions, where the row identifier matched with our ground truth. It might also be, that the models simply made less predictions than the expected 29 rows. We find this especially for the *nano* models. We even found some predictions, where some rows have been predicted multiple time, resulting in more than the expected 29 rows.

Figure B.2 also shows, that we find responses, where no single *null* value is returned. This means, that the models hallucinate numeric values. This is true for the large GPT-4-1 as well, if it does not get three examples for in-context learning. In contrast, GPT-5-mini and gpt-oss-120B are (almost) without hallucinated values.

Nevertheless, even without a strict schema, we find GPT-4-1 and GPT-5 performing almost as well as Qwen3-235B. This shows, that a guided decoding approach could also work and that a strict schema is not necessary for larger models.

Unfortunately, we could not use batch processing with the closed-weight models resulting in run times of 30 to 135 minutes. Interestingly, GPT-5-nano had the longest runtime and produced much more output tokens as the other models. This is similar to the gpt-oss models that use the new harmony response format, that creates many tokens in a chain of thought stream, before it returns the tokens for the requested json table. This might bring insights in the models processes, but increases the response time a lot. We would welcome the possibility to just get the final answer, as we can disable thinking in Qwen3.

6.2.3 Ground truth creation

GPT-Suggestion: Reflect on the challenges and lessons learned during ground truth creation, including how LLM predictions helped uncover errors in the gold standard and the iterative improvement process.

During the second pass of the ground truth creation for the information extraction task we find, that 2.4 % of the values differ among the previously created gold standard and the results of the second pass. In the first pass the values are copied manually, while the results of the second pass are LLMs predictions, that we double checked. We find 75 values differing in the 24 documents that are part of both data collections. Table 6.2 shows the nature of errors and their counts. Most errors are distributed in the *omission* classes.

Errors of this type result from an inconsistent coding process. In one pass a value might have been included, while it is not included in the other pass. Or the value is matched to different row identifiers during the two passes. To resolve this kind of errors a strict and detailed coding manual is necessary. Additionally, the coding should be done from experts of the field instead of the data scientist.

Inspecting the predictions of the first experiments for the classification task, yielded interesting information, tool. Qwen 2.5 consistently classified some pages right after of the listed **GuV** pages to be of type **GuV** as well. And it was correct. For the company *IBB* the **GuV** spreads over two pages. This led to an adjustment of the ground truth, including all pages of tables, that span multiple pages.

6.2.4 Context rot

We reported the worsening performance of Llama 4 Maverick, when it gets to many examples presented in both main tasks. Since we used the FP8 version, we tested if this is a problem of to low precision in the calculation. But we find the same behavior with the FP16 version. It is the only model we detect the issue of *context rot* for.

Context rot is a term introduces in (Kelly Hong & Anton Troynikov, 2025) technical report. The investigated an advanced *Needle in the Haystack* problem, including distractors and requiring the LLMs to find semantic similarity instead of exact term matching. They find that the accuracy often starts to decrease with 10 k input tokens and more.

Table 6.2: Showing the nature of errors and their counts. Errors with multiple difference have Levenshtein distance greater one.

error_type	count
ommited in first pass	29
ommited in second pass	20
multiple differences	13
missing digit	10
swapped digits	2
comma instead of dot	1

Meta claims that Llama 4 Maverick has a context length 1 M (Llama 4 Scout even 10 M), where other models often ar limited to 128 k or 32 k or less. We limited our input token length to 32 k in most cases and reached this limit multiple times. We find it remarkable, that Llama 4 Maverick already shows *context rot* at inputs of length 10 k - 100 times shorter than their context window.

Levy et al. (2024) show a notable degradation in LLMs' reasoning performance at much shorter input lengths than their technical maximum. They also show, that Mistral achieves the highest accuracy, when the relevant information is at the end of the prompt. We are not sure, how to relate these findings, since we do not include irrelevant information.

6.2.4.1 Page identification

Figure 6.2 shows the amount of correct (matching) and incorrect classifications by Llama 4 Maverick for the binary classification tasks ordered by target type and method. One can see, that the *n_rag_example* strategy starts predicting the target class too often with increased number of examples. This behavior is not observed for the *n_random_examples* strategy.

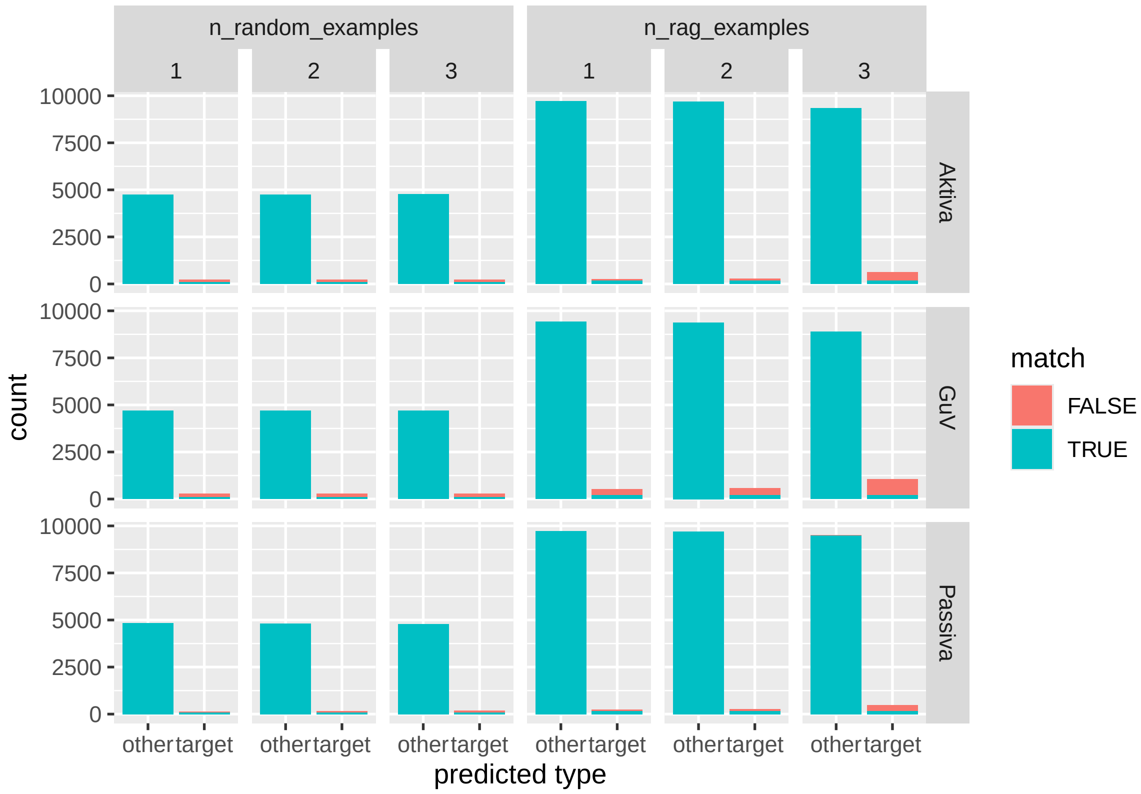


Figure 6.2: Comparing the amount of correct classifications by Llama 4 Maverick for the binary classification tasks ordered by target type and method. With increased number of examples the n-rag-example strategy starts predicting the target class too often.

Figure 6.3 is showing the confusion matrices for the multi-class classification with Llama 4 Maverick grouped by *method_family* and *n_examples*. Teal bordered tiles are correct predictions and red bordered tiles represent wrong predictions. The number is showing the percentage of classifications by the LLM of a certain type (*predicted_type*) based on the true count of observations with that type (*type*). They sum up to one column-wise.

We can see, that the *n_rag_example* strategy starts to predict **GuV** too often, when presented with two or more examples. We observe the same for the *n_random_examples* strategy starting from three provided examples. The LLM is not just over-predicting **GuV**, but also other target classes. The over-prediction rate for *other* is lowest. Those pages often have no page filling table and thus are more different from the target classes and easier to distinguish (for a human).

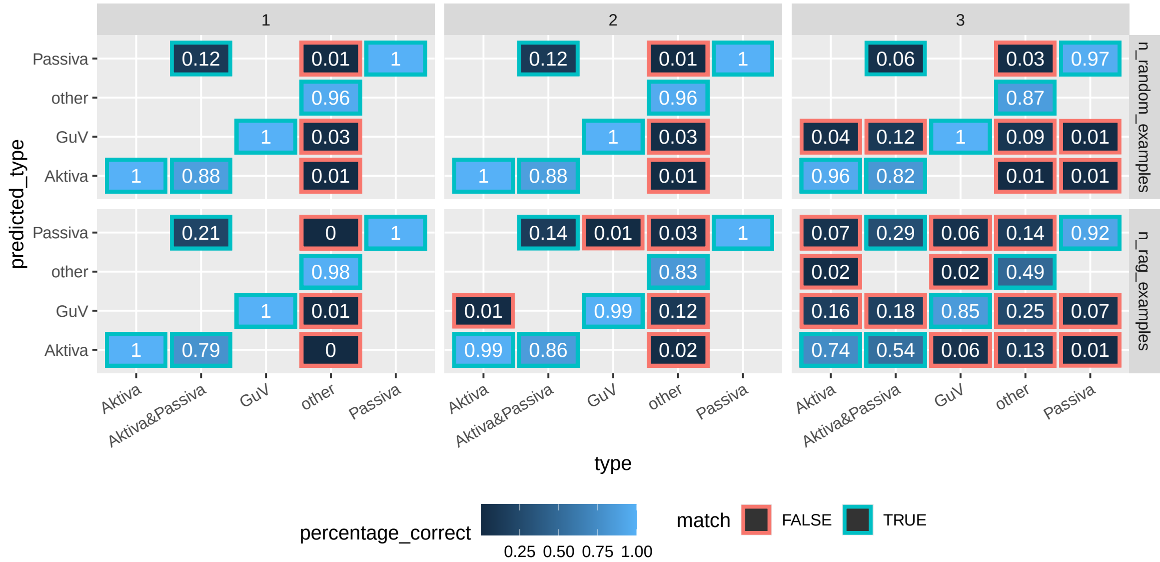


Figure 6.3: Showing the confusion matrices for the multi-class classification with Llama 4 Maverick grouped by method-family and n-examples.

A possible explanation fro over-predicting **GuV** most might be, that the examples for **GuV** are presented first, because we just iterate over the *phrase_dict* dictionary (see code below). Liu et al. (2023) describe a behavior of LLMs, that they are better with identifying relevant information, when it is placed in the beginning or end of the context. Since all examples are provided in the same manner, examples for classes other than the target class, could be interpreted as distractors with a high similarity. Kelly Hong & Anton Troynikov (2025) shows, that in the presence of similar LLMs' performance can degrade quickly. They do not present results, if the position of the picked distractor is important. Thus, we formulate the hypotheses for future investigation:

1. LLMs tend to choose distractors at the beginning or end of the prompt.
2. LLMs tend to choose distractors that apprear first or last.
3. There is an interaction effect, with the position, where the task itself is specified.

```
phrase_dict = {
  "GuV": "a 'Gewinn- und Verlustrechnung' (profit and loss statement) table",
  "Aktiva": "a 'Aktiva' (assets) table",
  "Passiva": "a 'Passiva' (liabilities) table",
  'other': "a text that does not suit the categories of interest",
}
```

6.2.4.2 Information extraction

Figure 6.4 shows the confusion matrix for the information extraction task with Llama 4 Maverick and five examples. It shows, that the LLM starts to predict numeric values for every row instead of prediction *null* if a row is missing. Figure 6.5 shows, what kind of numeric values are predicted. We find two peaks for predicting floating point numbers close to zero or close to 30, while the true values (and values from the examples provided) are in a range of 1_000 and 10_000_000. Thus, we assume the values are hallucinated and not wrongly picked from the examples provided.

Local and global context window / attention (Khowaja, 2025). Trained on 256 k tokens with FP8 precision.

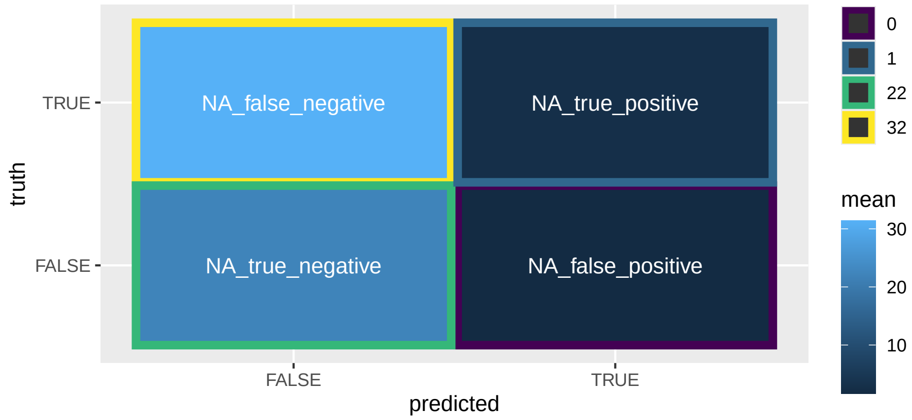


Figure 6.4: Showing the confusion matrix for the information extraction task with Llama 4 Maverick and five in-context learning examples.

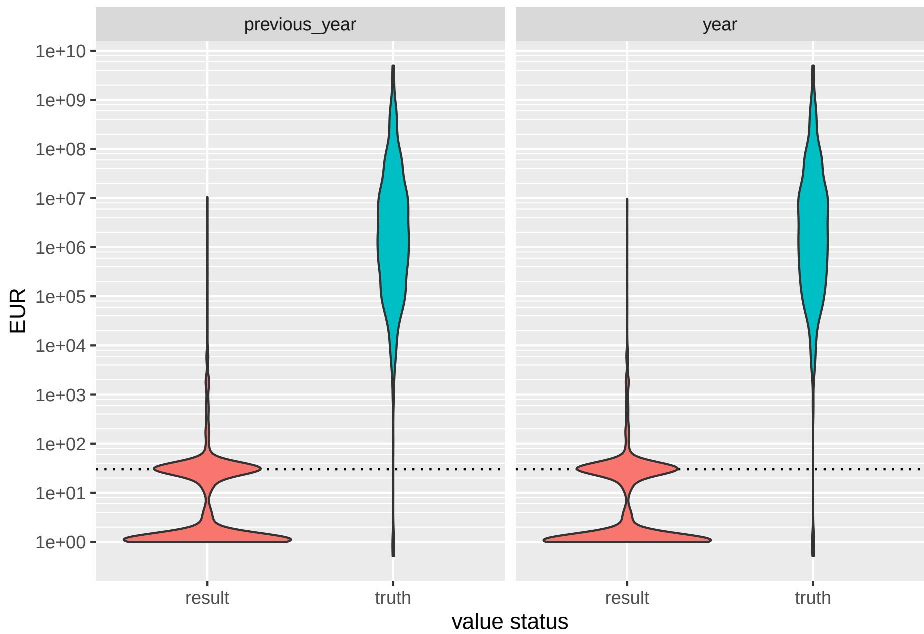


Figure 6.5: Comparing the predicted numeric values with the true value distribution for the information extraction task with Llama 4 Maverick and five in-context learning examples. The dotted line is marking the value 30 EUR.

6.3 Practical Implications and Limitations

This section discusses the results from an engineers perspective, focusing on implications for the planned HITL application and possible improvements.

6.3.1 Page identification

A detailed view on the results shows, a combination of two LLMs would be necessary, to get the best results for each target class. A more general approach would be using Llama 4 Scout for multi-class classification for all target classes. If there is little VRAM Ministral-8B also does a decent job in multi-class classification.

We recommend, to refine the page range, using a term frequency based approach, for efficiency reasons discussed in Section 6.3.1.5. It is an important prerequisite, that any method used for page refinement has a recall of 1.0. Otherwise, a user potentially has to inspect the whole document and no improvement is reached compared to the manual processing.

Afterwards the LLM can be used to perform a multi-class classification on the top k pages from the ranking resulting from the term frequency approach. The page, that gets the highest confidence score from the LLM should be used for the information extraction task. The confidence score ranking should be kept for presenting alternative pages, if the chosen one is incorrect.

Figure 6.6 visualizes our recommendation, to not include an obligatory step, to confirm the selected page by a human user, but start the information extraction right away. When the user is checking the results, a wrong page will be noticed immediately. Alternative pages can be inspected manually, following the order in the confidence score ranking.

Already classified pages should be stored in a vector database together with their class label. Thus, they can be used for future classification tasks and improve the system's performance. The examples for the in-context learning few-shot strategy should be chosen based on the vector similarity and include documents from the same company. We recommend to fill the vector database document by document in the beginning, before starting with batch wise processing.

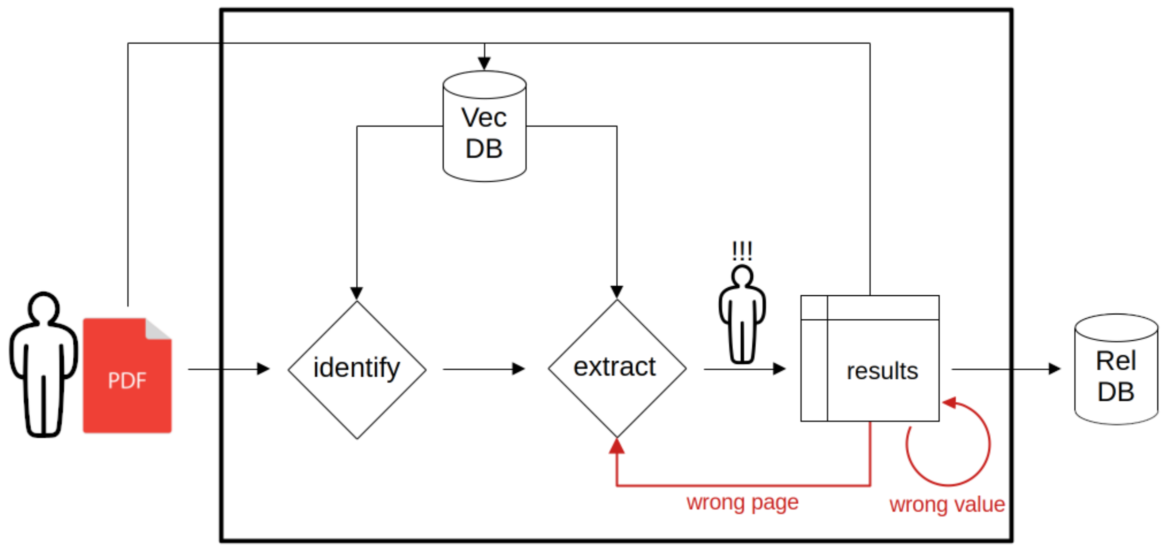


Figure 6.6: Showing the information extraction process in a HITL application. We propose to include user action only after the information extraction. If a wrong page is selected, this can be fixed and extraction runs again. Wrong extracted values and handling unknown row identifiers should be done in one place.

6.3.1.1 Transfering the system to new problems

Introducing new areas of application should be easily possible and manageable even from a regular user. For the term frequency approach we can set up a pipeline, where the user has to enter a list of keywords and then gets presented a page ranking, based on TF-IDF values. The user might adjust the key word list or select correct pages right away, to build a ground truth. If there are more measures of interest (e.g. a float frequency as well) the system can automatically train a random forest classifier as well.

Another approach is, that the user provides documents and a list on which page the information of interest is located. This can be the base for a retrieval augmented few-shot classifier, that will improve in the process of classifying more pages.

6.3.1.2 Limitations of transfer

The term frequency and LLM classification might perform worse, if the information of interest, is just making up a small part of the pages content. If the information is placed in a table, we can use a visual table detection model, to identify all tables. We show in the Appendix E.3.2.2, that this is a promising approach. Then we can use the retrieval augmented few-shot approach to identify, which table is the correct one.

If the information is not even in a table, but part of a regular sentence, it might get difficult to find the correct page with this approach. Maybe the TOC approach could be used for page range refinement in this case, if the information is found in a section with known heading.

6.3.1.3 Possible improvement

Term frequency approach Based on the error analysis for the term frequency approach in Section 6.2.1.2, we recommend, instead of using the normalized sum of term frequencies, sum of boolean indicators or the sum of BM25 scores one could also use each single indicator as a feature for random forest. But for this case it would be necessary, to provide a larger train and test dataset to prevent overfitting, because the number of features would be large.

Table of contents understanding approach H. Li, Gao, et al. (2023) use a few-shot learning strategy with the TOC approach. We implemented only a zero-shot strategy. With more expertise and concrete examples in the prompt, this approach probably could perform better as reported here. Also they used OpenAI's GPT-4 for the TOC understanding task, which has much more parameters than Ministral-8B. One could investigate, if Qwen3-235B would perform much better with the text based TOC.

6.3.1.4 Conclusion

gleiches Level?

6.3.1.5 Energy usage and runtime

The fastest and least energy consuming strategy, using only LLMs, is to use a small model as Ministral-8B-Instruct for the multi-class approach. This is more effective than running three binary classifications.

An alternative approach could be to binary predict if the page is of any target type and then perform a classification, which type exactly the page is of. But this would probably consume as much energy as the multi-class approach, because we have to provide a balanced amount of examples for each class. The results of the multi-class strategy are good enough to run it right away.

In both strategies the k required for perfect recall is three, using the Ministral-8B-Instruct model[2].

Nevertheless, it is more promising, to reduce the number of pages, to classify with the LLM in the first place. This can be achieved, by running the term-frequency approach first to refine the page range, and then use the LLM approach.

[2] Potentially smaller fine tuned models can solve the task even more efficient.

Compare with manual page identification The manual approach is the slowest. We identified the pages of interest for all target classes in ten random documents for the benchmark. We used the TOC and the search function to find key words like **Aktiva** or **Bilanz**. Anyhow, it is almost as fast as the full multi-classification using Llama 4 Scout, while consuming eight times less energy. Comparing it to Ministral-8B-Instruct it takes three times longer but consumes less then half of the energy.

Thus, the only arguments for a LLM classification without previous page refinement are, that the human user could perform another task, while the LLM is classifying a whole batch of documents and that it frees the user from a boring task. With a view on the energy usage driven climate change process we would discard both arguments.

Not taken into account fo this comparison are factors as:

- costs to buy and maintain hardware (i.e. a GPU cluster).
- higher costs per runtime if the LLM compute is purchased from cloud providers. The number of response tokens per page can be limited to one. In contrast here are the counts of input tokens needed to classify a single page:
  - four classes (3 random examples): 11 k input tokens
  - binary (3 random examples): 6.5 k input tokens
- payment and insurance to pay for a human (e.g. student coworker).
- the training time and energy consumption for training either
  - a LLM (probably done by the LLM provider).
  - a human (growing up, getting educated).
- the energy consumed to produce the hardware.

Nevertheless, the argument that probably will be most important to many CEOs are costs. The costs presented in table 5.3 only include the costs for energy.

For a human employee we have to add their payment and insurance costs. Even for a student worker this will sum up to 0.5 CENTS per second. This totals in 319.80 € for identifying the target pages for 1_000 documents. If we assume, one uses GPT-4.1-mini hosted in the Azure cloud instead of running the \ac rt{LLM} locally, we estimate a price of 118.80 €.

6.3.2 Information extraction

We have not investigated yet, if the 2 % wrong extracted numeric values are caused by not respecting currency units, or if there is another reason. A potential reason may be numeric values, that get stitched together by faulty text extraction. If this would be the case, the question would arise, if more effort should be invested into more sophisticated table extraction methods. Personally we would recommend investing the energy into establishing end-to-end data pipelines, to ensure that document parsing is unnecessary at all.

Nevetheless, we checked other extraction libraries for the text extraction, too. Our benchmark includes advanced approaches as *Azure Document Intelligence* and *Docling*, that can detect and extract tables. We also tried to provide Markdown instead of plain text, generated by *Docling* and *pymupdf* to maintain the tabula structure information. We also tested an OCR approach using *tesseract*.

Table 6.3 shows the results aggregated over all documents for the best prompting method for Qwen3-235B for each input type. We can find no improvement over the results we achieve with the text extracted by *pdfium*. In contrast, we find that the performance with Markdown generated by *pymupdf* or the text generated using OCR are worse. But this is not a Markdown specific problem. Qwen3-235B performs equally well with the Markdown generated by *Docling* as with the plain text extract.

Table 6.3: Comparing the best prompting method for different types of input for the information extraction task with Qwen3235B.

model	method	extractor	input_format	mean_total
Qwen3-235B-A22B-Instruct-2507-FP8	top_3_rag_examples	docling	markdown	0.958
Qwen3-235B-A22B-Instruct-2507-FP8	top_5_rag_examples	docling	text	0.967
Qwen3-235B-A22B-Instruct-2507-FP8	top_5_rag_examples	pdfium	text	0.970
Qwen3-235B-A22B-Instruct-2507-FP8	5_random_examples	pymupdf	markdown	0.871
Qwen3-235B-A22B-Instruct-2507-FP8	top_5_rag_examples	pymupdf	text	0.970
Qwen3-235B-A22B-Instruct-2507-FP8	top_5_rag_examples	tesseract	text	0.848

Limitations The current approach uses a single strict schema. This enables easy evaluation of the results and processing in down stream tasks. But it excluded reports of well known companies as BVG and BSR. For the final system we intend to add an additional classifier between the page identification and information extraction, that detects what granularity the asset table is reported in. Then a proper schema is applied in the information extraction task.

But this will still not use all information found in all tables. Therefore, we intend to access, which row identifiers did not match and extract the corresponding rows in a separate json formatted list. We have to test, if this can be performed in a single step or with another information extraction prompt.

Another approach would be to use guided instead of restricted decoding. For this strategy one can pass the description of the target structure in the prompt and just enforce the generation of valid json code. We unwillingly made first tests with this approach, because we were not able to use a strict schema with the models of OpenAI. We describe our findings about this in Section 6.2.2.6.

Conclusion To summarize, we recommend to invest resources into an end-to-end supply of machine-readable information and good user experience instead of system that achieves perfect extraction results. We believe that the performance as it is is already sufficient. Especially if the document extraction database is build document by document to use a RAG architecture to take advantage of in-context learning with same-company examples.

6.3.3 Error rate guidance

A measure to guide a users attention, which predictions to inspect and which to trust as they are would be important for the information extraction task. Checking the extracted values for a single table takes up to three minutes. This totals in 300 minutes prediction checking for our sample. This is up to four times faster, than manually cpoy over all the values.

Selecting a smaller model that is finishing after 2:30 minutes instead of 3:00 minutes is not speeding up the process a lot, as long as the error checking has to be performed on all values afterwards. This would be the case, if we get perfect results in the first place, or if a criteria is found, the predictions can be grouped by forming bins with near zero error rate.

An interesting criteria for this goal could be the company which creates a document. We show in 6.2.2.2 that the information extraction already yields error free results for many companies. If the characteristics of the table of interest do not change, we can expect to find the same empirical error rate for future documents, too. But we already identified some companies in our sample, where the structure changed in important details, e.g. starting to report values in *T€*.

Furthermore, we recommend to choose a threshold value close to zero instead of exact zero, to define groups of predictions with a sufficient low empirical error rate. Otherwise those groups are too unstable.

One could also test to use more sophisticated confidence measures. But we are not to optimistic to find a much better discriminating criteria this way.

Conclusion The discrimination between correct and incorrect classified pages works well. Even though it will be rarely needed there, because of the almost perfect results, it can be used to form a ranking of pages, that can be used to find the most promising alternative, if the classification should be wrong.

For the information extraction task additional segmentation might help to find a use for the confidence score there as well. But alternative segmentation criteria might remove the need for the confidence scores as additional criteria at all.

6.4 Not covered

In this section we briefly mention the topics we have not investigated in this thesis.

Table extraction and document parsing There are classical libraries and recent machine learning models, that are specialised on extracting tables from documents. We tested *tabula* as a classical solution, but were not satisfied by its results. Without borders in the tables it rarely identified rows and columns correct.

We did not manage to implement recent models for this task, for example visual LLMs. There are promising results for their performance on table extraction, but a easy to use version for *transformers* or *vLLM* is missing most of the time. We were surprised and glad to see, that our approach works well with the basic text extract and did not continue to pursue the attempt to use more sophisticate extraction tolls or models.

Optical character recognition If documents have no textual information, because they are just a collection of (scanned) images, OCR is a necessary step in data preparation, because all our approaches use the text extract. In the OCR process the textual information of the image is converted into machine-readable text. OCR systems often perform a document layout analysis too, in order to handle multi-column layouts and tables.

Fine-tuning LLMs can be fine tuned on specific tasks. For example, they can be trained to produce text of a specific style or on classification tasks. For classification tasks a softmax pooling layer is added. Another example is the object detection model Yolo 12 we are using. It is fine tuned on detecting tables.

We do not perform fine tuning with LLMs. Instead we are using in-context learning and test, if this yields sufficient results. It would be interesting to estimate, how many in-context prompts have to be queried, so that the energy consumption for the additional token processing is becoming greater than the energy consumption of a fine tuning.

Training an encoder-only model Instead of fine tuning a LLM it probably would be more efficient to train a smaller (encoder-only) model, e.g. BERT (Bidirectional Encoder Representations from Transformers). Such a model would probably be pretty efficient for the whole classification task as well. But in contrast to an LLM we would have to retrain the model for every new classification task and build compose a training dataset for this.

UX design study We have not performed a UX design study so far. This might be performed in near future to create an application, using the information extraction processes investigated in this thesis, that can be used well by the employees of RHvB. Participating potential users early in the process, is meaningful for successful software development (*User Participation in Software Development Projects – Communications of the ACM*, 2010) and can prevent developing unnecessary features or non intuitive, cumbersome processes. At the same time it can increase the willingness and motivation to use the final AI driven product (Errida & Lotfi, 2021).

Advanced prompting techniques Advanced prompting strategies as Chain-of-Thought, ReAct or Self-Consitency (*Prompt Engineering Guide*, n.d.), prompting to think step-by-step or put in a lot of effort. But since we already find pretty good performance and also first cases of content rot, we do not recommend to invest more effort in advanced prompt template building.

6.5 Outlook

In order to build the outlined system at RHvB we have to overcome some infrastructural hurdles. Since the local infrastructure gets discarded, we need to move into a cloud environment, ensure secure connections with the Berlin intranet and secured processing of confidential data. The short term strategy does not include any GPU resources yet.

Among the possible improvements of the system we see increasing the flexibility and extending the number of use-cases as most important. This includes, finding a solution, how unknown row identifiers can be handled by the users and setting up an automated process to fill the RAG component with examples, that can be used for the identification and extraction tasks.

Overall, we still have to build a user interface, that guides the process and allows users to correct correct errors and handle unknown row identifiers. Furthermore, we have to discuss how to proceed with new entries. Should we aggregate them with known row identifiers or extend the list of known entities in the main database? Maybe introducing a new hierarchy level, to handle the finer grained reporting of housing cooperative for some of the enties?

Additionally we want to set up a machine learning operations pipeline that performs health check for our system and can be used for additional benchmarks. Thus we could test new models' performance - and more important - check if the new examples provided by the employees might be harmful for the systems performance.

Additionally, we will request the annual reports in XBRL (eXtensible Bbusiness Reporting Language) format. German companies are obligated to create reports in this format since 2012 for reporting to the tax authorities (§ 5b EStG). A fact not used at RHvB so far. We hope to build an error free solution based on this machine-readable format for this specific task.

6

6.6 Ethical & Practical Considerations

6.6.1 PDF extraction limitations

Pdfminer informs that the text of some annual reports from *IBB* and *Berlinovo* should not be extracted. This information is given in a meta data field of the PDF. We use the text extract from these documents for our study anyway.

Errors catched by HITL approach before they have down stream implications.

6.6.2 Computational constraints

The extraction with LLMs is computationally demanding and should be run on GPUs. To run Qwen3-235B-A32B-Instruct-FP8 - the model that yields the best results for the extraction task - four H200 GPUs are needed. To run Llama 4 Scout - the model that performed best in the page identification task - another four H200 GPUs are needed. It make no sense, to load these models ad-hoc, since the setup time is around 30 minutes. For running Ministral-8B and Qwen3-8B a single Nvidia 5090 RTX or A100 are needed.

6.6.3 Generalizability scope

The approach tested here is probably using on other companies annual reports as well. To extract information that is only filling a small part of a page the framework may has to be adjusted. The page identification could be trickier with some approaches if only a single key word is searched.

6.6.4 Ethical considerations

The extraction of numeric information is not the same as making decisions. It probably isn't affected by any bias, that is discriminating humans.

The automatisation of information extraction is potentially replacing low requirements work places. At RHvB there are no jobs for such a task anymore. More free time for other tasks. Shifting to more complex tasks.

AI Act does probably not apply, since decisions are not made on individual level?: Are there restrictions on the use of automated decision-making? Yes, **individuals should not be subject to a decision that is based solely on automated processing** (such as algorithms) and that is legally binding or which significantly affects them.

Chapter 7

Conclusion

This thesis investigated the use of large language models (LLMs) and related approaches for extracting financial information from German annual reports, with a focus on supporting the audit processes at the RHvB. We addressed three main research questions: how to effectively locate specific information in financial reports using LLMs, how to extract this information accurately, and whether additional information from the extraction process can guide users in verifying results.

Our results demonstrate that LLMs can be used effectively to identify relevant pages and extract structured financial data, achieving performance close to or exceeding human benchmarks in many cases. Combining LLMs with term-frequency-based methods further improves efficiency, reducing computational costs without sacrificing performance. The experiments also show that open-weight models are viable alternatives to proprietary solutions, especially when data privacy is a concern.

We found that LLMs are capable of extracting multiple numeric values from complex tables, matching entities, and handling missing or ambiguous data accurately. The use of confidence scores for guiding users in human-in-the-loop (HITL) workflows, highlighting which values require verification and which can be trusted, is found to be less successful. Other extraction metadata might help to achieve this goal.

Our findings have practical implications for automating information retrieval in regulatory and financial contexts, particularly where documents are unstructured and machine readability is limited. The developed framework and evaluation strategies can be adapted to similar tasks in other domains.

Future work could explore further improvements in system architectures, prompt engineering, and integration with advanced document layout analysis. Expanding the scope to more heterogeneous document types and new tasks, as well as refining HITL strategies, will help ensure robust and scalable solutions for real-world applications.

References

Adhikari, N. S., & Agarwal, S. (2024). *A Comparative Study of PDF Parsing Tools Across Diverse Document Categories* (arXiv:2410.09871). arXiv. https://doi.org/10.48550/arXiv.2410.09871

Amgen Scholars Program. (n.d.). *How to Interpret Violin Charts.* https://www.labxchange.org/library/items/lb:LabXchange:4

Auer, C., Lysak, M., Nassar, A., Dolfi, M., Livathinos, N., Vagenas, P., Ramis, C. B., Omenetti, M., Lindlbauer, F., Dinkla, K., Mishra, L., Kim, Y., Gupta, S., Lima, R. T. de, Weber, V., Morin, L., Meijer, I., Kuropiatnyk, V., & Staar, P. W. J. (2024). *Docling Technical Report* (arXiv:2408.09869). arXiv. https://doi.org/10.48550/arXiv.2408.09869

Bentley, S. V. (2025). Knowing you know nothing in the age of generative AI. *Humanities and Social Sciences Communications*, *12*(1), 409. https://doi.org/10.1057/s41599-025-04731-0

Berkov, M. (2025). Understanding LLM Logprobs. In *Thinking Sand*.

Blecher, L., Cucurull, G., Scialom, T., & Stojnic, R. (2023). *Nougat: Neural Optical Understanding for Academic Documents* (arXiv:2308.13418). arXiv. https://doi.org/10.48550/arXiv.2308.13418

Boseak, C. (2025). Evaluating Log-Likelihood for Confidence Estimation in LLM-Based Multiple-Choice Question Answering. *Innovative Journal of Applied Science*, *02*(04). https://doi.org/10.70844/ijas.2025.2.29

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). *Language Models are Few-Shot Learners* (arXiv:2005.14165). arXiv. https://doi.org/10.48550/arXiv.2005.14165

Burnwal, Y., & Jaiswal, Dr. R. C. (2023). A Comprehensive Survey on Prediction Models and the Impact of XGBoost. *International Journal for Research in Applied Science and Engineering Technology*, *11*(12), 1552–1556. https://doi.org/10.22214/ijraset.2023.57625

Cai, W., Jiang, J., Wang, F., Tang, J., Kim, S., & Huang, J. (2025). A Survey on Mixture of Experts in Large Language Models. *IEEE Transactions on Knowledge and Data Engineering*, 1–20. https://doi.org/10.1109/TKDE.2025.3554028

Carvalho, F., & Guedes, G. P. (2020). *TF-IDFC-RF: A Novel Supervised Term Weighting Scheme* (arXiv:2003.07193). arXiv. https://doi.org/10.48550/arXiv.2003.07193

Chamberlain, P. (2020). Knowledge is not everything. *Design for Health*, *4*(1), 1–3. https://doi.org/10.1080/24735132.2020.1731203

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Collis, J., & Hussey, R. (2014). *Business research: A practical guide for undergraduate & postgraduate students* (1. Publ.; 4. ed). Palgrave Macmillan.

Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Chang, B., Sun, X., Li, L., & Sui, Z. (2024). A Survey on In-context Learning. In Y. Al-Onaizan, M. Bansal, & Y.-N. Chen (Eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (pp. 1107–1128). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.emnlp-main.64

El-Haj, M., Alves, P., Rayson, P., Walker, M., & Young, S. (2020). Retrieving, classifying and analysing narrative commentary in unstructured (glossy) annual reports published as PDF files. *Accounting and Business Research*, *50*(1), 6–34. https://doi.org/10.1080/00014788.2019.1609346

Errida, A., & Lotfi, B. (2021). The determinants of organizational change management success: Literature review and case study. *International Journal of Engineering Business Management*, *13*, 18479790211016273. https://doi.org/10.1177/18479790211016273

Francis, Sofi. A., & Sangeetha, M. (2025). A comparison study on optical character recognition models in mathematical equations and in any language. *Results in Control and Optimization*, *18*, 100532. https://doi.org/10.1016/j.rico.2025.100532

Google. (n.d.). Gemma 3n model overview. In *Google AI for Developers*. https://ai.google.dev/gemma/docs/gemma-3n.

Gougherty, A. V., & Clipp, H. L. (2024). Testing the reliability of an AI-based large language model to extract ecological information from the scientific literature. *Npj Biodiversity*, *3*(1), 13. https://doi.org/10.1038/s44185-024-00043-9

Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: An Overview* (arXiv:2008.05756). arXiv. https://doi.org/10.48550/arXiv.2008.05756

Grootendorst, M. (2024). *A Visual Guide to Mixture of Experts (MoE)*. https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts.

Haddouchi, M., & Berrado, A. (2024). *A survey and taxonomy of methods interpreting random forest models* (arXiv:2407.12759). arXiv. https://doi.org/10.48550/arXiv.2407.12759

HGB. (2025). *Handelsgesetzbuch in der im Bundesgesetzblatt Teil III, Gliederungsnummer 4100-1, veröffentlichten bereinigten Fassung, das zuletzt durch Artikel 1 des Gesetzes vom 28. Februar 2025 (BGBl. 2025 I Nr. 69) geändert worden ist*.

Hintze, J. L., & Nelson, R. D. (1998). Violin Plots: A Box Plot-Density Trace Synergism. *The American Statistician*, *52*(2), 181–184. https://doi.org/10.1080/00031305.1998.10480559

Hong, Z., Ward, L., Chard, K., Blaiszik, B., & Foster, I. (2021). Challenges and Advances in Information Extraction from Scientific Literature: A Review. *JOM*, *73*, 1–18. https://doi.org/10.1007/s11837-021-04902-9

Hu, L., & Wang, K. (2023). *Computing SHAP Efficiently Using Model Structure Information* (arXiv:2309.02417). arXiv. https://doi.org/10.48550/arXiv.2309.02417

Huang, X., & Marques-Silva, J. (2024). On the failings of Shapley values for explainability. *International Journal of Approximate Reasoning*, *171*, 109112. https://doi.org/10.1016/j.ijar.2023.109112

IBM Global Technology Services. (2006). *The toxic terabyte*.

Jhguch. (2025). Box plot. *Wikipedia*.

Jr, J., Moffitt, K., & Byrnes, P. (2015). How Big Data Will Change Accounting. *Accounting Horizons*, *29*, 150227130540002. https://doi.org/10.2308/acch-51069

Kang, Z., Zhao, X., & Song, D. (2025). *Scalable Best-of-N Selection for Large Language Models via Self-Certainty* (arXiv:2502.18581). arXiv. https://doi.org/10.48550/arXiv.2502.18581

Kauf, C., Chersoni, E., Lenci, A., Fedorenko, E., & Ivanova, A. A. (2024). *Log Probabilities Are a Reliable Estimate of Semantic Plausibility in Base and Instruction-Tuned Language Models* (arXiv:2403.14859). arXiv. https://doi.org/10.48550/arXiv.2403.14859

Kelly Hong, & Anton Troynikov. (2025). *Context Rot: How Increasing Input Tokens Impacts LLM Performance*. https://research.trychroma.com/context-rot.

Khowaja, S. A. (2025). Analysis of Llama 4's 10 Million Token Context Window Claim. In *Medium*.

Kong, K., Zhang, J., Shen, Z., Srinivasan, B., Lei, C., Faloutsos, C., Rangwala, H., & Karypis, G. (2024). *OpenTab: Advancing Large Language Models as Open-domain Table Reasoners* (arXiv:2402.14361). arXiv. https://doi.org/10.48550/arXiv.2402.14361

Krzywinski, M., & Altman, N. (2014). Visualizing samples with box plots. *Nature Methods*, *11*(2), 119–120. https://doi.org/10.1038/nmeth.2813

Kuka", "Valeriia. (n.d.). *Shot-Based Prompting: Zero-Shot, One-Shot, and Few-Shot Prompting*. https://learnprompting.org/docs/basics/few_shot.

Kulkarni, V. Y., & Sinha, P. K. (2012). Pruning of Random Forest classifiers: A survey and future directions. *2012 International Conference on Data Science & Engineering (ICDSE)*, 64–68. https://doi.org/10.1109/ICDSE.2012.6282329

Kulkarni, V., & Sinha, P. (2013). Random forest classifiers: A survey and future research directions. *International Journal of Advanced Computing*, *36*, 1144–1153.

Levy, M., Jacoby, A., & Goldberg, Y. (2024). *Same Task, More Tokens: The Impact of Input Length on the Reasoning Performance of Large Language Models* (arXiv:2402.14848). arXiv. https://doi.org/10.48550/arXiv.2402.14848

Li, H., Gao, H. (Harry)., Wu, C., & Vasarhelyi, M. A. (2023). *Extracting Financial Data from Unstructured Sources: Leveraging Large Language Models* ({{SSRN Scholarly Paper}} 4567607). Social Science Research Network. https://doi.org/10.2139/ssrn.4567607

Li, H., Wei, D. (Kathy)., Moffitt, K., & Vasarhelyi, M. A. (2023). *Addressing the Last Mile Problem in Open Government Data: Using AIS Technologies to Enhance Governmental Financial Reporting* ({{SSRN Scholarly

8

Paper}} 4385883). Social Science Research Network. https://doi.org/10.2139/ssrn.4385883

Li, W.-D., & Ellis, K. (2024). *Is Programming by Example solved by LLMs?* (arXiv:2406.08316). arXiv. https://doi.org/10.48550/arXiv.2406.08316

Li, Z., Bouazizi, M., Ohtsuki, T., Ishii, M., & Nakahara, E. (2024). Toward Building Trust in Machine Learning Models: Quantifying the Explainability by SHAP and References to Human Strategy. *IEEE Access*, *12*, 11010–11023. https://doi.org/10.1109/ACCESS.2023.3347796

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2023). *Lost in the Middle: How Language Models Use Long Contexts* (arXiv:2307.03172). arXiv. https://doi.org/10.48550/arXiv.2307.03172

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2019). *Explainable AI for Trees: From Local Explanations to Global Understanding*. arXiv. https://doi.org/10.48550/ARXIV.1905.04610

Lundberg, S., & Lee, S.-I. (2017). *A Unified Approach to Interpreting Model Predictions* (arXiv:1705.07874). arXiv. https://doi.org/10.48550/arXiv.1705.07874

Ma, H., Chen, J., Zhou, J. T., Wang, G., & Zhang, C. (2025). *Estimating LLM Uncertainty with Evidence* (arXiv:2502.00290). arXiv. https://doi.org/10.48550/arXiv.2502.00290

Mahmudi, A. (22 Dec 2024 6:46pm). Natural Language Processing. In *School of Computing and Information Systems*. https://cis.unimelb.edu.au/research/artificial-intelligence/research/Natural-Language-Processing.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. In *Cambridge Aspire website*. https://www.cambridge.org/highereducation/books/introduction-to-information-retrieval/669D108D20F556C5C30957D63B5AB65C; Cambridge University Press. https://doi.org/10.1017/CBO9780511809071

Mienye, I. D., & Jere, N. (2024). A Survey of Decision Trees: Concepts, Algorithms, and Applications. *IEEE Access*, *12*, 86716–86727. https://doi.org/10.1109/ACCESS.2024.3416838

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2025). *Large Language Models: A Survey* (arXiv:2402.06196). arXiv. https://doi.org/10.48550/arXiv.2402.06196

Molnar, C. (2025). *Interpretable machine learning: A guide for making black box models explainable* (Third edition). Christoph Molnar.

Mosqueira-Rey, E., Hernández-Pereira, E., Alonso-Ríos, D., Bobes-Bascarán, J., & Fernández-Leal, Á. (2023). Human-in-the-loop machine learning: A state of the art. *Artificial Intelligence Review*, *56*(4), 3005–3054. https://doi.org/10.1007/s10462-022-10246-w

Nannini, L., Alonso-Moral, J. M., Catalá, A., Lama, M., & Barro, S. (2024). Operationalizing Explainable Artificial Intelligence in the European Union Regulatory Ecosystem. *IEEE Intelligent Systems*, *39*(4), 37–48. https://doi.org/10.1109/MIS.2024.3383155

Natarajan, S., Mathur, S., Sidheekh, S., Stammer, W., & Kersting, K. (2024). *Human-in-the-loop or AI-in-the-loop? Automate or Collaborate?* (arXiv:2412.14232). arXiv. https://doi.org/10.48550/arXiv.2412.14232

O'Sullivan, C. (2023). *The mathematics behind Shapley Values*.

Paul, R. (2025). *State-of-the-Art Model Architectures for Document Layout Analysis*. https://www.rohan-paul.com/p/state-of-the-art-model-architectures.

Prompt Engineering Guide. (n.d.). https://www.promptingguide.ai/techniques.

Qin, L., Chen, Q., Feng, X., Wu, Y., Zhang, Y., Li, Y., Li, M., Che, W., & Yu, P. S. (2025). *Large Language Models Meet NLP: A Survey* (arXiv:2405.12819). arXiv. https://doi.org/10.48550/arXiv.2405.12819

Qwen Team. (2025). Qwen3: Think Deeper, Act Faster. In *Qwen*. https://qwenlm.github.io/blog/qwen3/.

Qwen/Qwen3-4B · Hugging Face. (2025). https://huggingface.co/Qwen/Qwen3-4B.

Raj, M., & Mishra, N. (2025). *Exploring new Approaches for Information Retrieval through Natural Language Processing* (arXiv:2505.02199). arXiv. https://doi.org/10.48550/arXiv.2505.02199

Raschka, S. (2025). *Instruction Pretraining LLMs*. https://magazine.sebastianraschka.com/p/instruction-pretraining-llms.

Rathi, R. N., & Mustafi, A. (2023). The importance of Term Weighting in semantic understanding of text: A review of techniques. *Multimedia Tools and Applications*, *82*(7), 9761–9783. https://doi.org/10.1007/s11042-022-12538-3

Raymaekers, J., Rousseeuw, P. J., Verdonck, T., & Yao, R. (2024). Fast Linear Model Trees by PILOT. *Machine Learning*, *113*(9), 6561–6610. https://doi.org/10.1007/s10994-024-06590-3

Rivera-Lopez, R., Canul-Reich, J., Mezura-Montes, E., & Cruz-Chávez, M. A. (2022). Induction of decision trees as classification models through metaheuristics. *Swarm and Evolutionary Computation*, *69*, 101006. https://doi.org/10.1016/j.swevo.2021.101006

Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for IDF. *Jour-*

nal of Documentation, *60*(5), 503–520. https://doi.org/10.1108/00220410410560582

Robertson, S. E., & Walker, S. (1994). Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In B. W. Croft & C. J. Van Rijsbergen (Eds.), *SIGIR '94* (pp. 232–241). Springer London. https://doi.org/10.1007/978-1-4471-2099-5_24

Robertson, S., & Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, *3*, 333–389. https://doi.org/10.1561/1500000019

Saito, T., & Rehmsmeier, M. (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, *10*(3), e0118432. https://doi.org/10.1371/journal.pone.0118432

Shapley, L. S. (2016). 17. A Value for n-Person Games. In H. W. Kuhn & A. W. Tucker (Eds.), *Contributions to the Theory of Games, Volume II* (pp. 307–318). Princeton University Press.

Smock, B., Pesala, R., & Abraham, R. (2022). PubTables-1M: Towards comprehensive table extraction from unstructured documents. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4624–4632. https://doi.org/10.1109/CVPR52688.2022.00459

Tahir. (2025). 🧠Understanding LLM Context Windows: Tokens, Attention, and Challenges. In *Medium*.

User Participation in Software Development Projects – Communications of the ACM. (2010).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need* (arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762

Wickham, H., & Stryjewski, L. (2011). *40 years of boxplots*.

Willard, B. T., & Louf, R. (2023). *Efficient Guided Generation for Large Language Models* (arXiv:2307.09702). arXiv. https://doi.org/10.48550/arXiv.2307.09702

Wohlin, C., & Aurum, A. (2015). Towards a decision-making structure for selecting a research design in empirical software engineering. *Empirical Software Engineering*, *20*(6), 1427–1455. https://doi.org/10.1007/s10664-014-9319-7

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2024). *Experimentation in Software Engineering*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-69306-3

Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., & He, L. (2022). A Survey of Human-in-the-loop for Machine Learning. *Future Generation Computer Systems*, *135*, 364–381. https://doi.org/10.1016/j.future.2022.05.014

Yang, J. (2022). *Fast TreeSHAP: Accelerating SHAP Value Computation for Trees* (arXiv:2109.09847). arXiv. https://doi.org/10.48550/arXiv.2109.09847

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). *Dive into Deep Learning* (arXiv:2106.11342). arXiv. https://doi.org/10.48550/arXiv.2106.11342

Zhang, D., Song, J., Bi, Z., Yuan, Y., Wang, T., Yeong, J., & Hao, J. (2025). *Mixture of Experts in Large Language Models* (arXiv:2507.11181). arXiv. https://doi.org/10.48550/arXiv.2507.11181

Zhang, Q., Huang, V. S.-J., Wang, B., Zhang, J., Wang, Z., Liang, H., Wang, S., Lin, M., Zhang, W., & He, C. (2024). *Document Parsing Unveiled: Techniques, Challenges, and Prospects for Structured Information Extraction* (arXiv:2410.21169). arXiv. https://doi.org/10.48550/arXiv.2410.21169

Zhao, T. Z., Wallace, E., Feng, S., Klein, D., & Singh, S. (2021). *Calibrate Before Use: Improving Few-Shot Performance of Language Models* (arXiv:2102.09690). arXiv. https://doi.org/10.48550/arXiv.2102.09690

Zhong, X., Tang, J., & Yepes, A. J. (2019). *PubLayNet: Largest dataset ever for document layout analysis* (arXiv:1908.07836). arXiv. https://doi.org/10.48550/arXiv.1908.07836

Zhu, Y., Yuan, H., Wang, S., Liu, J., Liu, W., Deng, C., Chen, H., Dou, Z., & Wen, J.-R. (2024). *Large Language Models for Information Retrieval: A Survey* (arXiv:2308.07107). arXiv. https://doi.org/10.48550/arXiv.2308.07107

List of Figures

8

8

8

8

List of Tables

8

Glossary

PDF Portable Document Format
LLM large language model
HITL human-in-the-loop
AI artificial intelligence
RHvB Rechnungshof von Berlin
GPT generative pre-trained transformers
regex regular expression
TOC table of contents
NLP natural language processing
MoE mixture of experts
RAG retrieval augmanted generation
SHAP SHapley Additive exPlanations
TF-IDF Frequency-Inverse Document Frequency
BM25 best matching 25
IDF Inverse Document Frequency
OCR optical character recognition
FFN feed forward network
PLE Per-Layer Embedding
XGBoost Extreme Gradient Boosting
XAI explainable artificial intelligence
UX user experience
AUC area under the curve
json JavaScript Object Notation
GPU graphics processing unit
BHT Berliner Hochschule für Technik
vLLM Virtual Large Language Model
HTML hyper text markup language
TDP thermal design power
ebnf extended Backus–Naur form
ACFR Annual Comprehensive Financial Report
ESG environmental, social, and governance
BERT Bidirectional Encoder Representations from Transformers
XBRL eXtensible Bbusiness Reporting Language
MCC multi-class classification
UI user interface

8

Chapter A

Appendix A - Page identification report

The first research question asks, how LLMs can be used, to effectively locate specific information in a financial report. The task for this thesis is to identify the pages where the balance sheet (*Bilanz*) and the profit-and-loss-and-statement (*Gewinn- und Verlustrechnung, GuV*) are located. The balance sheet is composed of two tables showing the assets (*Aktiva*) and liabilities (*Passiva*) of a company. Often, these two tables are on separate pages. Hereafter, the German terms **Aktiva**, **Passiva** and **GuV** will be used.

A.1 Baseline: Regex

The first approach presented in this section is, to use a key word list and regex (regular expression) to filter out irrelevant pages. It is setting the performance baseline for the following approaches. Building a sound regular expression often is an iterative process. In a first approach a very *simple regex* was implemented. To increase the recall to 1.0 the regular expression was extended[1]. This second regex is called *exhaustive regex*. In a third attempt minor changes have been made to the *exhaustive regex* to increase the precision without decreasing the recall. This regular expression is called *exhausitve regex restricted*. The regular expressions can be found in the appendix (see section E.5).

Table A.1 shows the mean performance for precision, recall and F1 for the three regular expressions for the three types of pages to identify[2]. It was possible to create a regular expression that has a high recall for all target types. The precision is low for all tested regular expressions and target types. Figure A.1 gives insight into performance differences between the companies. There is only one document from *Berlin Energie und Netzholding* where the **GuV** is not identified except with the *exhausitve regex restricted*[3].

The regular expressions have been tested on the texts extracted with multiple Python libraries. The reported standard deviations are very small. This means that there are no substantial differences in the extracted texts on a word level[4]. But table E.1 in section E.3.1 shows that there are differences in the extraction speed.

Code can be found at "benchmark_jobs/page_identification/page_identification_benchmark_regex.ipynb"

Summary Nothing works well

A.2 Table of Contents understanding

The second approach presented in this section leverages the TOC understanding capabilities of LLMs. H. Li, Gao, et al. (2023) use this approach with long documents as a first step to determine a page range of

[1] The idea is that the regular expression approach is computationally cheap. If we can rely on the fact, that it keeps all relevant pages we can use additional, computationally more expensive approaches to further refine the page range.

[2] See Figure G.2 for a graphical representation.

[3] I don't understand why the restricted version is finding the page but the non-restricted regex is not.

[4] Since the results are not depending on the text extraction library, the *exhaustive regex restricted* ran only with the text extracted by the fastest extraction library: *pdfium*. This library is used for the most tasks in this thesis. Later faced issues with the text extracted by *pdfium* are discussed in @ref().

A

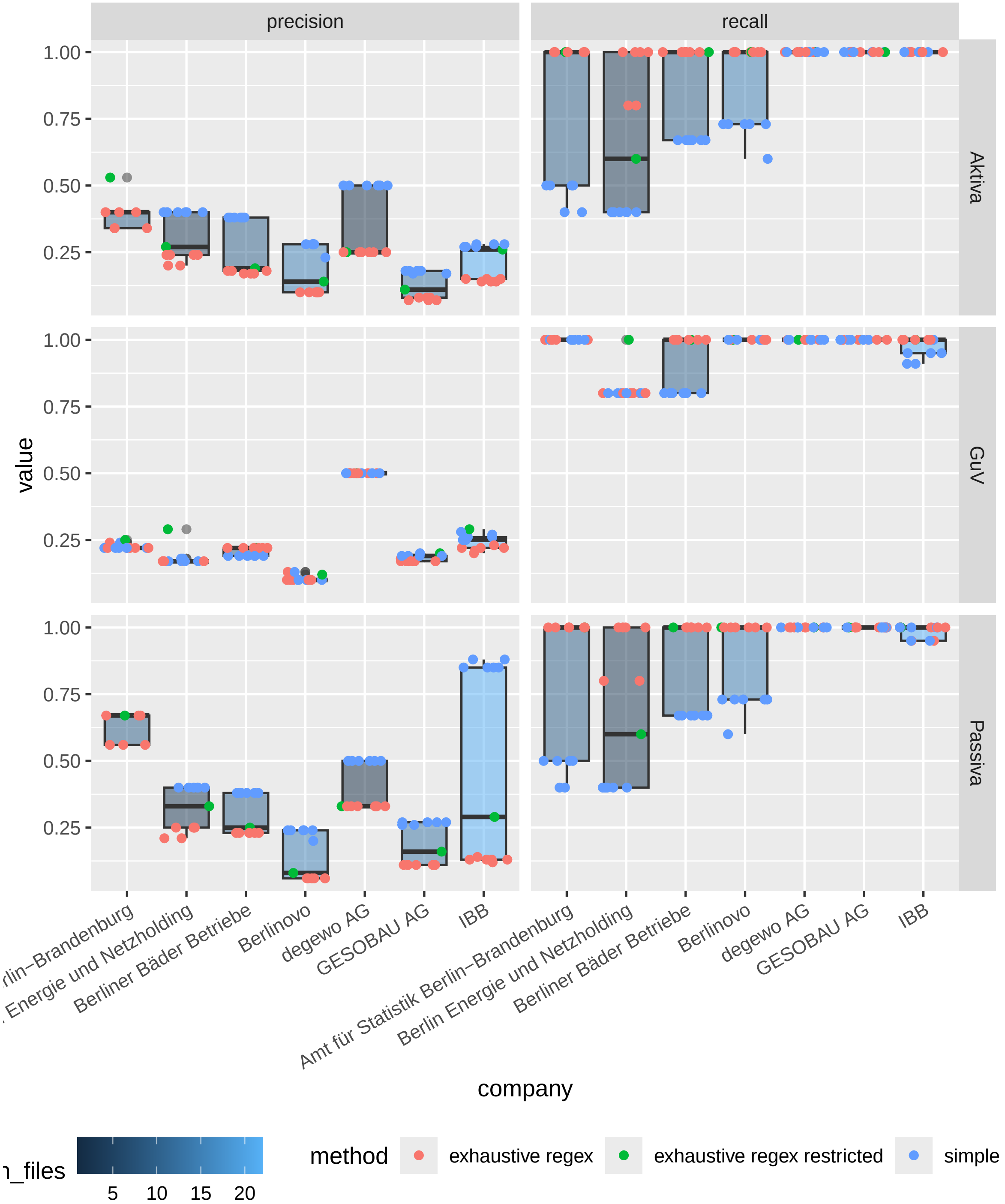


Figure A.1: Comparing the performance among different companies.

Table A.1: Comparing page identification metrics for different regular expressions for each classification task by type of the target table.

method	type	precision	recall	F1
Aktiva				
simple regex	Aktiva	**0.273** ± 0.005	0.788 ± 0.010	**0.403** ± 0.005
exhaustive regex restricted	Aktiva	0.190	0.990	0.320
exhaustive regex	Aktiva	0.132 ± 0.004	**0.997** ± 0.005	0.233 ± 0.008
Passiva				
simple regex	Passiva	**0.400** ± 0.009	0.780 ± 0.009	**0.530** ± 0.009
exhaustive regex restricted	Passiva	0.190	0.980	0.320
exhaustive regex	Passiva	0.130 ± 0.000	**0.993** ± 0.010	0.230 ± 0.000
GuV				
simple regex	GuV	0.180 ± 0.006	0.938 ± 0.008	0.302 ± 0.010
exhaustive regex restricted	GuV	**0.210**	**1.000**	**0.350**
exhaustive regex	GuV	0.173 ± 0.008	**1.000** ± 0.000	0.295 ± 0.012

interest. If the predicted page range is correct and narrow, this approach is more efficient than processing the whole document with a LLM directly. The TOC in a PDF document can be embedded in a standardized, machine readable format or be presented in varying, human readable forms of text on any page. Of course there are documents without any TOC.

Thus, the task is investigated based on two different input data formats In one case the LLM is provided with text extracted from the beginning of the document. In the other case the LLM is provided with the Markdown formatted version of the machine readable TOC embedded in the document. Subsection A.2.1 shows the results for the text based approach. Subsection A.2.1 shows the results for the approach, using the embedded TOC.

Additionally, each approach is performed three times with minor changes in the prompt. The prompts used for both approaches can be found at E.4.1. The prompt was adjusted two times to tackle shortcomings in the results. The first change adds the information, that assets and liabilities are part of the balance sheet. It is the balance sheet, that is listed in the TOC - not the assets or liabilities itself. The second change specifies the information, that assets and liabilities are often on separated pages, into, liabilities often are found on the page after the assets.

The code can be found in:

- “benchmark_jobs/page_identification/toc_extraction_mistral.ipynb”
- “benchmark_jobs/page_identification/toc_extraction_qwen.ipynb”

A.2.1 Details for the approaches

Text based H. Li, Gao, et al. (2023) used the TOC to identify the pages of interest. In their approach the table of contents is extracted from the text. Based on their observation, that the TOC in ACFRs is found within the initial 165 lines of the converted document (H. Li, Gao, et al., 2023, p. 20), they use the first 200 lines of text.

My initial expectation was to find the TOC within the first five pages. Often there are way less than 200 lines of text on the five first pages (see Figure A.2). In my approach the first step is to prompt the LLM to identify and extract the TOC in a given text extractˆ[The prompt can be found in section E.4.1]. For the same documents Ministral 2410 8B findsˆ[The strings extracted in this step have not been checked in detail.]

- 63 strings that should represent a table of contents among the first five pages.
- 68 strings that should represent a table of contents among the first 200 lines.

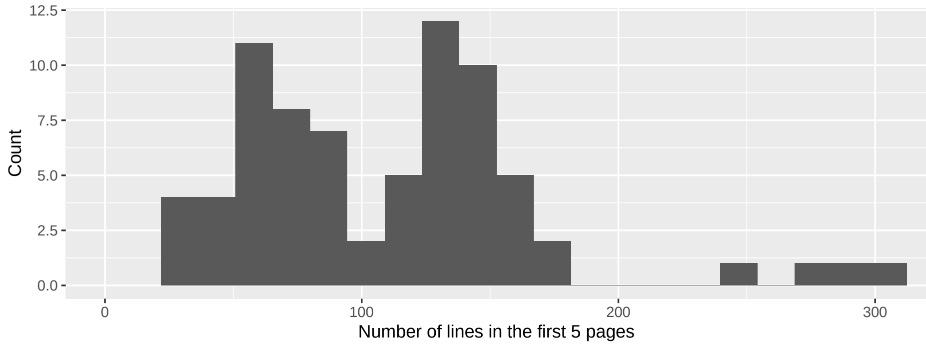


Figure A.2: Histogram of the number of lines in the first 5 pages of the annual reports

Machine readable TOC based I also tested to use the TOC representation embedded within the PDF files. First, this limits the text amount to process. Second, this hopefully increases the quality of the data passed to the LLM. 43 of the 80 annual reports have a machine readable embedded TOC. The embedded TOC is converted into markdown format before it gets passed to the LLM. Here is an example:

```
## |   hierarchy_level | title                         |   page_number |   enumeration |
## |------------------:|:------------------------------|--------------:|--------------:|
## |                 1 | Lagebericht                   |             5 |             1 |
## |                 1 | Bilanz                        |             7 |             2 |
## |                 1 | Gewinn- und Verlustrechnung   |            10 |             3 |
## |                 1 | Anhang                        |            13 |             4 |
## |                 1 | Lagebericht                   |            17 |             5 |
## |                 1 | Bilanz                        |            25 |             6 |
## |                 1 | Anhang                        |            31 |             7 |
## |                 1 | Anlagenspiegel                |            39 |             8 |
## |                 1 | Bestätigungsvermerk           |            42 |             9 |
```

A.2.2 Results

A.2.2.1 Comparison of the different approaches

base prompt Table A.2 shows that the machine readable TOC approach has the highest rate of correct page ranges for all types with the base prompt. It also predicts the most correct page ranges in absolute numbers for **Aktiva** and **GuV**. Thus, it also has the highest rate of correct page ranges based on the total number of page ranges to identify over all documents - no matter, if there was a TOC of any type in the document or not - for **Aktiva** and **GuV** of around 27 %.

Table A.2: Comparing the number and percentage of correct identified page ranges among the approaches.

benchmark_type	type	n_correct	n	n_total	perc_correct	perc_correct_total
200 lines	Aktiva	9.0	63	82	14.3	11.0
200 lines	GuV	22.0	95	102	23.2	21.6
200 lines	Passiva	**6.0**	62	81	9.7	**7.4**
5 pages	Aktiva	7.0	58	82	12.1	8.5
5 pages	GuV	15.0	89	102	16.9	14.7
5 pages	Passiva	3.0	57	81	5.3	3.7
machine readable	Aktiva	**22.0**	35	82	**62.9**	**26.8**
machine readable	GuV	**28.0**	56	102	**50.0**	**27.5**
machine readable	Passiva	4.0	34	81	**11.8**	4.9

Figure A.3 shows that the amount of correct predicted page ranges for **Passiva** is lowest for all approaches but can be improved by simply extending the predicted end page number by one the most. This improvement would be best for the machine readable TOC approach. This approach is the only one, where the number of correct page ranges **Aktiva** would not increase if we extend its range by one. Table A.3 shows that this is the case, because the machine readable TOC approach predicts the same end page for **Passiva** as for **Aktiva** in 84.8 % of the cases, even though the prompt for all approaches included the information, that **Aktiva** and **Passiva** are on separate pages.

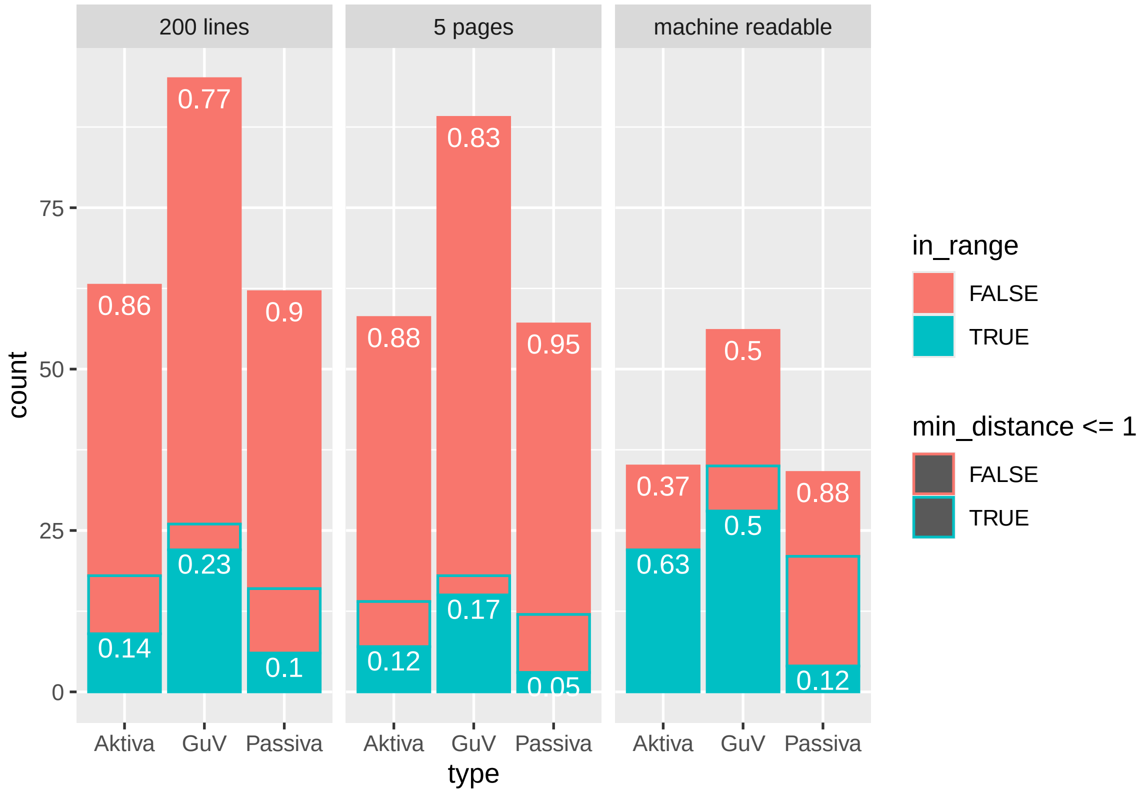


Figure A.3: Comparing number of found TOC and amount of correct and incorrect predicted page ranges

advanved prompts As a first attempt, to increase the correct page range rate for **Passiva** I tried to specify, that assets and liabilities are part of the balance sheet. This did work for the text based approaches, but not for the machine readable approach (see Figure G.3). Figure A.4 shows that it is more successful, to explicit tell the LLM that the liabilities table is often on the page, after the assets table.

Table A.4 shows the results from the final zero shot prompt. The machine readable TOC approach is now predicting best for all types. Nevertheless, a correct page range prediction rate below 60, 45, 50 % is still unsufficient to build downstream task on withou human checkups. Table A.5 shows, that the machine readable TOC approach is the fastest as well.

Table A.6 shows, that this advantage of the machine readable TOC approach is not coming from wide predicted page ranges. It has the smallest median range size among all approaches. Figure A.5 shows, that

Table A.3: Comparing the number and percentage end pages prediction for Aktiva and Passiva that are equal.

benchmark_type	equal_end_page	n	perc_equal_end_page
200 lines	20	58	34.5
5 pages	26	53	49.1
machine readable	28	33	84.8

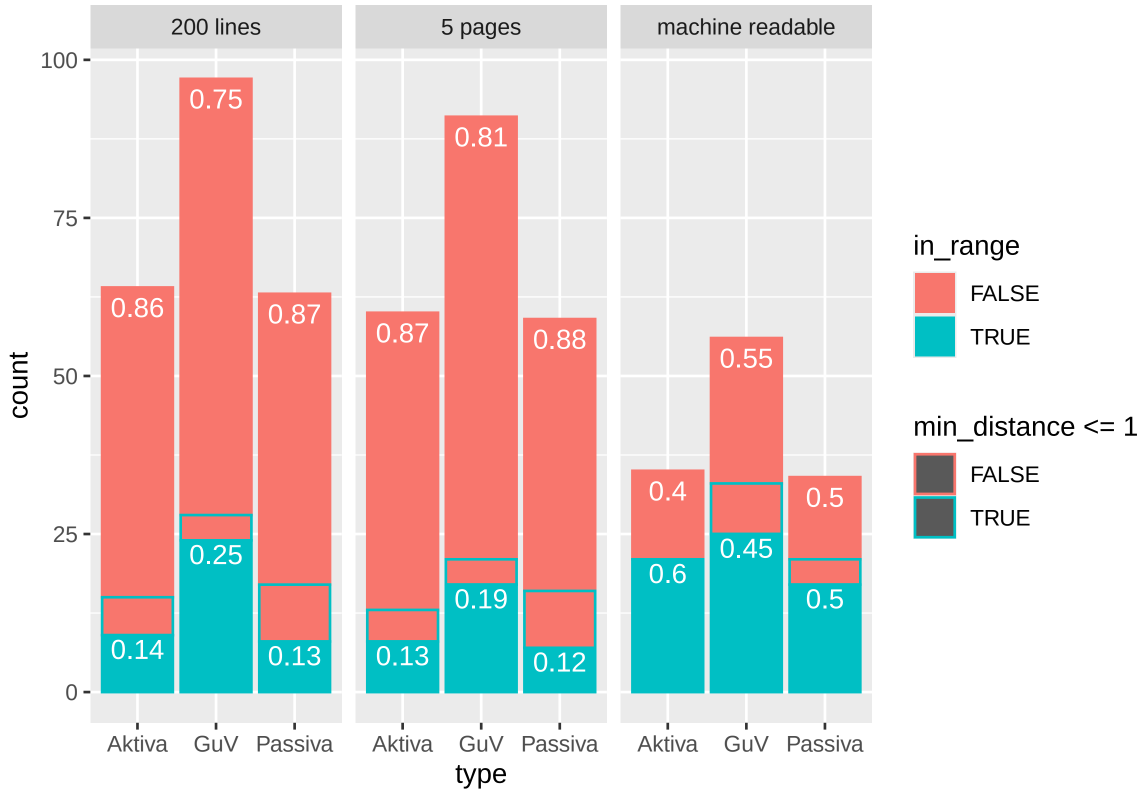


Figure A.4: Comparing number of fount TOC and amount of correct and incorrect predicted page ranges

A

Table A.4: Comparing the number and percentage of correct identified page ranges among the approaches.

benchmark_type	type	n_correct	n	n_total	perc_correct	perc_correct_total
200 lines	Aktiva	9.0	64	82	14.1	11.0
200 lines	GuV	24.0	97	102	24.7	23.5
200 lines	Passiva	8.0	63	81	12.7	9.9
5 pages	Aktiva	8.0	60	82	13.3	9.8
5 pages	GuV	17.0	91	102	18.7	16.7
5 pages	Passiva	7.0	59	81	11.9	8.6
machine readable	Aktiva	**21.0**	35	82	**60.0**	**25.6**
machine readable	GuV	**25.0**	56	102	**44.6**	**24.5**
machine readable	Passiva	**17.0**	34	81	**50.0**	**21.0**

Table A.5: Comparing GPU time for page range prediction and table of contents extraction. Time in seconds per text processed.

Benchmark Type	Page range predicting	TOC extracting
200 lines	0.57	3.8
5 pages	**0.56**	**2.19**
machine readable	0.63	NA

especially the ranges for **GuV** are not normally distributed. Some far off lying range sizes are shifting the mean off from the median.

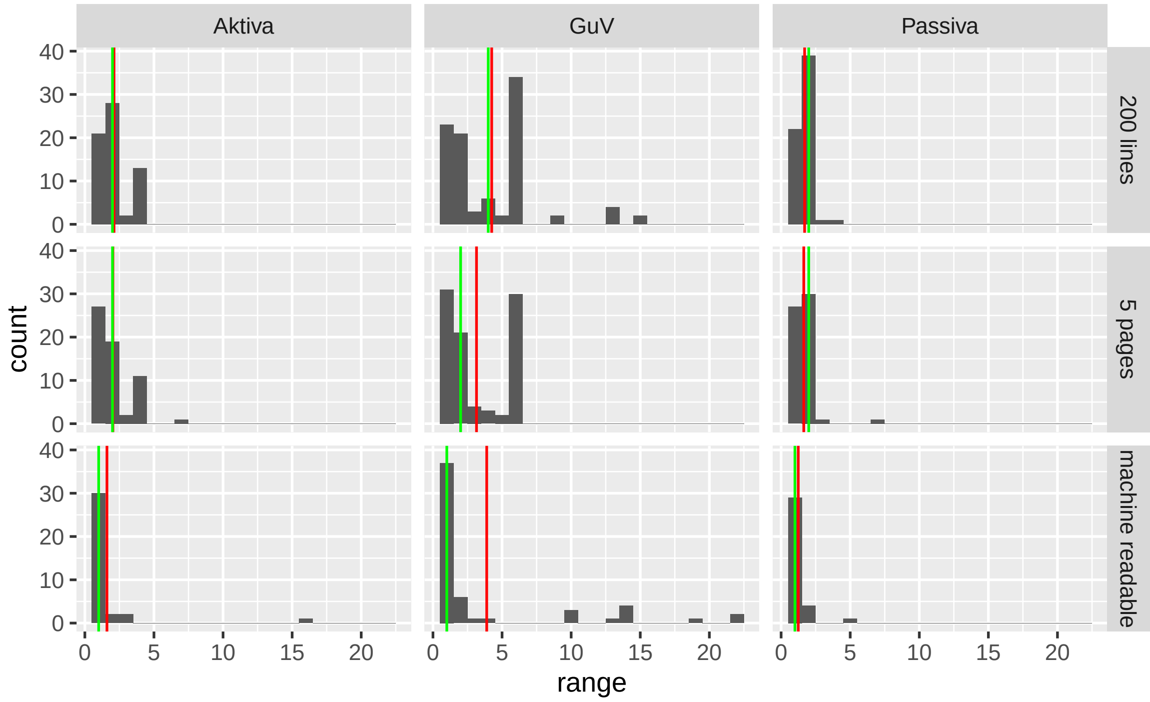


Figure A.5: Comparint the predicted page range sizes. The red vertical line shows the mean and the green one shows the median of these sizes.

Figure A.6 shows that the confidence of the LLMs responses is higher for the machine readable TOC approach as well. Besides a single group that was predicted far off, the page ranges are closer to the correct pages too. A linear regression of the correlation between minimal page distance and logistic probability shows that is has a similar slope for all approaches and target types.

A

A.2.3 Machine readable TOC approach specific results

Figure A.7 shows, that correct predictions for the page range are more probable when the embedded TOC has a medium number of entries. It is possible to drop documents with less than 9 without loosing a single correct prediction. This means that the LLM was not able to make a correct prediction for documents with TOC, that have less then 9 entries. This is not surprising since neither **Bilanz** nor **GuV** are mentioned there explicit.

Table A.6: Comparing the mean and median page range sizes.

benchmark_type	type	mean_range	SD_range	median_range	MAD_range
200 lines	Aktiva	2.11	1.09	2	1.48
200 lines	GuV	4.25	3.29	4	2.97
200 lines	Passiva	1.7	0.59	2	0
5 pages	Aktiva	2.03	1.29	2	1.48
5 pages	GuV	3.15	2.17	2	1.48
5 pages	Passiva	1.64	0.89	2	0
machine readable	Aktiva	1.6	2.56	**1**	0
machine readable	GuV	3.89	5.75	**1**	0
machine readable	Passiva	1.24	0.74	**1**	0

Figure A.6: Showing the minimal distance of the predicted page range to the actual page number overthe logprobs of the models response confidence.

It has no big influence on the predictions, if the TOC is passed formatted as markdown or json. With the json formatted TOC it found two more correct page ranges[5]. This was tested because the relation between heading and value for the column *page_number* might have been clearer[6] in json for a one-dimensional working LLM.

A

A.3 Classification with LLMs

The third approach we present in this section, uses pretrained LLMs to classify, if a given text extract is including any of the target tables. Two classification approaches are presented.

On the one hand, a binary classification is used three times, to predict, if the text extract is including an **Aktiva**, **Passiva** or **GuV** table, once at a time. In this case the LLM is forced to answer with either *yes* or *no*. On the other hand, the LLM performed a \acr{mcc}. For the MCC (multi-class classification) the LLM is forced to answer *Aktiva*, *Passiva*, *GuV* or *other*. The prompts can be found in appendix in section E.4.2.

The different classification tasks are combined with different prompting strategies. A zero shot approach is setting the baseline. In a second approach the excerpt of the relevant law is provided with the context. Additionally, three few shot approaches are used.

In the few shot approaches text examples and a correct classification for the text examples are provided. Figure A.8 shows how many examples the LLM gets provided, depending on the classification type and chosen parameter *n_example*[7]. For both approaches three example selection strategies are implemented. First, random examples for each page type get sampled from the truth dataset. Second, a vector database provides the entries that are closest to the target text for each page type. Third, the vector database just provides the texts that are closest to the target text without considering the page type of the examples returned.

For the binary classification task the LLM is provided with more examples for the target type than for other types. Thus, the number of examples and tokens is reduced. This should reduce the runtime as well. On

[5]This result is based on a single test run.
[6]With json the key *page_number* gets repeated every line, while it is just mentiones once in the beginning of the markdown formatted tables.
[7]See also Table F.1.

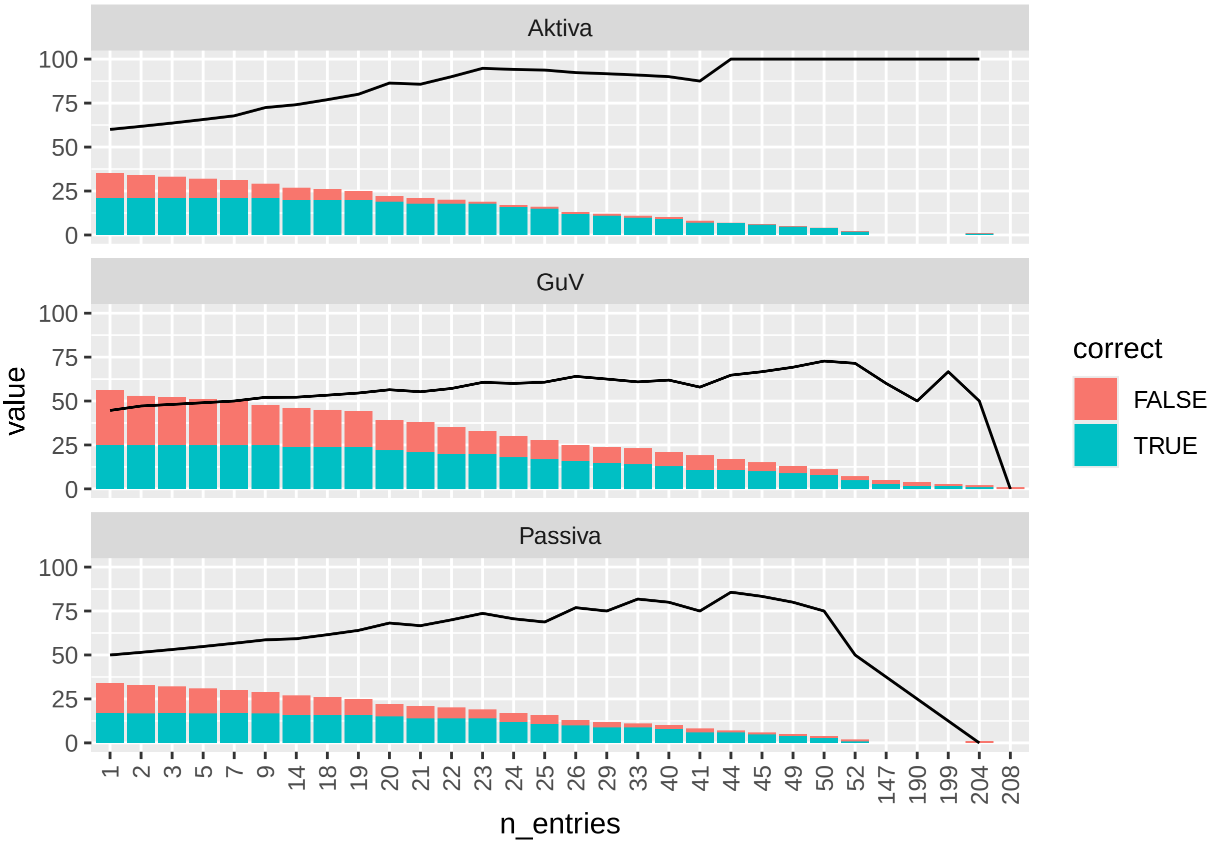


Figure A.7: Showing the amount of correct and incorrect predicted page ranges (bars) and the percentage of correct predictions (black line).

A

the same time the LLM should get enough information about the structure and contents of the target class and some information how it differs from other big tables or general text pages.

For the MCC the same amount of every possible class is provided. Thus, the relation between the parameter *n_examples* and the number of tokens to process is stronger for the *n_random_examples* and *n_rag_examples* strategies.

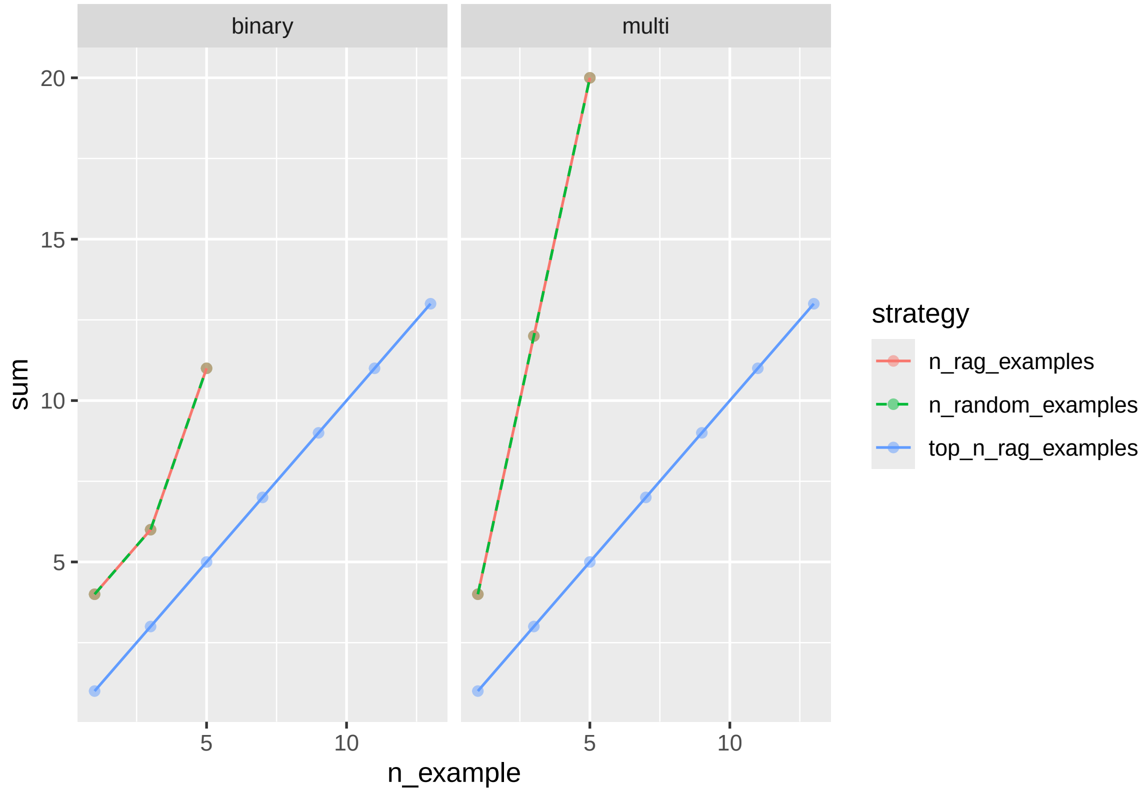


Figure A.8: Comparing the actual number of provided examples depending on the classification type, example selection strategy and chosen parameter n-examples. The slope for the top-n-rag-examples strategy is the same for both approaches. The line for the strategies n-random-examples and n-rag-examples is equal within each approach.

Table A.7 shows, which models have been used in the classification benchmarks. Overall 25 models from 8 model families have been tested. Prerequirement for a model to be tested is, that it can be used with the vLLM library, accessed via hugging face and fits into the combined VRAM of 8 nvidia B200 graphic cards (1.536 TB). The models cover a wide range of (active) parameter sizes. Especially for the Qwen family many models of different parameter sizes are used in the benchmark, to investigate if there is a clear minimum amount of parameters needed, to solve the classification task.

The results of the benchmarks have been logged as json files totaling in 2.1 GB of data for the final results.

To do:

- compare out of company vs in company rag

A.3.1 Binary classification

Table A.8 shows the best performing combination of model family and prompting method for each classification target type. The classification of **GuV** tables works best and is solved almost perfectly. The F1 score for **Aktiva** and **Passiva** are 0.07 lower for the top performing model. The median F1 score of **GuV** is 0.69 0.1 higher than the median F1 score for **Aktiva** (0.815) and 0.2 higher than the median F1 score for **Passiva** (0.78).

Table A.7: Overview of benchmarked LLMs for the classification tasks.

model_family	model	parameter_count
Falcon3-10B-Instruct	Falcon3-10B-Instruct	10
gemma-3-12b-it	gemma-3-12b-it	12
gemma-3-27b-it	gemma-3-27b-it	27
gemma-3-4b-it	gemma-3-4b-it	4
gemma-3n-E4B-it	gemma-3n-E4B-it	4
Llama-3	Llama-3.1-8B-Instruct	8
Llama-3	Llama-3.1-70B-Instruct	70
Llama-3	Llama-3.3-70B-Instruct	70
Llama-4	Llama-4-Maverick-17B-128E-Instruct-FP8	17
Llama-4	Llama-4-Scout-17B-16E-Instruct	17
Ministral-8B-Instruct-2410	Ministral-8B-Instruct-2410	8
Mistral-Large-Instruct-2411	Mistral-Large-Instruct-2411	124
Mistral-Small-3.1-24B-Instruct-2503	Mistral-Small-3.1-24B-Instruct-2503	24
phi-4	phi-4	15
Qwen 2.5	Qwen2.5-0.5B-Instruct	0.5
Qwen 2.5	Qwen2.5-1.5B-Instruct	1.5
Qwen 2.5	Qwen2.5-3B-Instruct	3
Qwen 2.5	Qwen2.5-7B-Instruct	7
Qwen 2.5	Qwen2.5-14B-Instruct	14
Qwen 2.5	Qwen2.5-32B-Instruct	32
Qwen 2.5	Qwen2.5-72B-Instruct	72
Qwen 3	Qwen3-8B	8
Qwen 3	Qwen3-30B-A3B-Instruct-2507	30
Qwen 3	Qwen3-32B	32
Qwen 3	Qwen3-235B-A22B-Instruct-2507	235

Ministral 8B Instruct 2410 is performing best for the binary classification task for each target type. Llama-4-Scout-17B-16E-Instruct is performing second best for **Aktiva** and **GuV** and is close to the second best for **Passiva** as well. The runtime of Ministral 8B Instruct 2410 is four times lower than the runtime of Llama-4-Scout-17B-16E-Instruct. In addition, the time to load Llama-4-Scout-17B-16E-Instruct into the VRAM is much longer[8], because it has a total of 109B parameters. It was surprising that Googles gemma models perform so bad[9].

Figure A.9 shows, the classification performance for Ministral 8B 2410 in detail. It shows the F1 score for each target type over the models runtime. It shows the results for the different prompting strategies (*method_families*) with differently colored shapes. The *zero_shot* strategy performs worst with a F1 score below 0.6. Next come the *law_context* and *top_n_rag_examples* strategy. Above those the *n_random_examples* and finally the *n_rag_examples* strategy perform best.

The shape is giving information, if the example proveded to the LLM are selected from other companies than the target table comes from only, or if they can also be selected from documents of the same company. This is only relevant for strategies that get the examples picked by the documents vector embedding distances. The LLM performs better[10], if examples from documents of the same company can be used. If this is not permitted, the *n_random_example* approach performs better than the *n_rag_example* for the classification of **GuV** and **Passiva** tables.

The number inside of the shapes is referring to the *n_examples* function parameter. Most models got benchmarked with an *n_examples* value of up to three. The actual number of examples provided to the models is depending on the method family / example selection strategy and can be looked up in Table F.1.

The best performing model, Ministral 8B 2410, was provided with mode examples to investigate the effect of a richer context. The predictions do not get better by providing more and more examples. Figure A.9 shows, that the improvements get smaller naturally going from three to five examples while approaching an F1 score of 1.0.

But for the *n_rag_example* strategy we find a significant drop in the F1 score, if we set the *n_examples* to five[11] and examples pages come from annual reports of other companies. This is caused by a sever recall drop. For the *n_random_example* strategy we see a small drop with the F1 score for the class **Passiva** as well. Taking into account that the runtime also almost is twice as high, this is very inefficient.

Figure A.9 also shows, that the results are stable[12]. Running the benchmark three times shows similar results in the F1 score for each strategy. This is reflected by closely overlapping shapes of the same color with the same number within.

Figure A.10 shows the experiments for Ministral-8B-Instruct-2410 with *n_examples* greater or equal three. This time the actual number of examples provided to the LLM are shown in the shapes to increase the comparability among the different strategies. Additionally, it shows results for the *top_n_rag_example* strategy with *n_examples* up to 13. The F1 score of the *top_n_rag_example* strategy stays lower than the F1 score of the *n_rag_examples* strategy, even though there are more examples used. This is mainly caused by lower precision scores, probably because there are no contrasting examples provided.

Figure G.4 and Figure G.5 shows the F1 performance over normalized rundtime for all benchmarked models. Comparing Ministral-8B-Instruct-2410 with Mistral-124B-Instruct-2411 shows that one can spend over tenfold amount of computation power without getting better results.

It also shows, that with Qwen 2.5 it needs at least the 3B parameter model to achieve good results. Comparing the 32B and 72B parameter models shows, that th performance does not increase anymore, but starts to decrease. For Qwen 3 it shows, that only the newer mix of experts models give reasonable results.

The mix of expert models show good performance for the Llama 4 family as well and reduce the compute time compared with the 72B models of LLama 3. But for LLama 4 Maverick the performance drops using the *n_rag_examples* strategy with three *n_examples*. The performance of Llama 3.1 70B was higher than the performance of Llama 3.3 70B.

[8] It takes around 30 minutes to setup a vllm instance with Llama-4 Scout compared to 4:30 minutes setuptime for Ministral 8B 2410.

[9] This is not due to a temporary technical problems caused by a bug in the transformers version shipped with the vllm 0-9-2 image. Those problems have been overcome. The performance stays bad.

[10] It has a better F1 score, when examples from the same company are permitted. The recall is better with examples from same company. The precision is better without. The improvement in the recall is stronger.

[11] In this case five examples for the target table type and two examples for each other type are provided, totaling at twelve examples.

[12] Earlier experiments on a subset of the pages have been run five times indicating stable results. Running the experiments up to tree times in this very task indicate this as well.

Table A.8: Overview of benchmarked LLMs for the binary classification tasks. Limiting the number of examples provided for the few shot approach to 3.

model_family	model	classification_type	method_family	n_examples	f1_score
mistralai	Ministral-8B-Instruct-2410	GuV	n_rag_examples	3	0.99
Llama-4	Llama-4-Scout-17B-16E-Instruct	GuV	n_rag_examples	3	0.98
Qwen 2.5	Qwen2.5-32B-Instruct	GuV	n_rag_examples	1	0.93
mistralai	Ministral-8B-Instruct-2410	Passiva	n_rag_examples	3	0.92
mistralai	Ministral-8B-Instruct-2410	Aktiva	n_rag_examples	3	0.92
Llama-4	Llama-4-Scout-17B-16E-Instruct	Passiva	n_rag_examples	3	0.86
Qwen 2.5	Qwen2.5-32B-Instruct	Aktiva	n_rag_examples	1	0.85
Qwen 3	Qwen3-235B-A22B-Instruct-2507	Aktiva	n_rag_examples	3	0.85
Llama-4	Llama-4-Scout-17B-16E-Instruct	Aktiva	n_rag_examples	1	0.84
Llama-4	Llama-4-Scout-17B-16E-Instruct	Aktiva	n_rag_examples	3	0.84
Qwen 2.5	Qwen2.5-32B-Instruct	Passiva	n_rag_examples	1	0.81
Qwen 3	Qwen3-235B-A22B-Instruct-2507	Passiva	n_rag_examples	3	0.79
Llama-3	Llama-3.1-70B-Instruct	Aktiva	n_rag_examples	1	0.79
Llama-3	Llama-3.1-70B-Instruct	Aktiva	n_rag_examples	3	[redacted]
Llama-3	Llama-3.1-70B-Instruct	Passiva	n_rag_examples	1	0.76
microsoft	phi-4	Aktiva	law_context	1	[redacted]
Llama-3	Llama-3.1-70B-Instruct	GuV	law_context	1	0.69
Qwen 3	Qwen3-30B-A3B-Instruct-2507	GuV	n_rag_examples	3	0.68
microsoft	phi-4	Passiva	law_context	1	0.66
google	gemma-3-27b-it	Passiva	n_rag_examples	1	0.58
google	gemma-3-27b-it	Aktiva	n_rag_examples	1	0.54
google	gemma-3-27b-it	GuV	n_rag_examples	1	0.52
tiiuae	Falcon3-10B-Instruct	Passiva	n_random_examples	1	0.5
tiiuae	Falcon3-10B-Instruct	Aktiva	n_rag_examples	1	0.45
tiiuae	Falcon3-10B-Instruct	GuV	top_n_rag_examples	1	0.34

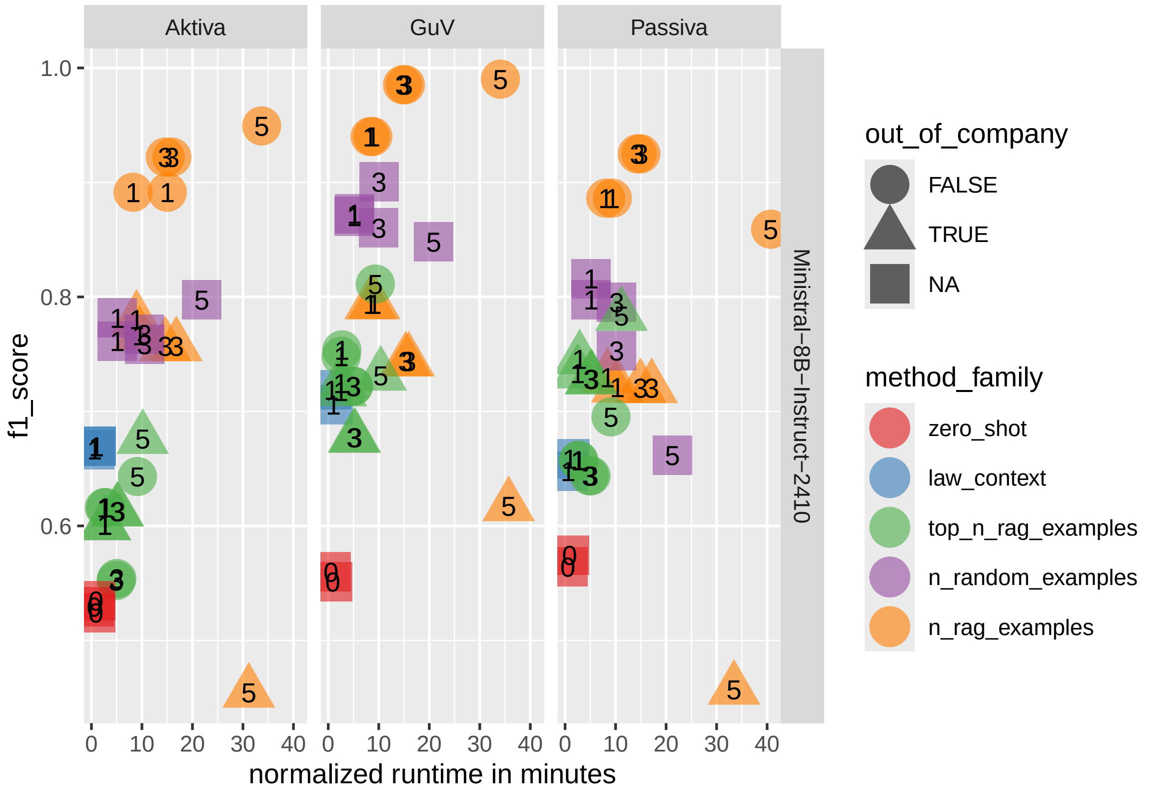


Figure A.9: Showing F1 score performance over normalized runtime for binary classification for Ministral-8B-Instruct-2410.

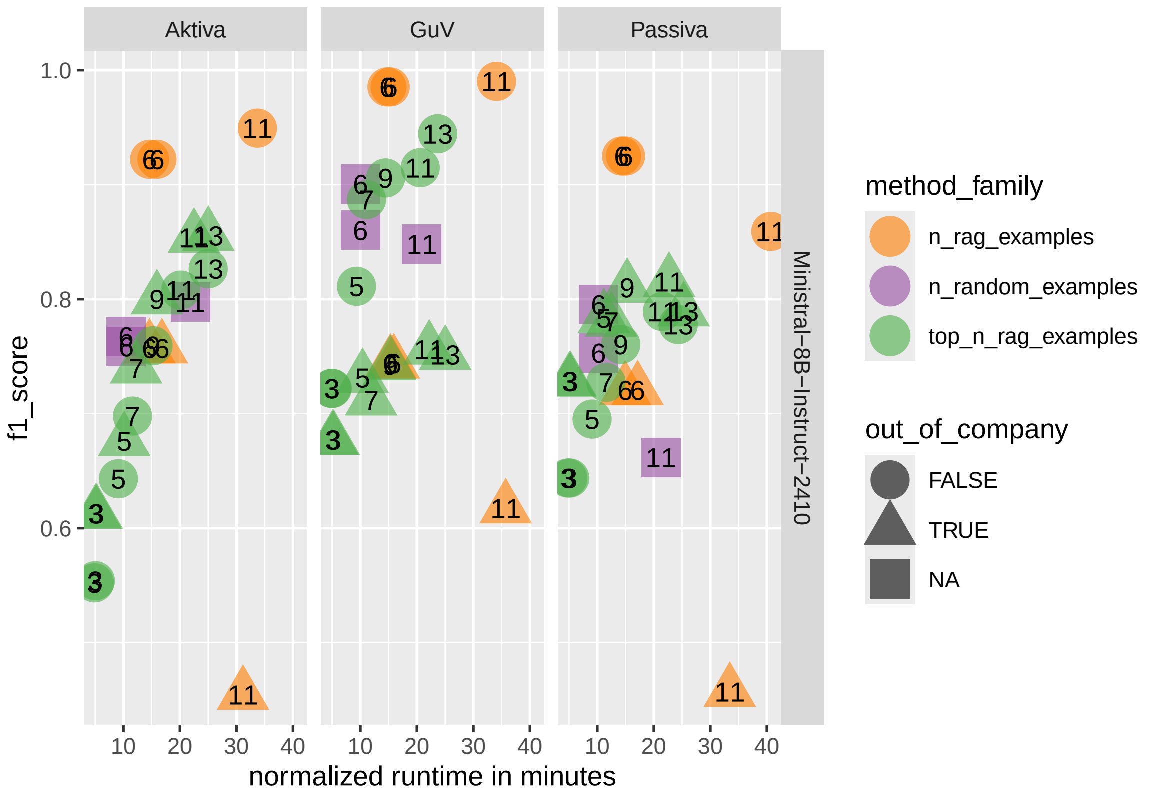


Figure A.10: Showing F1 score performance over normalized runtime for binary classification for Ministral-8B-Instruct-2410. Comparing the performance based on the real number of provided examples.

Summary:

Neither do newer generations always improve the performance for the binary classification task, nor do more parameters always improve or at least show stable performance.

A.3.2 Multi-class classification

Table A.9 shows that Llama-4-Scout solves the MCC task almost perfect for all classes. Mistral-Large-Instruct-2411 performs second best. In contrast to the binary classification task no order is visible, what class was easiest to predict. Googles gemma models perform much better in the MCC task with F1 scores of 0.89 instead of 0.58 for the binary classification task.

Table A.10 shows that the smaller models do perform good, too. Ministral-8B performs good but is around tenfold faster than Mistral-Large and Llama-4 Scout. For the larger models the *n_rag_examples* strategy is performing best. For the smaller models the *top_n_rag_examples* strategy is performing good as well and is faster because of shorter contexts.

Figure A.11 shows, the micro averaged F1 score for the three minority classes over the normalized runtime for two models. It shows the results for the different prompting strategies (*method_families*) with differently colored shapes.

The shape is giving information, if the examples provided to the LLM, are exclusively selected from other companies than the target table comes from, or if they can also be selected from documents of the same company. This is only relevant for strategies that get the examples picked by the documents vector embedding distances (*top_n_rag_examples* and *n_rag_examples*). The LLM performs better, if examples from documents of the same company can be used.

The number inside of the shapes is referring to the *n_examples* function parameter. Most models got benchmarked with an *n_examples* value of up to three. The actual number of examples provided to the models is depending on the method family / example selection strategy and can be looked up in Table F.1.

n_rag_examples better for Llama 4 Scout than *n_random_examples* ; For Ministral it is depending on the *out_of_company* setting

On can see, that Ministral–8B 2410 reaches a good performance already with few examples, but only if *out_of_company* is false. It performs moderate with the *law-context* strategy and *zero_shot*, too. Adding more examples does not improve the performance. Best with *top_n_rag_examples*

A

A.4 Term frequency based classifier

The fourth approach uses term frequencies for a key word list and the number of floats to rank pages. The approach is inspired by TF-IDF - a technique commonly used for information retrieval. It is similar to the baseline approach, because it uses a key word list and regular expressions to count terms and floats. But it is more flexible because the works in the key word list are not mandatory. This makes the approach robust against issues in the text extracts for single key words.

The key word list is generated removing the stop words from the law about **Aktiva**, **Passiva** and **GuV**. The key words from the regex approach are added, e.g. *GuV* and *Gewinn- und Verlustrechnung*. Since real life representations of those target types never contain all entries, it is not possible to include most of those words in a strict regex search as presented in the first approach.

This approach sums the counts of each word from the key word list per page in a first variable. In a second variable it counts the number of floats on each page. These two variables are then divided by the number of words found on the page. These densities are used to rank all pages from a single document. This is done with a unique key word list for each target type.

A random forest is trained to determine which density should be weighted to what amount. Because of the imbalanced data set undersampling is used when the training data set is created[13]. A single random forest is trained because the density of floats and specific words is assumed to be similar. The actual type of the

[13]The random forest build with undersampling performs much better as a classifier, that is trained using n oversamples train dataset.

Table A.9: Overview of benchmarked LLMs for the multiclass classification tasks. Limiting the number of examples provided for the few shot approach to 3.

model_family	model	metric_type	method_family	n_examples	f1_score	runtim
Llama-4	Llama-4-Scout-17B-16E-Instruct	Aktiva	n_rag_examples	3	1	
Llama-4	Llama-4-Scout-17B-16E-Instruct	GuV	n_rag_examples	1	1	
mistralai	Mistral-Large-Instruct-2411	Passiva	n_rag_examples	1	0.99	
Llama-4	Llama-4-Scout-17B-16E-Instruct	Passiva	n_rag_examples	3	0.99	
mistralai	Mistral-Large-Instruct-2411	Aktiva	n_rag_examples	3	0.98	1
Qwen 2.5	Qwen2.5-32B-Instruct	Aktiva	n_rag_examples	3	0.98	
Qwen 3	Qwen3-235B-A22B-Instruct-2507	GuV	n_rag_examples	3	0.97	1
Qwen 2.5	Qwen2.5-72B-Instruct	Passiva	n_rag_examples	1	0.97	
Qwen 3	Qwen3-30B-A3B-Instruct-2507	Aktiva	n_rag_examples	3	0.96	
Llama-3	Llama-3.1-70B-Instruct	Aktiva	top_n_rag_examples	3	0.96	
mistralai	Mistral-Large-Instruct-2411	GuV	n_rag_examples	1	0.96	
Llama-3	Llama-3.1-8B-Instruct	Passiva	n_rag_examples	1	0.95	
Qwen 3	Qwen3-235B-A22B-Instruct-2507	Passiva	n_rag_examples	3	0.95	1
Qwen 2.5	Qwen2.5-72B-Instruct	GuV	n_rag_examples	3	0.91	1
google	gemma-3-27b-it	Aktiva	n_rag_examples	3	0.89	
google	gemma-3-27b-it	Passiva	n_rag_examples	1	0.82	
google	gemma-3-27b-it	GuV	n_rag_examples	1	0.79	
tiiuae	Falcon3-10B-Instruct	GuV	n_rag_examples	1	0.71	
tiiuae	Falcon3-10B-Instruct	Aktiva	n_rag_examples	3	0.71	
microsoft	phi-4	Passiva	n_rag_examples	2	0.67	
Llama-3	Llama-3.1-8B-Instruct	GuV	top_n_rag_examples	1	0.65	
microsoft	phi-4	Aktiva	n_random_examples	1	0.6	
tiiuae	Falcon3-10B-Instruct	Passiva	top_n_rag_examples	3	0.59	
microsoft	phi-4	GuV	n_rag_examples	1	0.46	

Table A.10: Overview of benchmarked LLMs for the multiclass classification tasks focussing on models with less than 17B parameters. Limiting the number of examples provided for the few shot approach to 3.

model_family	model	metric_type	method_family	n_examples	f1_score	runtime in
mistralai	Ministral-8B-Instruct-2410	Aktiva	n_rag_examples	1	0.98	68
mistralai	Ministral-8B-Instruct-2410	Passiva	top_n_rag_examples	3	0.96	27
mistralai	Ministral-8B-Instruct-2410	GuV	top_n_rag_examples	3	0.95	27
Llama-3	Llama-3.1-8B-Instruct	Passiva	n_rag_examples	1	0.95	59
Qwen 2.5	Qwen2.5-3B-Instruct	Aktiva	n_rag_examples	1	0.86	49
Llama-3	Llama-3.1-8B-Instruct	Aktiva	top_n_rag_examples	3	0.86	26
google	gemma-3-12b-it	Aktiva	n_rag_examples	3	0.85	273
Qwen 2.5	Qwen2.5-3B-Instruct	Passiva	top_n_rag_examples	1	0.83	18
Qwen 2.5	Qwen2.5-3B-Instruct	GuV	n_rag_examples	1	0.76	49
tiiuae	Falcon3-10B-Instruct	GuV	n_rag_examples	1	0.71	86
tiiuae	Falcon3-10B-Instruct	Aktiva	n_rag_examples	3	0.71	239
google	gemma-3-12b-it	Passiva	n_rag_examples	3	0.69	273
microsoft	phi-4	Passiva	n_rag_examples	2	0.67	166
Llama-3	Llama-3.1-8B-Instruct	GuV	top_n_rag_examples	1	0.65	20
microsoft	phi-4	Aktiva	n_random_examples	1	0.6	49
tiiuae	Falcon3-10B-Instruct	Passiva	top_n_rag_examples	3	0.59	49
google	gemma-3-12b-it	GuV	top_n_rag_examples	1	0.47	23
microsoft	phi-4	GuV	n_rag_examples	1	0.46	172
Qwen 3	Qwen3-8B	Aktiva	law_context	1	0.07	6
Qwen 3	Qwen3-8B	Passiva	n_rag_examples	3	0.07	134
Qwen 3	Qwen3-8B	GuV	n_rag_examples	1	0	68
Qwen 3	Qwen3-8B	GuV	n_rag_examples	1	0	63
Qwen 3	Qwen3-8B	GuV	n_random_examples	1	0	31
Qwen 3	Qwen3-8B	GuV	n_rag_examples	3	0	134
Qwen 3	Qwen3-8B	GuV	n_rag_examples	3	0	130
Qwen 3	Qwen3-8B	GuV	n_random_examples	3	0	97
Qwen 3	Qwen3-8B	GuV	law_context	1	0	6
Qwen 3	Qwen3-8B	GuV	top_n_rag_examples	1	0	17
Qwen 3	Qwen3-8B	GuV	top_n_rag_examples	1	0	33
Qwen 3	Qwen3-8B	GuV	top_n_rag_examples	3	0	29
Qwen 3	Qwen3-8B	GuV	top_n_rag_examples	3	0	32
Qwen 3	Qwen3-8B	GuV	zero_shot	0	0	5

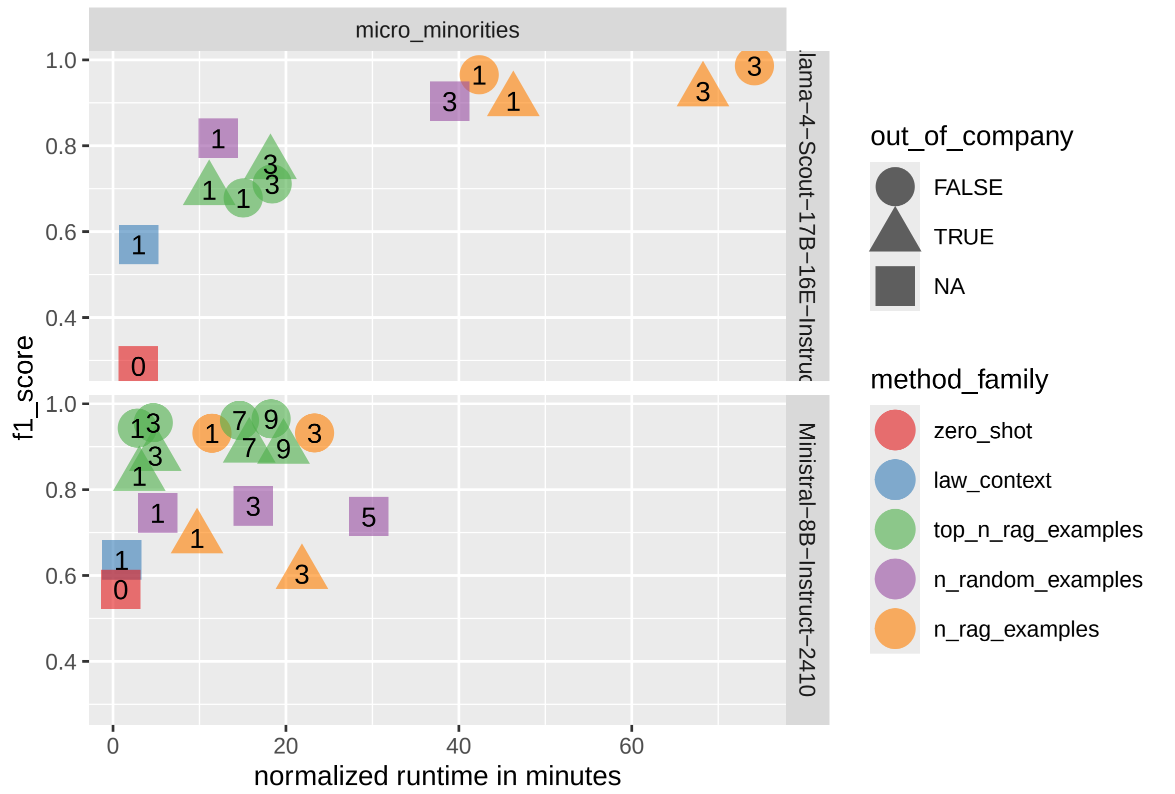


Figure A.11: Comparing F1 score micro averaged for the minority classes for two models over their normalized runtime.

page is not taken into account. The model just knows if the page is a page of any target type, based on the term and float density. This trippels the data points of the target class.

This single random forest performs much better than random forests that are trained using the dataset for each target type separatly. The performance is tested on all data points not included in the undersampled train dataset. Thus the test dataset is again highly imbalanced.

The random forest performs a binary classification task. But instead of the actual classifications, the predicted scores are used to rank the pages. Instead of precision or recall the metric used for the evaluation is top k recall. It is of interest which value of k is required to get a recall of 100 %.

The code can be found at: “benchmark_jobs/page_identification/term_frequency.ipynb”

- top 1
- top k

low precision llm linked to position of correct page? numeric frequency?

Figure A.12 shows how the test data points are distributed in the two dimensional value map for the random forest with two predictors. The target pages have a *float_frequncy* between 0.2 and 0.5 and pages with a *term_frequency* value over 0.07 get classified as target. One target page shows a lower *term_frequency* and thus does not get ranked correct. (recall, precision?)

A second random forest is trained supplementing the two predictors *term density* and *float density* with two additional predictors: *date count* and *integer count*. Figure A.13 shows the top n recall for both random forests. On the left side the top n recall on the imbalanced test dataset is shown. On the right side the performance on the train dataset.

Both random forests perform similar on the train dataset. The random forest with four predictors reaches perfect recall faster for **Aktiva** on the test dataset. Thus, with $n = 5$ 100 % recall is reached for the random forest with four predictors. With the random forest with two predictors it needs $n = 7$.

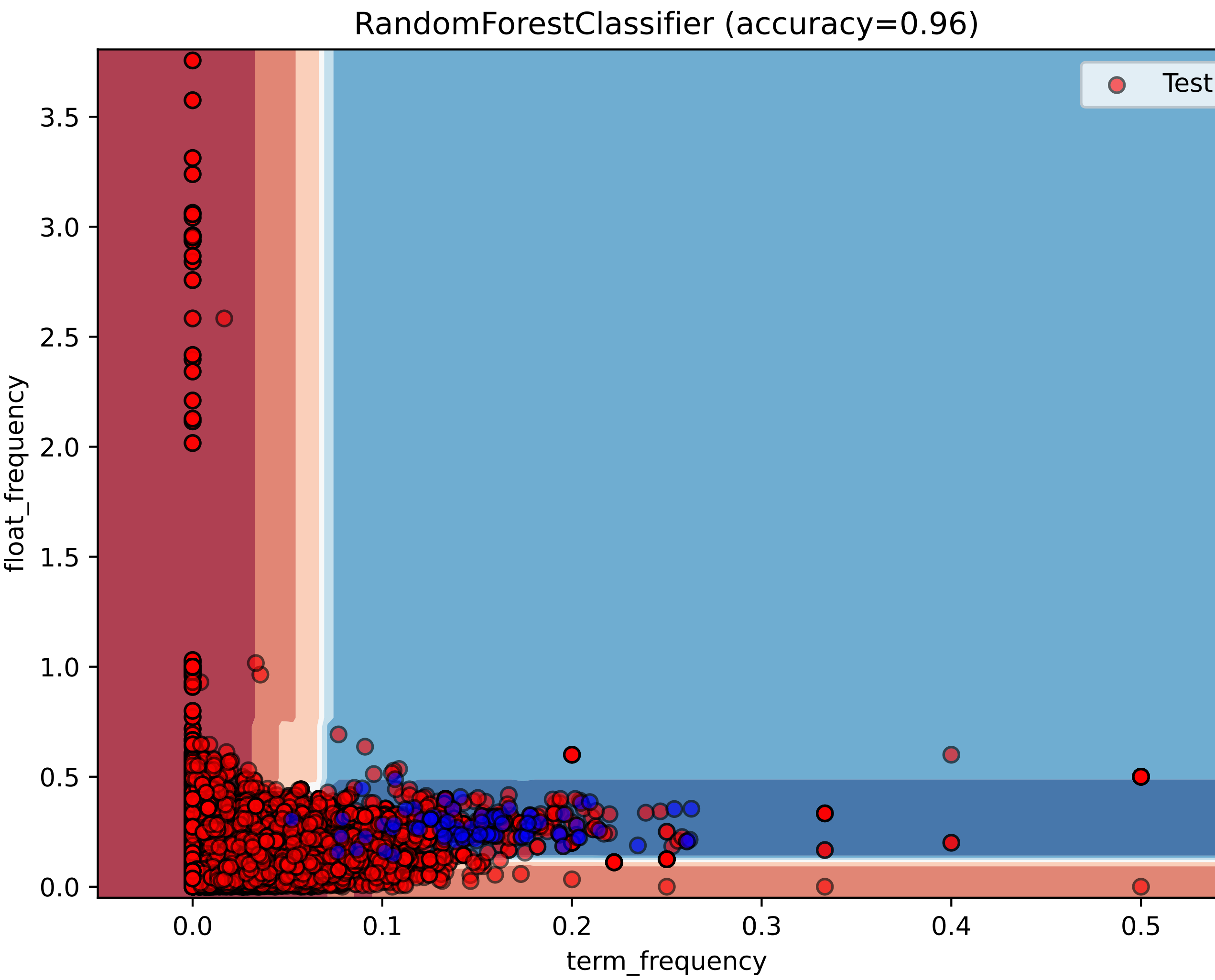


Figure A.12: Classification map showing which score a data point gets based on its term and float frequency and which type the data points in the test dataset actually have.

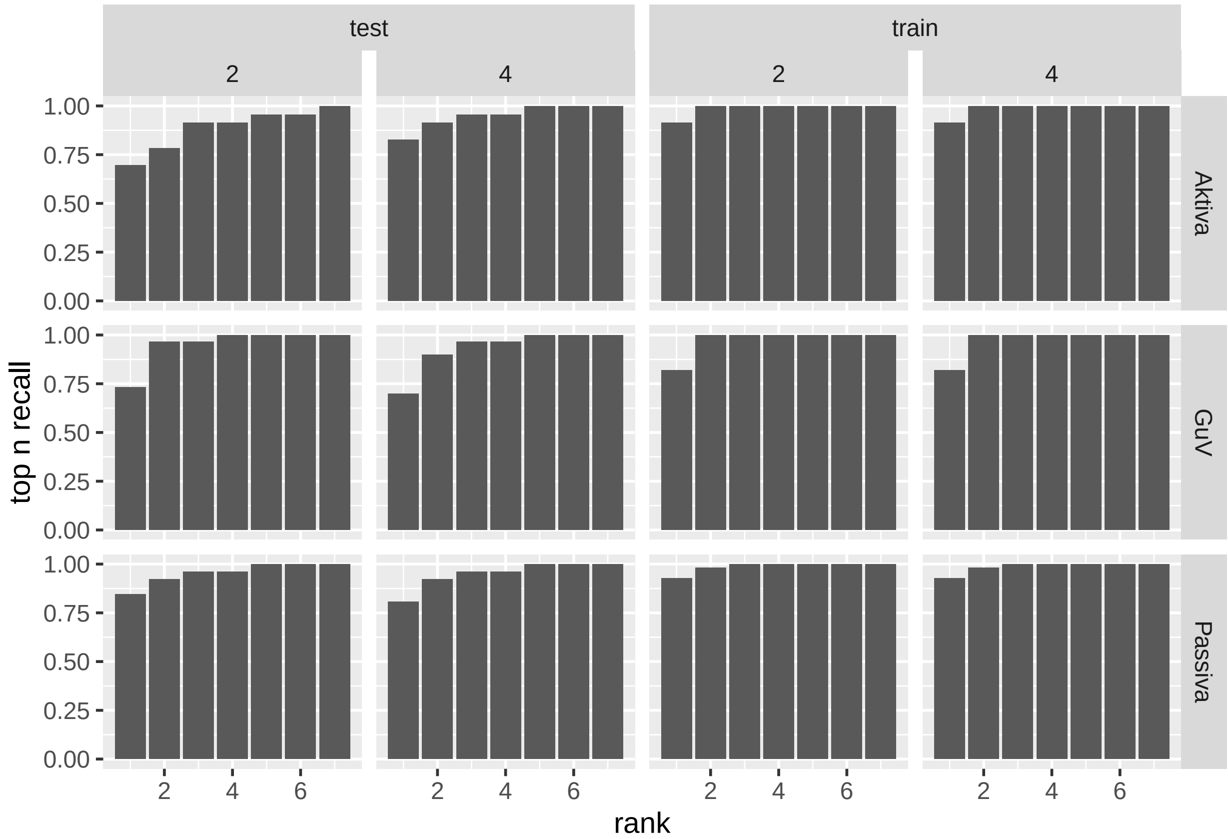


Figure A.13: Comparing the top n recall on training and test dataset among the random forest with two and four predictors.

A

Figure A.14 shows that the two additional predictors *date_count* and *integer_count* have little importance. But since it is computationally cheap to determine their value and the effiency of a random forest classifier, there is little reason not to use them.

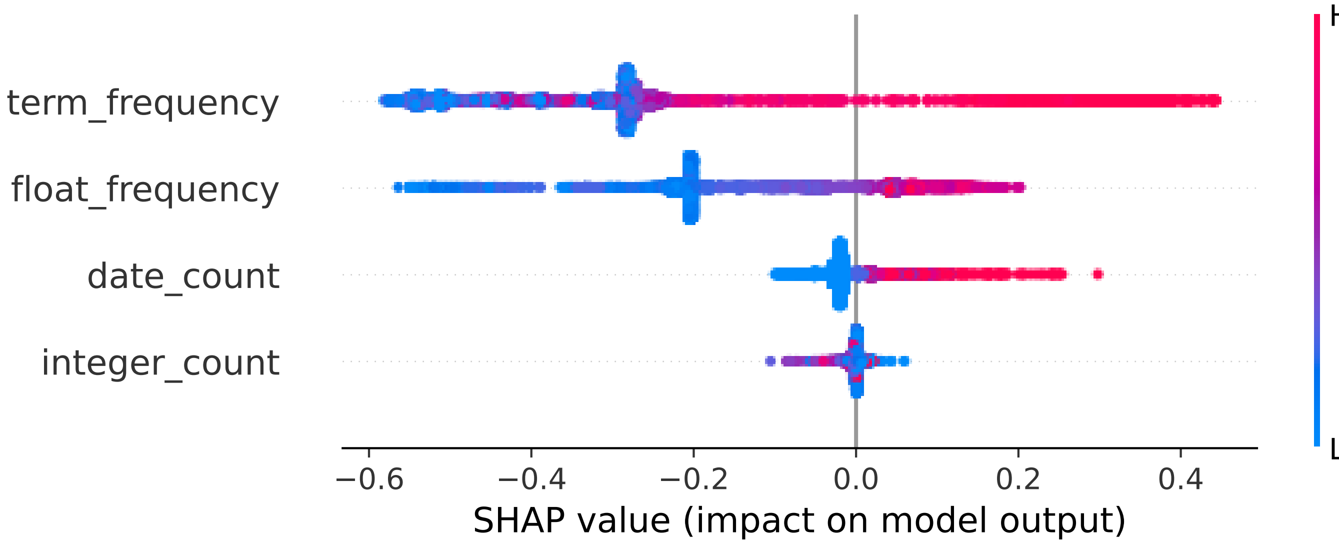


Figure A.14: Beeswarm plot of SHAP importance values for the four predictors of the second random forest classifier.

Fianlly, figure A.15 shows the precision-recall-curves for the term frequency approach for all three target types. The AUC for all types is below 0.5. The precision and F1 score stay below 0.5 as well. A high recall can be maintained for all types for threshold values up to at least 0.72.

A.5 Summary

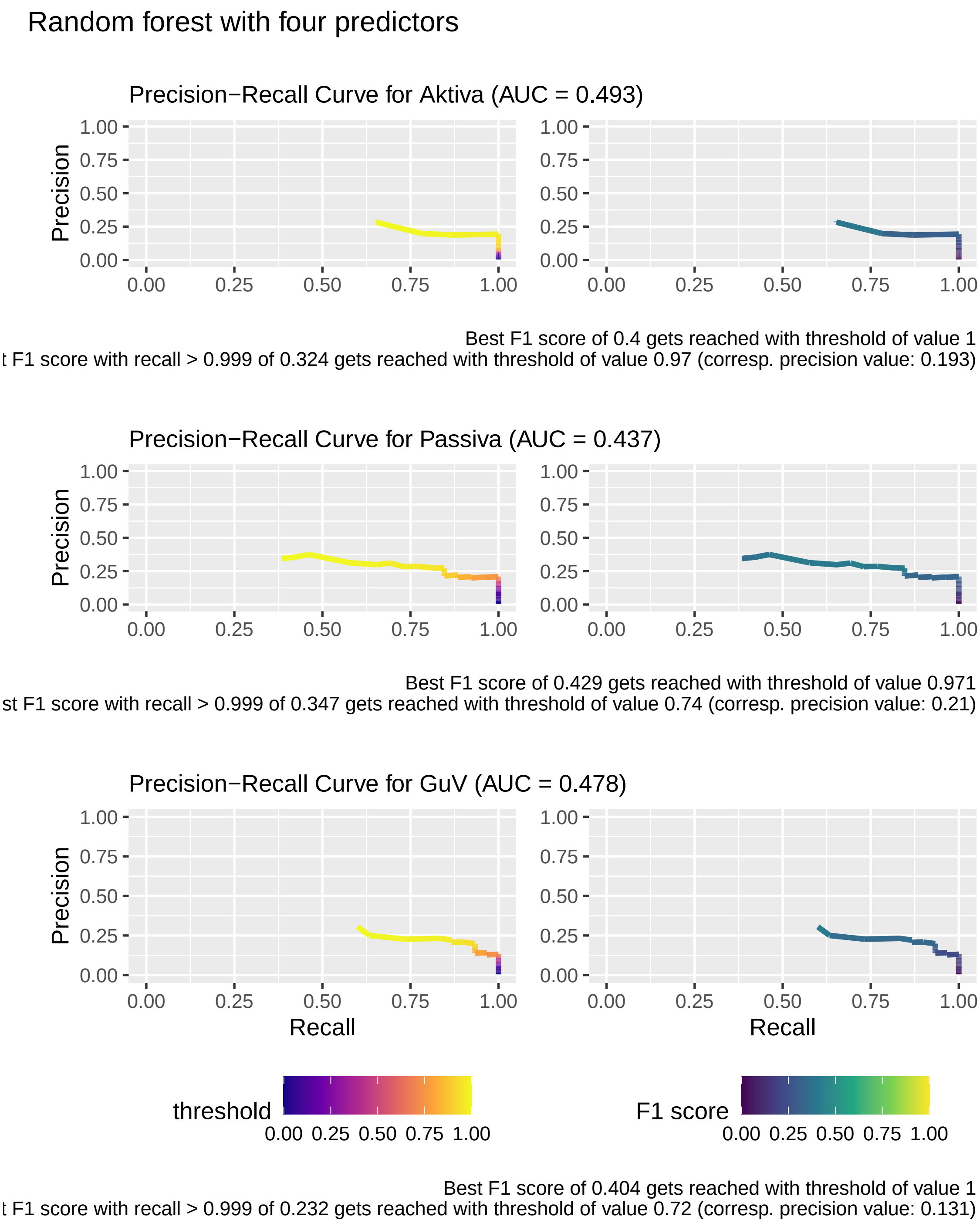


Figure A.15: Showing the precsion-recall-curve for the random forest with four predictors.

Chapter B

Appendix B - Information extraction report

The second research question asks, how LLMs can be used, to effectively extract specific information from a financial report. The task for this thesis is to extract the numeric values for the assets (*Aktiva*) table, which is part of the balance sheet (*Bilanz*). Hereafter, the German term **Aktiva** will be used. We are limiting the scope even further than in Subsection 5.1, because it takes more time to manually create the first reference dataset.

B.1 Baseline: Regex

The baseline for the table extraction task is set by an approach using regular expressions on the text extract. Figure B.1 shows the performance of this approach. In the first row (A) the percentage of correct predicted numeric values and the percentage of total correct responses is shown. The second row (B) shows the precision, recall and F1 score for identifying a value as missing and thus predicting *null*. The percentage of total correct responses is calculated as

$$percentage\_correct\_total = \frac{n_{correct\_numeric} + n_{missing\_true\_positive}}{n_{total\_entries}}$$

with $n_{total\_entries} = 58$. This implies that the correct prediction of missing values has more influence for tables, that have only a few numeric values in the ground truth. The minimal number of numeric values in a tables is ten. Figure G.1 shows, that the percentage of total correct responses is not a sufficient metric, because responses that only predicted *null* can have a high score if there are only a few numeric values in the ground truth table.

Performance In each frame there are two groups of two box-plots. The left group is showing the performance on real **Aktiva** tables. The right group shows the performance on synthetic **Aktiva** tables. Within the group the green (left) box shows the performance on text extracted with the *pdfium* library. The peach colored (right) box shows the performance on text extracted with the *pymupdf* library.

Figure B.1 shows, that the regex approach performs better[1] on the synthetic tables compared to the real tables. Even though, the performance is not perfect and more consistent on the text extracted with *pymupdf* compared to *pdfium*. In contrast, the used text extraction library has no noticeable influence on the real **Aktiva** tables.

The performance for the regex based table extraction is much better than the regex based page identification performance. The median performance scores of the regex approaches will be reflected by a dashed line in the box-plots in subsequent sections. The scores for the real **Aktiva** table extraction are:

[1]A comparison of the numeric values over all methods can be found in section 5.2.2.

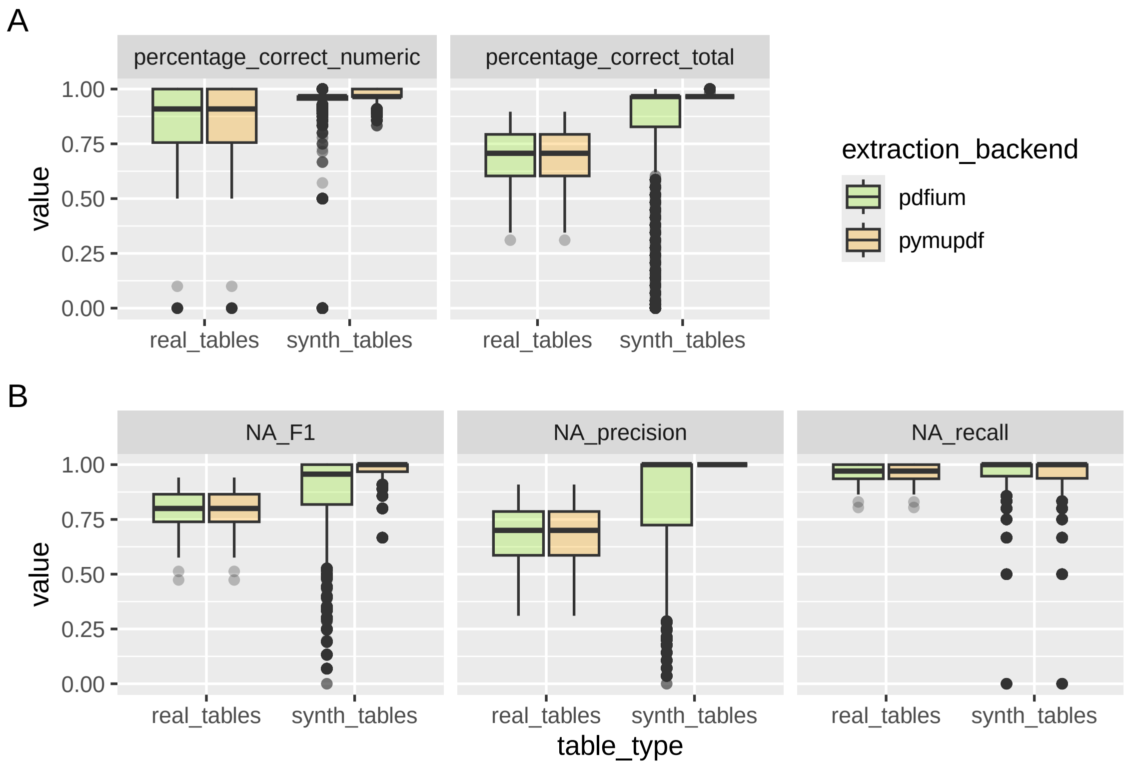


Figure B.1: Performance overall and on numeric value extraction with regular expressions.

B

Table B.1: Summarizing the median performance of the regex approaches for the real and synthetic table extracion task.

measure	real_mean	real_median	synthetic_mean	synthetic_median
percentage of correct predictions total	0.686	0.707	0.969	0.966
percentage of correct numeric predictions	0.778	0.909	0.973	0.966
F1 score	0.789	0.800	0.979	1.000

The difference in the precision scores for the regex approach on synthetic tables is discussed in section 6.2.2.5.

B.2 Extraction with LLMs

This section presents the results for the table extraction task performed with LLMs on **Aktiva** tables. Subsection B.2.1 compraes the performance of open source models on real tables. It also compares those results with the table extraction performance achieved with models by OpenAI.

Subsection B.2.2 presents the results on synthetic **Aktiva** tables. Subsection B.2.3 shows a hybrid approach, where synthetic tables are used for the in-context learning, to extract real **Aktiva** tables. Finally, we summarize the results for all approaches in subsection 5.2.2.

- confidence usable to head for user checks?
- not handled new entries
- five examples bring not much more, but a little

Explain *static_example* method.

B.2.1 Real tables

Performance For the table extraction task 32 open source models from 9 model families have been benchmarked[2]. Table B.2 shows the best performing combination of LLM and prompting strategy for each model family. The results are sorted by their mean percentage of total correct predictions. It also shows the normalized runtime in seconds and the parameter number of the model.

Qwen3-235B-A22B-Instruct performed best with a mean score of 0.97. This is equal with the performance that we achieved building the ground truth dataset. There are other models that perform almost as good with a score of 0.96 and more, but that don't match the human performance. These models show a median score of 1.0. Qwen3-235B-A22B-Instruct is the second fastest of those well performing models and needs less than six minutes. Only Llama 4 Maverick is faster. It needs half the time to extract the information.

Table B.3 shows the performance of models with less than 17B parameters, that have not been listed above. Qwen3-14B performs best among the smaller models achieving a mean of 0.93 and median of 1.0. It takes 1:49 minutes to extract the information from all **Aktiva** tables. Ministral-8B-Instruct does not perform as good as in the page identification task.

Most models need a context learning approach to beat the performance of the regular expression approach at total and numeric correctness rate and F1 score. Table B.4 shows, that 3 models perform better without

[2]The models *deepseek-ai_DeepSeek-R1-Distill-Qwen-32B* and *google_gemma-3n-E4B-it* have been tested as well but don't get presented as they never performed anywhere beyond random guessing.

Table B.2: Comparing best table extraction performance with real 'Aktiva' dataset for each model family

model_family	model	method_family	n_examples	parameter_count
Qwen 3	Qwen3-235B-A22B-Instruct-2507-FP8	top_n_rag_examples	5	235
mistralai	Mistral-Large-Instruct-2411	top_n_rag_examples	5	124
Qwen 2.5	Qwen2.5-72B-Instruct	top_n_rag_examples	5	72
Llama-4	Llama-4-Maverick-17B-128E-Instruct-FP8	top_n_rag_examples	1	402
Llama-3	Llama-3.1-70B-Instruct	top_n_rag_examples	5	70
microsoft	phi-4	top_n_rag_examples	2	15
openai	gpt-oss-120b	top_n_rag_examples	5	120
tiiuae	Falcon3-10B-Instruct	top_n_rag_examples	3	10
google	gemma-3-27b-it	top_n_rag_examples	3	27

Table B.3: Comparing best table extraction performance with real 'Aktiva' dataset for each model family for models with less than 17B parameters. Models that have been listes in the previous table are not listed again.

model_family	model	method_family	n_examples	parameter_count	mean_total	media
Qwen 3	Qwen3-8B	top_n_rag_examples	5	8	0.927	
Qwen 2.5	Qwen2.5-14B-Instruct	top_n_rag_examples	5	14	0.925	
mistralai	Ministral-8B-Instruct-2410	top_n_rag_examples	5	8	0.895	
Llama-3	Llama-3.1-8B-Instruct	top_n_rag_examples	5	8	0.832	
google	gemma-3-12b-it	top_n_rag_examples	3	12	0.811	

any guidance[3]. 6 models achieved an performance better than the regex baseline using the approach to learn with a fixed example from the synthetic dataset.

In contrast: most of the models achieved a better performance than the regex baseline when they were provided with one or more examples from real **Aktiva** tables. Table B.5 shows, that 11 don't consitently achieve a better score, when provided with three or five real **Aktiva** table examples. Here we find the smallest models with less than 2B parameters which don't achieve a consistence performance no matter how many examples they get. But we also find models that start to perform bad if they get a too long context with too many examples like the very recent and large model Llama 4 Maverick.

The results for all models are presented in Figure G.15, G.16 and G.17. In general the performance within a model family is positive correlated with the models number of parameters, if we provide real **Aktiva** examples. Once the 4B parameters are passed, the improvements get less and less, approaching the perfect performance. But no model achieves sperfect result on all documents. The *zero_shot* and *static_example* approach show some unpredicted performance drop, i.e. for Qwen3-14B.

OpenAI models Even though a lot of documents to process at RHvB will not be public and thus must not be processed on public cloud infrastructure, the performance of models like OpenAI's GPT are interesting benchmark references within this thesis and for comparing these findings with other papers results. Therefore for this thesis the public available versions of annual reports have been used instead of the ones used internally or for public administration purposes. Those public available reports often are visually more appealing and more heterogeneous in their structure.

Table B.6 shows the ranking for the best model-method combinations Qwen3 235B is performing best. gpt-4.1 and gpt-5-mini perform equally well and are almost as good as Qwen3 235B. All models but gpt-4.1-nano, gpt-5-nano and Qwen3-0.6B manage to beat the regex threshold. Qwen3-0.6B performs better than the nano models once it gets provided with an example.

Table B.7 shows the accumulated costs for the table extraction task for the models provided by Azure. Using gpt-4.1 is most expensive, followed by gpt-5-mini. Next is gpt-5-nano. This is caused by an unexpected high cost for output tokens. In general we find, that the ratio of output costs to input costs is much higher

[3]There is an external guidance through the provided xgrammar template but it is not communicated to the model in form of a promt.

Table B.4: Comparing table extraction performance with real 'Aktiva' dataset for models that perform well without or with little context learning

model	median_total_zero_shot	median_total_static_example
Llama-4-Maverick-17B-128E-Instruct-FP8	**0.897**	0.922
Qwen3-235B-A22B-Instruct-2507-FP8	**0.897**	**0.931**
gpt-oss-120b	**0.897**	0.897
Qwen2.5-32B-Instruct	NA	**0.931**
Qwen2.5-72B-Instruct	NA	0.897
Qwen3-30B-A3B-Instruct-2507	NA	0.879

Table B.5: Comparing table extraction performance with real 'Aktiva' dataset for models that perform worse than the regex baselin with 3 or 5 examples for incontext learning

model	method	parameter_count	median_total
Llama-3.1-8B-Instruct	3_random_examples	8	0.81
Llama-4-Maverick-17B-128E-Instruct-FP8	5_random_examples	**402**	0.017
Qwen2.5-0.5B-Instruct	3_random_examples	0.5	0.586
Qwen2.5-1.5B-Instruct	3_random_examples	1.5	0.724
Qwen2.5-3B-Instruct	3_random_examples	3	0.759
Qwen2.5-7B-Instruct	3_random_examples	7	0.862
Qwen3-0.6B	3_random_examples	0.6	0.612
Qwen3-1.7B	3_random_examples	1.7	0.776
gemma-3-12b-it	3_random_examples	12	0.793
gemma-3-4b-it	3_random_examples	4	0.664
gpt-oss-20b	3_random_examples	20	**0.897**

for gpt-5 models. Since gpt-5-mini gives consistently good results already with one provided example, this could be the most cost efficient strategy. But it takes gpt-5-mini more than three times longer to respond than gpt-4.1.

Discussion?:

Since the output token costs are not that different (2 $ for 1M output tokens with gpt-5-mini vs 1.6 $ with gpt-4.1-mini), the generated output token number has to be much higher for the gpt-5-mini models. But since the responses are based on the same schema and required the same numeric values there shouldn't be a big difference[4].

Figure B.2 shows the distribution of F1 score for up to three examples. It shows green crosses at the bottom of the abscissa that indicate prediction, where no *null* value is reported. This means, the model hallucinates many numeric values. This is only the case for OpenAI's models but not for Qwen3 models. This behaviour persists up to five examples for the nano as well the gpt-oss 20b model. For gpt-4.1 and gpt-4.1-mini these cases vanish when we provide three or more examples and never appeared for gpt-5-mini.

One can find the full plots in Figures G.19, G.20 and G.21).

We were not able to get OpenAI's models to stick to the provided json schema strictly. Passing the ebnf

[4]With the gpt-oss models we found the new Harmony response format to produce a lot of tokens in the chain of thought stream, we discarded, because we only need the json in the final stream Maybe this is similar for gpt-5 models as well but the chain of thought stream is kept on Azures side?

B

Table B.6: Comparing table extraction performance with real 'Aktiva' dataset for OPenAIs GPT models with a selection of Qwen3 models.

model	method	mean_percentage_correct_total	median correct total
Qwen3-235B-A22B-Instruct-2507-FP8	5_random_examples	0.97	1
gpt-4.1	top_5_rag_examples	0.95	0.97
gpt-5-mini	top_5_rag_examples	0.95	0.97
Qwen3-30B-A3B-Instruct-2507	top_5_rag_examples	0.92	0.97
gpt-4.1-mini	top_3_rag_examples	0.91	0.96
Qwen3-8B	top_5_rag_examples	0.89	0.93
gpt-oss-120b	5_random_examples	0.88	0.93
gpt-oss-20b	3_random_examples	0.84	0.9
Qwen3-0.6B	5_random_examples	0.65	0.65
gpt-5-nano	3_random_examples	0.3	0.24
gpt-4.1-nano	zero_shot	0.21	0.14

Table B.7: Comparing the costs for OpenAIs GPT models provided by Azure. Notice the high output cost for GPT 5 Nano.

model	cost_input	cost_output	cost_total	median runtime in minutes
gpt-4.1	18.07	10.35	28.42	29:53
gpt-5-mini	1.93	10.28	12.21	110:50
gpt-5-nano	0.41	6.99	7.4	135:37
gpt-4.1-mini	3.76	2.02	5.78	31:48
gpt-4.1-nano	0.08	0.06	0.14	10:28

grammar did not work at all. This means that with gpt-4.1-nano there have been 88 predictions that have been completely empty. For gpt-5-nano we find 6 such predictions. Figure @red(fig:table-extraction-llm-prediction-count-gpt) shows the distribution of responses with a wrong number of predictions (including *null* and numeric predictions). Overall there have been 34.3 % of the responses of OpenAI's models that were compatible with the schema but had a wrong number of rows predicted. The maximum number of returned values (by gpt-5-nano) is 714.

Using gpt-5-chat for the table extraction task with structured output is not working, returning an error informing that a *json_schema* can't be used with this model. Figure B.4 shows, where other models produced an answer that could not be parsed as valid json.Most errors occured for gpt-oss-20B and the *static_example* method. Over half of all tables could not be transcripted in json with in the 40_000 response token limit[5]. Only with gpt-5-mini we had no json parsing error.

Examples from same company performance Table B.8 shows the improvement for the percentage of correct predictions total, when **Aktiva** tables from the same company as the target tables company are provided for the in-context learning. It shows that this improvement is biggest for goolge and Qwen and smallest for Llama models.

Figure G.18 shows, that using **Aktiva** in-company examples improves the performance, mainly by reducing the number of bad predictions. The found improvement is present for all models but Llama 4 Maverick. Here the number of bad predictions gets larger if we provide three or more examples. With five examples the performances totally collapses.

The perfromance improvement for GPT-4.1-mini and GTP-4.1 with only one provided example seems to be big, because the box is getting much more narrow. But the median shifts not more than for other models.

Check results for openai, when 5 nano finished

[5] Without the Harmony format 4_000 are enough.

Table B.8: Comparing the extraction performance when Aktiva tables from the same company can be used for incontext learning or not.

model_family	improvement_mean	improvement_median
google	0.13	0.14
Qwen 3	0.12	0.07
chat-gpt	0.12	0.17
Qwen 2.5	0.1	0.12
tiiuae	0.09	0.07
mistralai	0.08	0.07
Llama-3	0.07	0.05
microsoft	0.07	0.06
openai	0.05	0.07
Llama-4	0.03	0

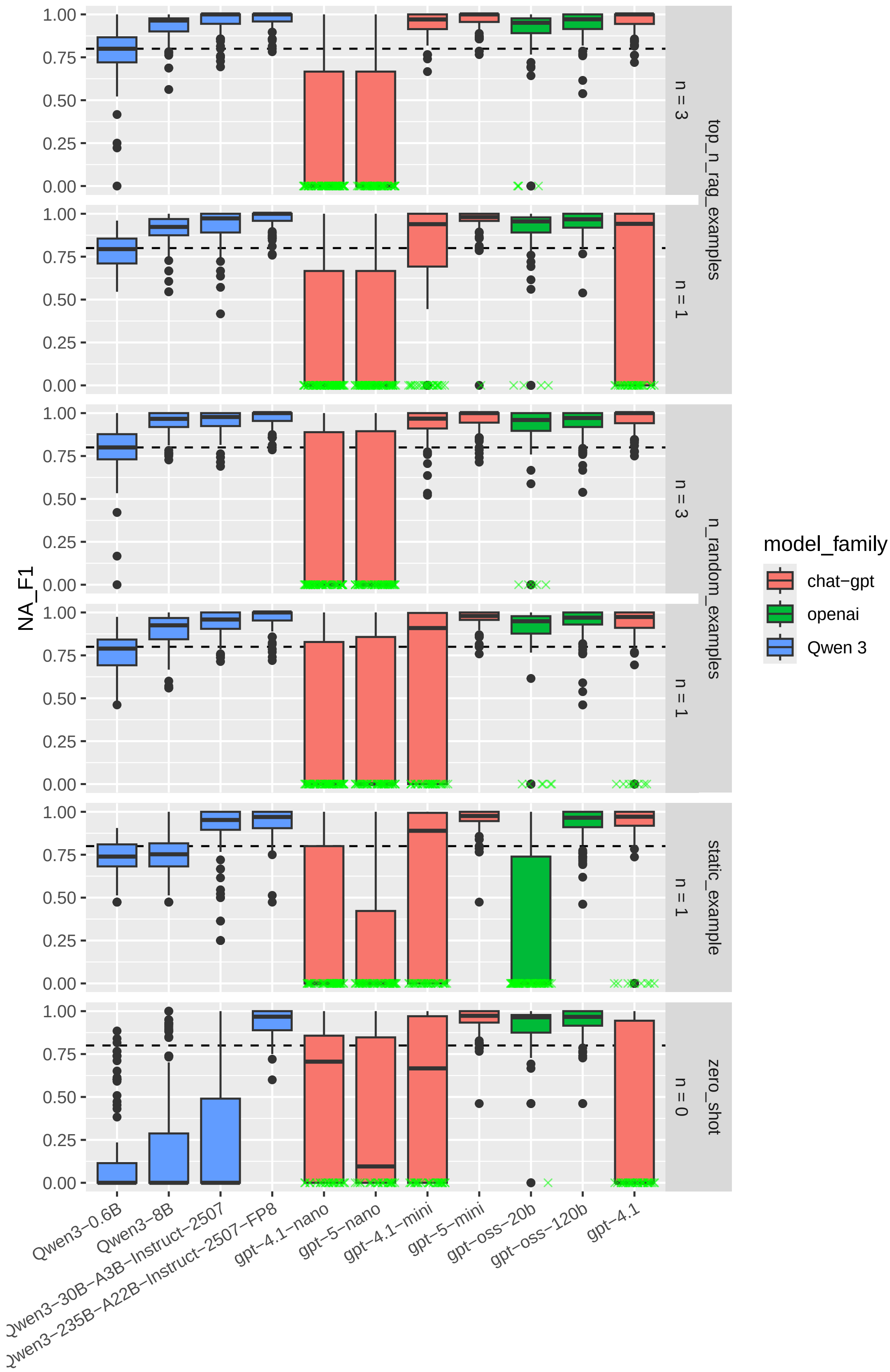


Figure B.2: Comparing the F1 score for predicting the missingness of a value for OpenAi's LLMs with some Qwen 3 models. The green crosses indicate results where a model has predicted only numeric values even though there have been missing values.

B

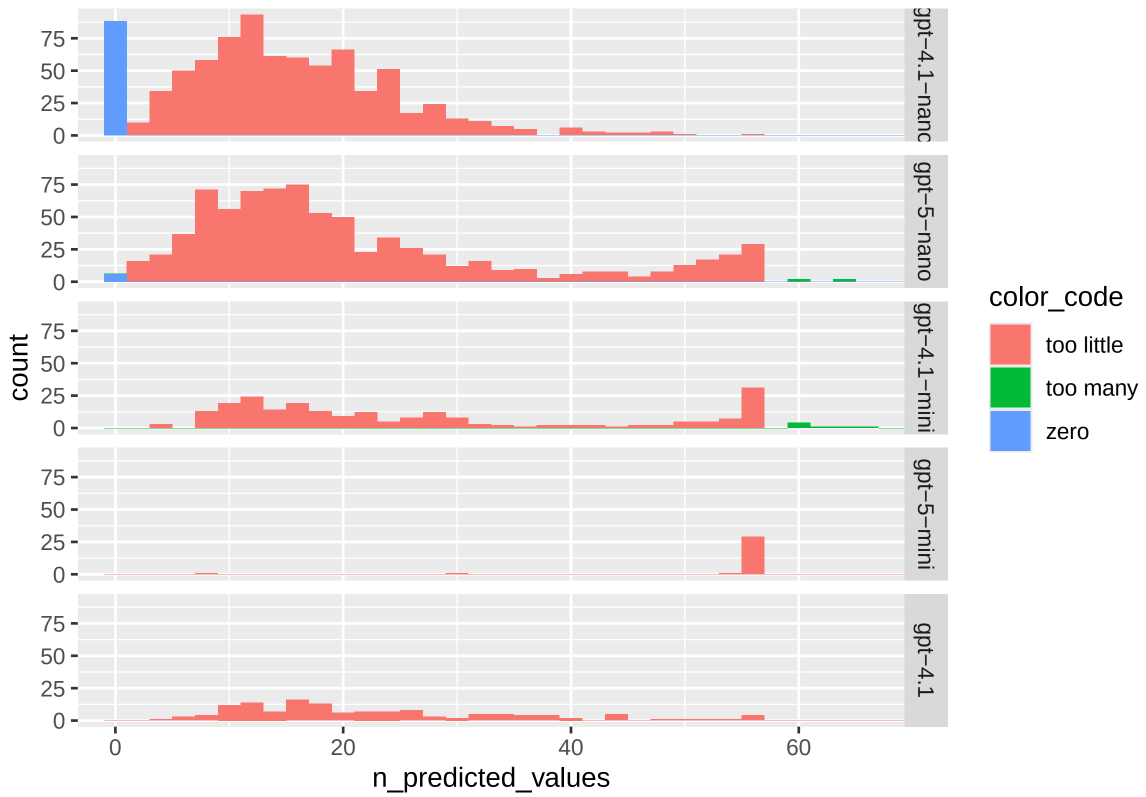


Figure B.3: Showing the number of predictions OpenAI's models made.

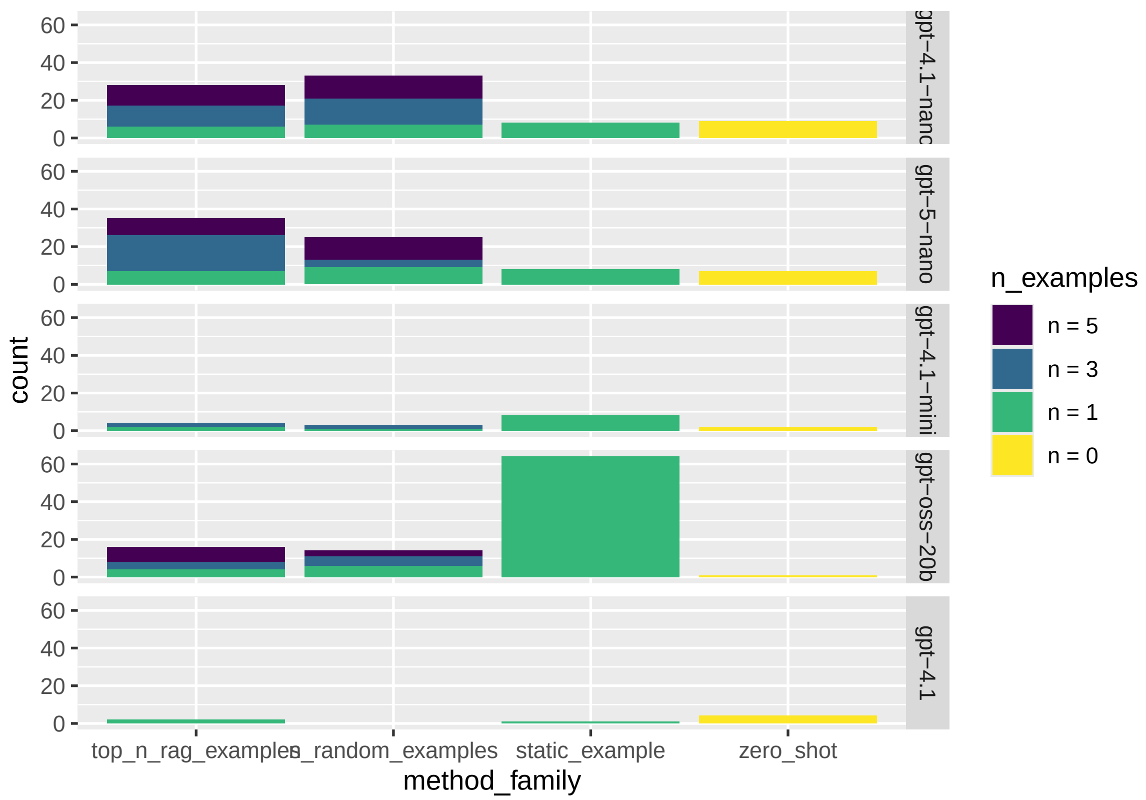


Figure B.4: Showing the number of predictions where not 58 values are returned grouped by model and prompting method.

B.2.2 Synthetic tables

Ground truth dataset For this task we created synthetic **Aktiva** tables that should allow to investigate the influence certain characteristics of tables on the extraction task. We systematically created **Aktiva** tables that vary over the following characteristics:

1. n_columns: Number of columns the numeric values are distributed over ranging from 2 to 4.
2. header_span: Span in the header rows.
3. thin: Including just a subset of all possible entries for a **Aktiva** table.
4. unit_in_first_cell: Is the currency unit (e.g. T€) given in the beginning of the table instead for each column.
5. enumeration: Are the rows numerated following the schema in the legal text.
6. many_line_breaks: Limiting the character length for the row descriptions to 50 to introduce line breaks.
7. shuffle_rows: The order of the rows within lower hierarchies can vary.
8. text_around: There is some random text before and after the table on the generated page.
9. sum_same_line: Summed values are in the same line as single values if there are more than two columns.
10. unit: Eight different currency units from. E.g.: €, TEUR, Mio. €
11. input_format: Table is exported as PDF, HTML or Markdown document.

This results in 49152 tables. A sample of 10 % is used for the extraction task. The *header_span* and *text_around* are only varied for the PDF format.

Extraction task overview For the table extraction task with synthetic **Aktiva** tables 17 open source models from 6 model families have been benchmarked. There have been the same seven methods tested with each LLM as described in section B.2. Each method was used twice, because in one trial the LLM is prompted to respect the currency units and in the other trail it is not.

This results in 531 files, that hold the results of 4_915 table extractions each. For the investigation of potential predictors influences the random forest is generated with a sample of 50_000 of these 68_810 results and finally, the SHAP values are calculated with 2_000 rows of data.

Performance Table B.9 shows the best performing combination of LLM and prompting strategy for each model family. The results are sorted by their mean percentage of total correct predictions. We only compare results for table extractions that work with a PDF document based table here.

For every model family there is at least one model-method combination that performed better than the regex baseline. For the synthetic table extraction task the baseline is 0.966. 67 from 129 model-method combinations perform better than this baseline. There has been no model that performed better than this baseline with the *zero_shot* or *static_example* method.

Table B.9 shows, that Qwen3-235B-A22B-Instruct performs best. Llama 4 Scout also performs very good but is three times faster. Table B.10 shows three small LLMs that also beat the median threshold for the synthetic table extraction task. But we would not prefer the Qwen3-8B model over the Llama Scout model, because its speed advantage is to small, compared to the performance decrease. But if there is limited VRAM available the Qwen3 model is a good choice. It can run well with 40 GB VRAM. The Llama Scout needs 640 GB VRAM to run well[6].

Detail:

Figure G.24 shows, that Llama 3.3 70B never manages to reduce the spread in the numeric prediction performance.

[6]When we say, it runs well, it gets rated as *okay* on the LLM Inference: VRAM & Performance Calculator.

Table B.9: Comparing best median table extraction performance with synthetic 'Aktiva' dataset for each model family

model_family	model	method_family	n_examples	mean_total	median_total	me
Qwen 3	Qwen3-235B-A22B-Instruct-2507	n_random_examples	5	0.991	**1**	
Qwen 2.5	Qwen2.5-72B-Instruct	top_n_rag_examples	5	0.988	**1**	
mistralai	Mistral-Large-Instruct-2411	top_n_rag_examples	5	0.987	**1**	
Llama-4	Llama-4-Scout-17B-16E-Instruct	top_n_rag_examples	3	0.974	**1**	
google	gemma-3-27b-it	top_n_rag_examples	5	0.914	0.931	
Llama-3	Llama-3.1-8B-Instruct	top_n_rag_examples	3	0.873	0.931	

Table B.10: Comparing best median table extraction performance with synthetic 'Aktiva' dataset for each model family for models with less than 17B parameters

model_family	model	method_family	n_examples	mean_total	median_total	median_r
Qwen 3	Qwen3-8B	top_n_rag_examples	3	0.944	**1**	18:38
Qwen 2.5	Qwen2.5-7B-Instruct	top_n_rag_examples	5	0.919	**0.966**	15:26
mistralai	Ministral-8B-Instruct-2410	top_n_rag_examples	3	0.908	**0.983**	31:12
Llama-3	Llama-3.1-8B-Instruct	top_n_rag_examples	3	0.873	0.931	14:8
google	gemma-3-12b-it	top_n_rag_examples	5	0.858	0.931	26:28

B.2.3 Hybrid approach

In this section we present the results of using synthetic **Aktiva** tables for the in-context learning to extract information from real **Aktiva** tables. We show that even such a hybrid approach can be used, to extend the extraction task by a unit conversation task.

Performance Table B.11 compares the overall performance for the extraction task of the best model-method combination in the hybrid approach per model with the *zero_shot* and real example training performance. Using real examples for in-context-learning for those model-method combinations is better than using the generated synthetic data. Qwen3-8B and gemma3-12b can improve the most using real examples instead of sythetic examples, normalized on the possible improvement from the synthetic learning results using this formula:

$$delta\_rate_{synth} = \frac{median(real) - median(synth)}{1 - median(synth)}$$

On the same time, gemma3-12b shows the lowest *delta_rate* with under 10 %, when the improvement of using synthetic examples is compared with the *zero_shot* method. For the other models this is more tan 48 % and highest for Llama Scout 4 with 87.5 % improving from 0.45 to 0.93. Qwen-235B score as high with both learning approaches, but scored best with just using a single synthetic example. Table F.2 shows that these observations are valid for the improvement with in the models independent from the selected method. Figure G.27 shows, that the improvement for using one or three synthetic examples is biggest for Qwen3-8B.

Learning to respect currency units Table B.12 shows, the difference in the percentage of correct predicted numeric values, if the LLM is prompted to respect currency units and gets synthetic **Aktiva** tables that show how to cope with different currency units, separate for the number of columns with currency units. There are 17 tables that have *T€* in the previous year column and 9 tables that have all columns listed in *T€*.

It shows, that Qwen3-235B, Llama 4 Scout, Mistral-Large and Ministral-8B all can apply the demonstrated numeric transformation for most of the values, if both columns have the *T€* unit. Qwen3-235B, Llama 4 Scout and Mistral-Large also can apply this, if only one columns has a unit corrency. This works best for

Table B.11: Comparing extraction performance for real Aktiva extraction task with synthetic and real examples for incontext learning with a zero shot approach for the best performing modelmethod combination in the hybrid

model	method	median_real	median_synth	median_zero_shot	del
Qwen3-235B-A22B-Instruct-2507-FP8	1_random_examples	**0.966**	**0.966**	**0.897**	
Llama-4-Scout-17B-16E-Instruct	5_random_examples	**0.966**	0.931	0.448	
Mistral-Large-Instruct-2411	5_random_examples	**0.966**	0.922	0.776	
Qwen3-8B	5_random_examples	0.94	0.802	0.336	
Llama-3.1-8B-Instruct	5_random_examples	0.836	0.776	0.552	
Ministral-8B-Instruct-2410	5_random_examples	0.897	0.767	0.552	
gemma-3-27b-it	3_random_examples	0.828	0.724	0.207	
gemma-3-12b-it	top_1_rag_examples	0.862	0.586	0.543	

Qwen3-235B. The target value to archive here is 0.5 instead of 1.0. This is worth to mentioning because there are no synthetic examples that haf different currency units for different columns. Ministral can not generalize this skill. It seems, that Qwen3 apllies numeric transformations regardles the fact, if there are currency units given for a column. Thus, it performs noticeably worse on the majority of all tables. Figure B.5 shows, that the performance of Llama 3.1 8B and gemma3 27B on colums with currency units does not change.

G.30, G.31 and G.32

Thus, synthetic data can be used to solve new tasks and substitute missing data for rare classes.

B.3 Summary

Table B.12: Comparing extraction performance for real Aktiva extraction task dependent on the promt addition to respect currency units and providing examples that show this transformation.

model	n_cols_T_EUR_0	n_cols_T_EUR_1	n_cols_T_EUR_2
Llama-3.1-8B-Instruct	-0.03	-0.02	0.01
Llama-4-Scout-17B-16E-Instruct	**0**	0.28	0.89
Ministral-8B-Instruct-2410	-0.08	0	0.79
Mistral-Large-Instruct-2411	**0**	0.39	0.9
Qwen3-235B-A22B-Instruct-2507-FP8	**0**	**0.44**	**0.96**
Qwen3-8B	-0.48	0	0.28
gemma-3-12b-it	-0.03	0.17	0.48
gemma-3-27b-it	**0**	0.06	0.02

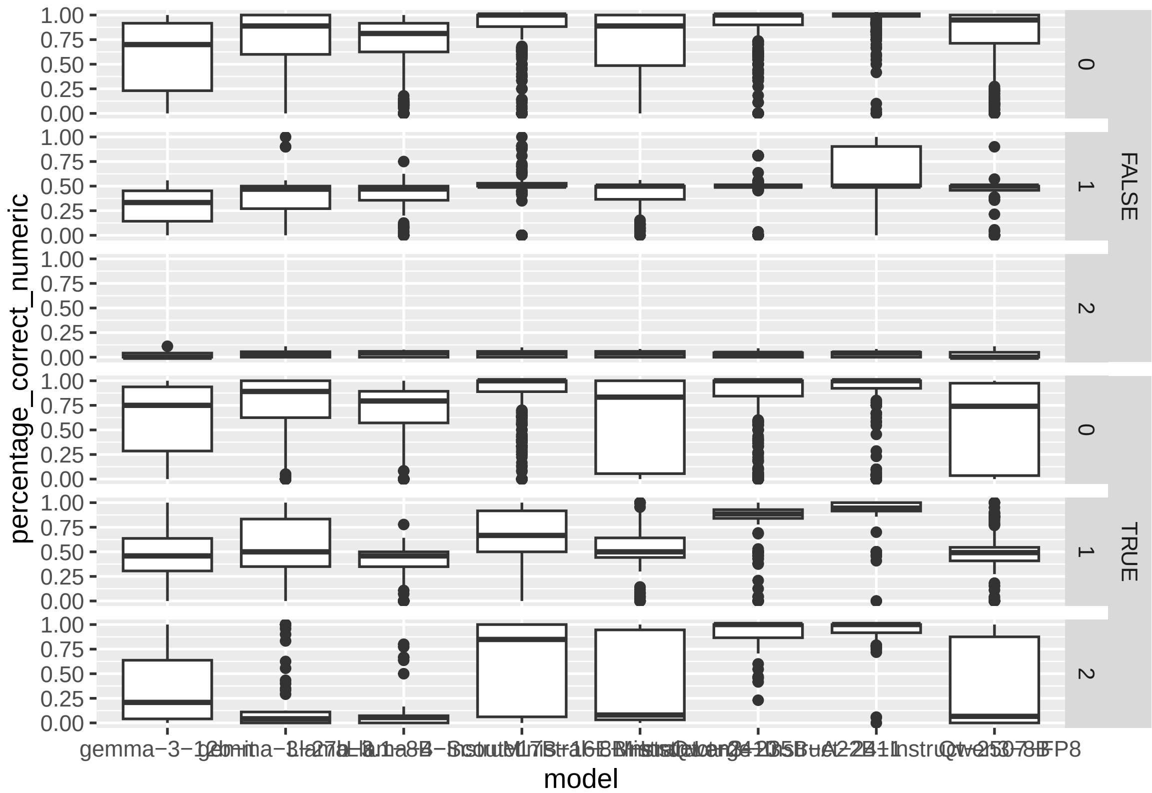


Figure B.5: Comparing the numeric prediction performance for the hybrid approach, based on the fact, if the LLM is prompted to respect currency units.

Chapter C

Appendix C - Error rate guidance report

The side research question asks, if it is possible to guide the users attention to predictions that have a higher empirical rate of errors. In this thesis we focus on the confidence score calculated from the reported log probabilities.

The formula for the confidence score is given in Equation (3.1).

C.1 Page identification

C.1.1 Binary classification

We investigate the relation between the reported confidence for an answer and its correctness, to check if it is possible to inform humans in the loop about results they should double check and which results they can trust. The LLM just returns one prediction and its confidence[1] for the binary classification task. We calculate its confidence as $confidence = exp(logprob)$, if the answer is *yes*. And we calculate its confidence as $confidence = 1 - exp(logprob)$, if the the answer is *no*,

Figure C.1 shows the distribution of reported confidence score for the binary classification with target type **Aktiva** for all table types grouped by their correctness for Ministral-8B-Instruct-2410. One can see that the predictions are very accurate making just 13 mistakes for 4981 predictions.

The reported confidence for answer *yes* is showing a wide spread from around 0.25 to 1.0. This is true for the answer *no* as well. Most wrong decisions are made for responses that have a reported confidence in the range from 0.25 to 0.75. But there are more correct answers in this range as well. It never misclassifies **GuV** or **Passiva**[2] as **Aktiva**. But it with shows some not recalled **Aktiva** tables and is predicting some of the pages of majority class, with not further described content and structure, as **Aktiva**.

This is different for models of most other model families. Figure C.2 shows, that Qwen2.5-32B-Instruct returns always high confidence scores, even when it is wrong. The model shows perfect recall but its precision is worse than the precision of the Mistral model.

Figure C.3 shows the precision-recall-curve for the best performing model twice for each target type. On the left plots the line color represents the threshold score one could use to decide when to accept a response as it is. On the right plots the line color is showing the F1 score that results with a chosen threshold.

The AUC value is lowest for **Aktiva**. Here the F1 score is highest for a threshold value of 0.73. This prevents to classify the pages of type *other* to get classified as **Aktiva**. If it is required to have a very high recall value a threshold of 0.44 should be chosen.

[1] The model could be forced to return multiple answers, but it was not. The confidence score is given as log probability. The exponential function was applied to show the results on the more common scale of 0 to 1.

[2] There was a single prediction where LLM predicts **Aktiva** with high confidence, when the truth is **Passiva** instead. Because Qwen was showing the same wrong prediction for one **Passiva** table, I double checked the ground truth. I found, that the page shows **Aktiva** and **Passiva** simultaneously and was not correct codified. This was not the only time, where a mistake in the gold truth was found, by examining potential LLM mistakes.

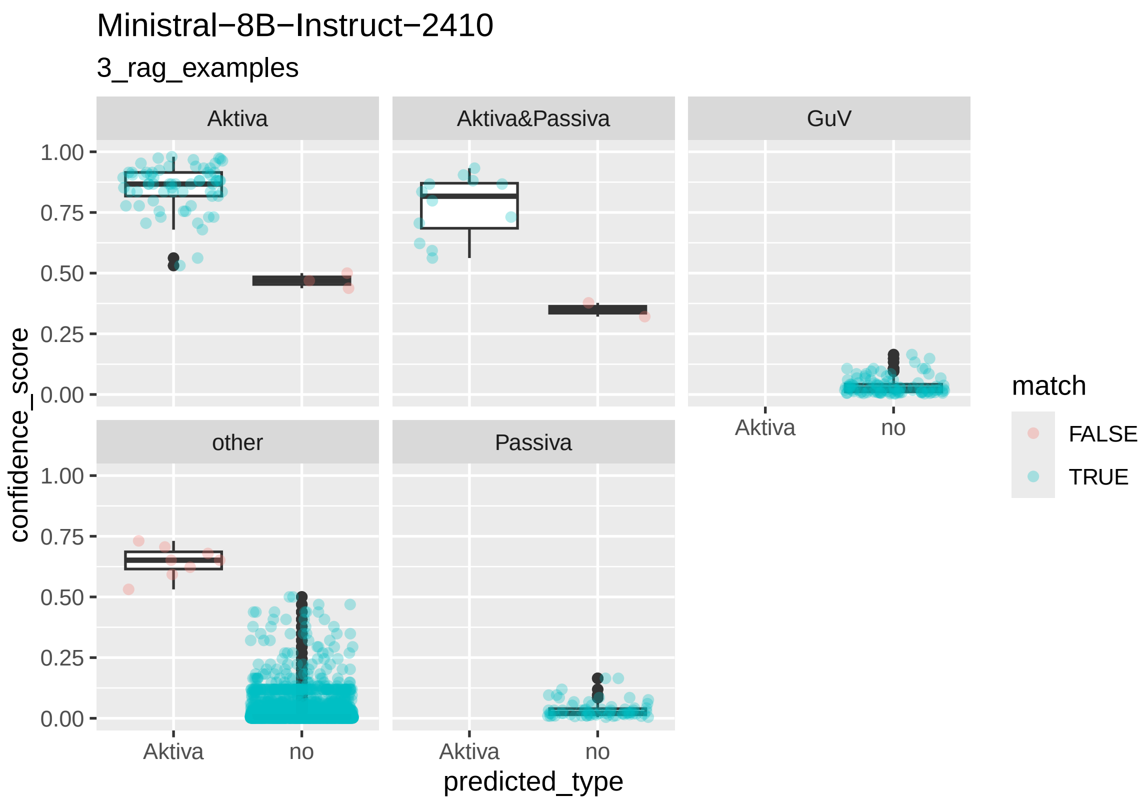


Figure C.1: Showing the confidence score for the Aktiva classification task grouped by table type and correctness for Mistral-8B-Instruct-2410.

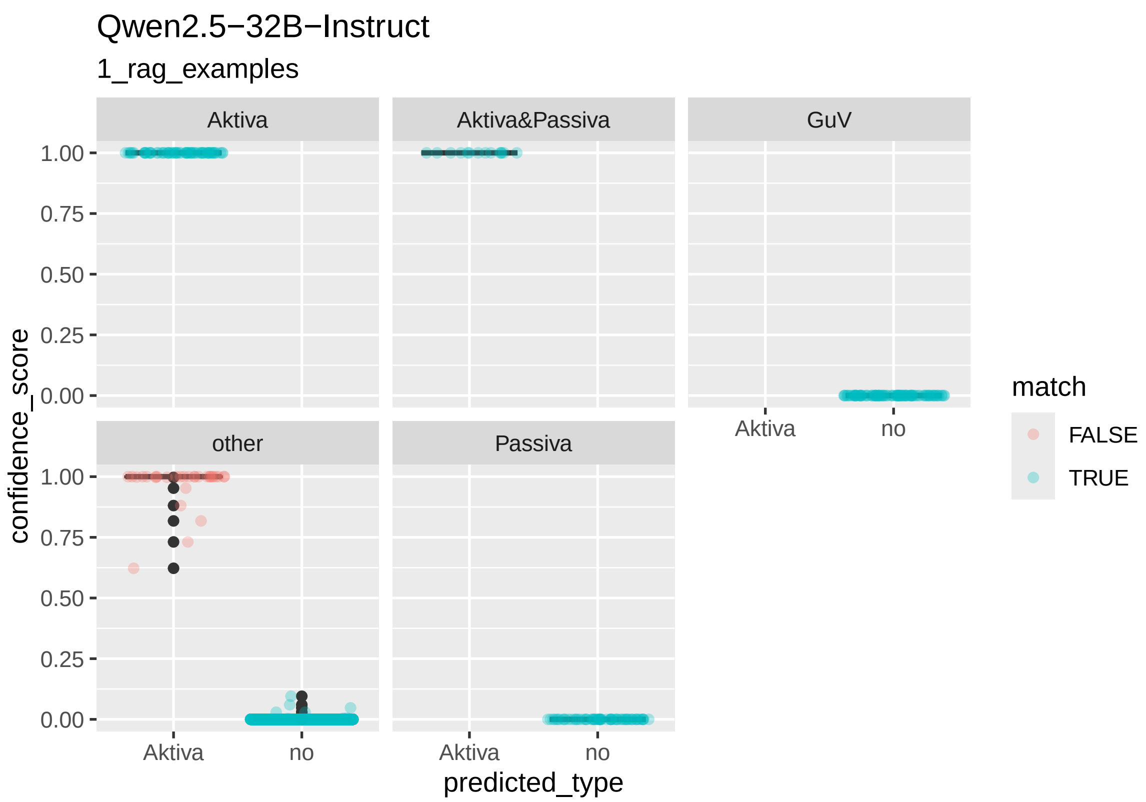


Figure C.2: Showing the confidence score for the Aktiva classification task grouped by table type and correctness for Qwen2.5-32B-Instruct.

The precision-recall-curve for **Passiva** is very similar but there is a step close to the recall value of 1.0. This has the effect that for a a guaranteed high recall a very low precision (0.24) and F1 (0.38) has to be accepted[3].

The shape of the precision-recall-curve for **GuV** almost perfectly reaches the top right corner. The highest F1 score is found with a threshold value of .56. With a threshold value of 0.5 a very high recall is guaranteed and the F1 score is just a little lower.

Figure C.4 summarizes the relation between reported confidence and correctness of the classification for all target types and compares it among the best performing model-strategy combinations for Ministral-8B-Instruct-2410 and Qwen3-8B. One can see, that the reported confidence for correct and incorrect classifications are separable in most cases for Mistral-8B. This separation is worse for Qwen3-8B and worst for target type **Passiva**.

Figure C.5 shows, that for Ministral-8B values with a confidence of 0.7 and more, a human don't has to double check the classification for target type **GuV**. This interval is smallest for **Passiva** where only confidences above 0.9 can be fully trusted. These empirical intervals might shrink, once more data is evaluated. If one is less strict and accepts misclassifiaction rates of 1 % the found interval for **Passiva** starts at 0.8 and is probably less depended on the sample evaluated. The percentage of predictions that can be trusted without risk is greater than 93 % even for target type **Passiva**.

For Qwen3-8B we find almost no range without any wrong classifications. For **GuV** this range includes 35 % of all predictions. The ranges that allow for 1 % of wrong classifications cover 57 % of all predictions at least.

Discussion:

- Could be more efficient to predict "is any of interest" and then which type, because dataset is highly imbalanced.
- Why takes n_rag_examples so much longer?
- **Aktiva** and **Passiva** sometimes on the same page and more similar than **GuV**?
- Recall = 1 for human in the loop (looking at selection of pages that could be target and none else, if the number of wrong pages are few => what says F1 with recall 1?)
- Confidence range to error rate

C.1.2 Multi-class classification

Confidence Figure C.6 shows the reported confidence scores for the predictions for the best performing model-strategy combination, Llama 4 Scout with *3_rag_examples*. It is confident for most correct predictions and only misclassifies some of the pages with unknown characteristics. The target types are all recognized correct. All confidences are greater than 0.5. Probably because there is no case where the confidences for all possible classes is below 0.5 and there always is a most probable class. It would have been interesting to use the classification framework of vLLM to get predictions for all competing classes. But this requires special trained models with pooling capability[4].

Figure C.7 shows the reported confidence scores for the predictions for the best performing model-strategy combination among the small models limited to *n_examples* with n smaller five[5], Ministral-8B-Instruct-2410 with *3_rag_examples*. One can see there are some wrong classifications for the minority classes as well. Especially, the **Passiva** target type is often classified as *other*. This is problematic for a smooth workflow (see discussion chapter?)

Figure C.8 shows the precision-recall-curve for Ministral-8B-Instruct-2410 with *3_rag_examples* twice for each target type. On the left plots the line color represents the threshold score one could use to decide when to accept a response as it is. On the right plots the line color is showing the F1 score that results with a chosen threshold.

[3] Thus, a human has in average to check four pages and select the correct **Passiva** page among them.

[4] It might be possible to request the n most probable answers to get confidence scores for all different predictions. But this was not investigated.

[5] The best performance results with *top_11_rag_examples* but the plot was less interesting and its F1 score was not listed in Table A.10.

C

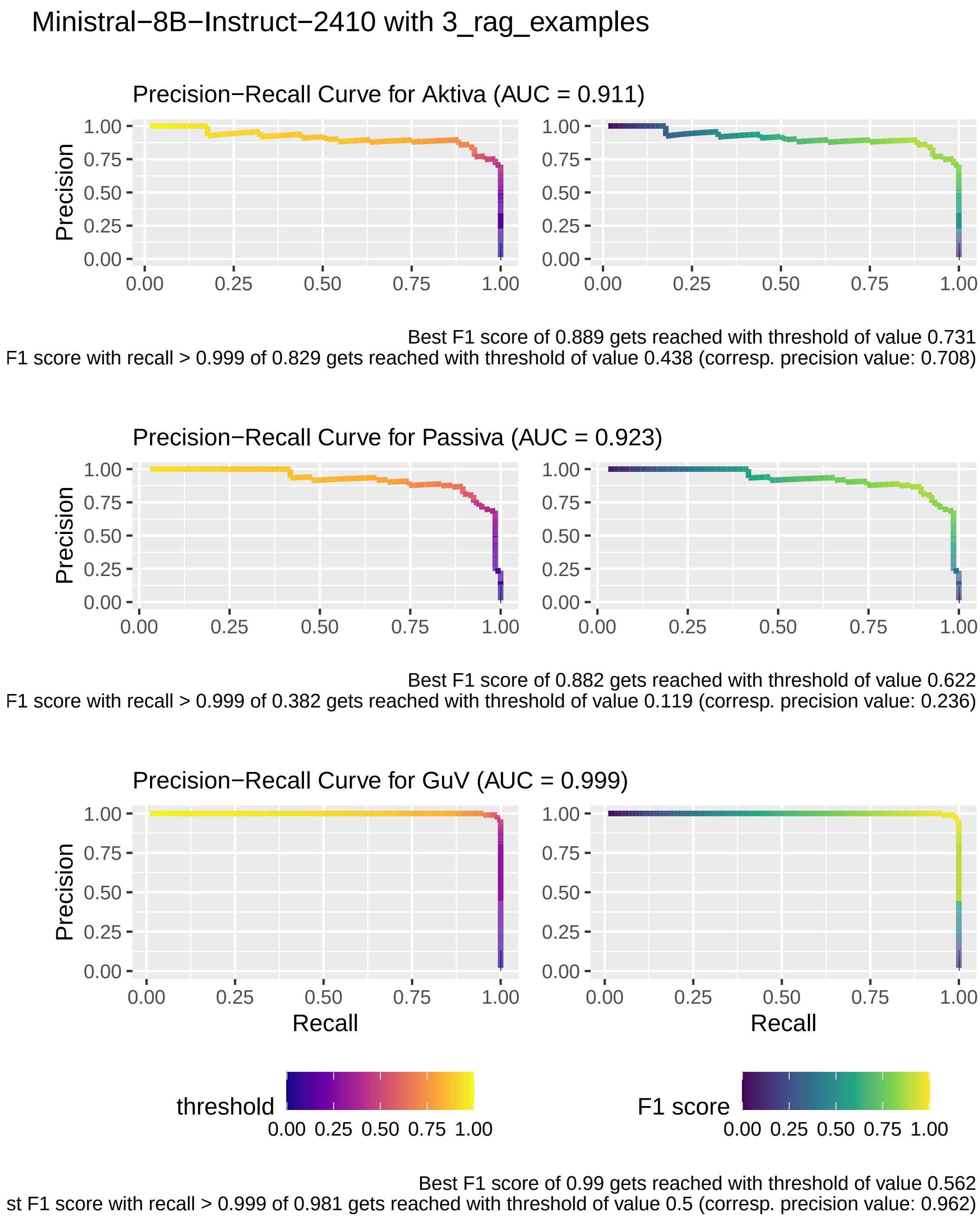


Figure C.3: Showing the precsion-recall-curve for the best performing model.

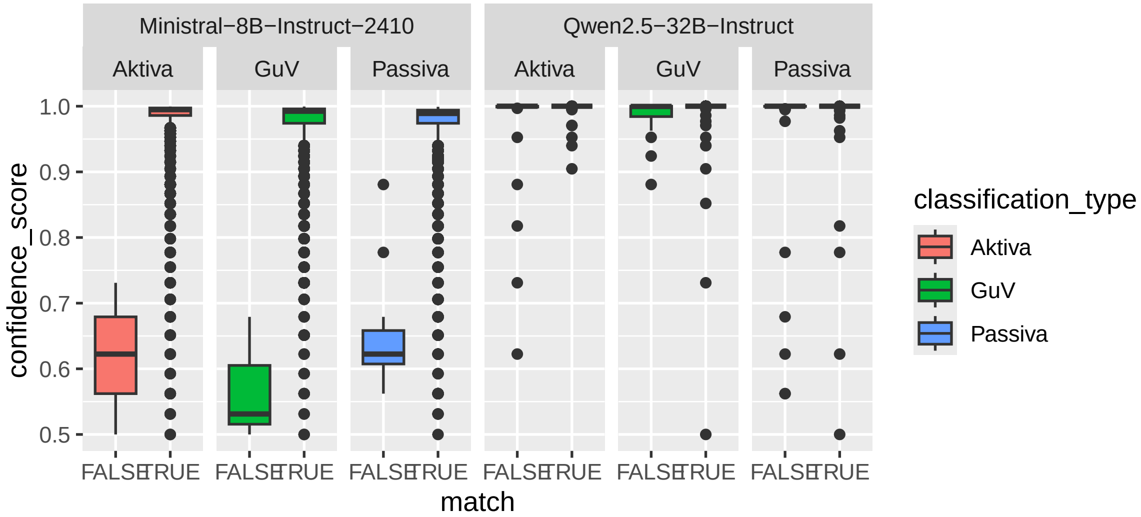


Figure C.4: Comparing the reported confidence scores for the page identification task for the Mistral and Qwen 3 with 8B parameters.

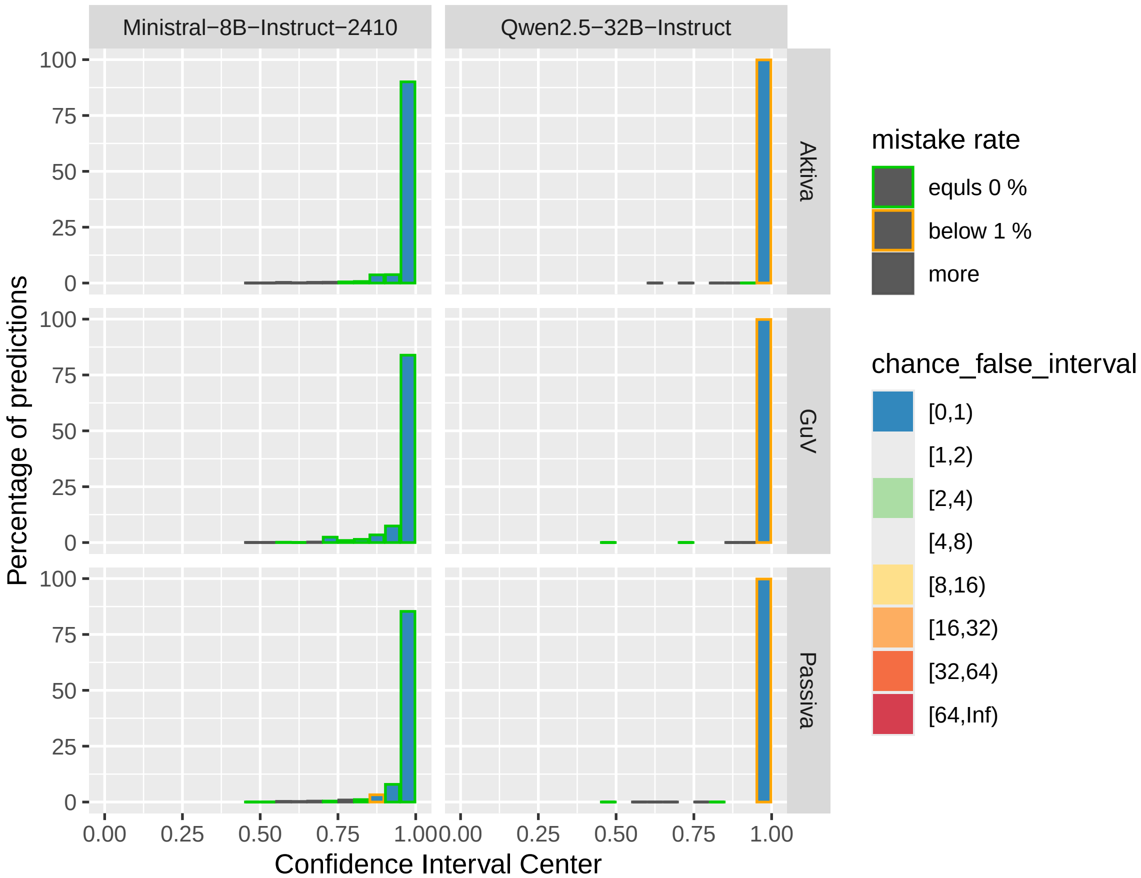


Figure C.5: Estimating the relative frequency to find a wrong classification over different confidence intervals

C

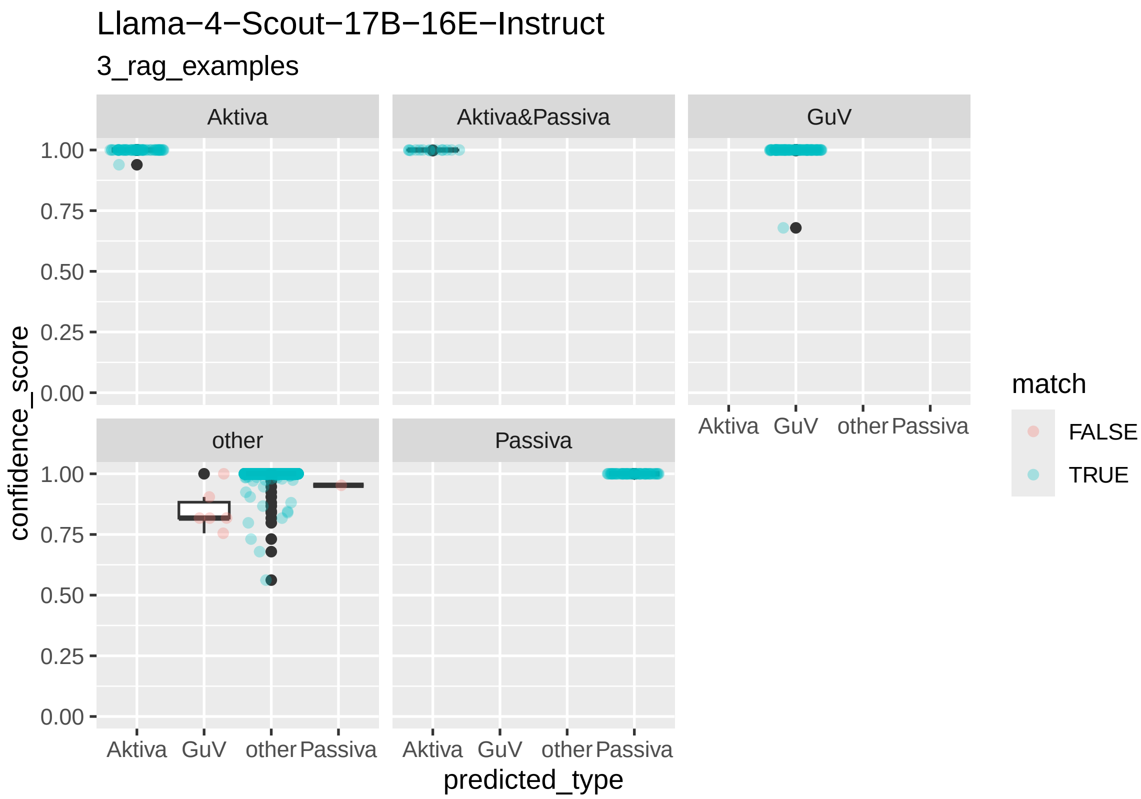


Figure C.6: Showing the reported confidence scores for all predictions of Llama 4 Scout grouped by the true target type. Errors have only been made within the majority class.

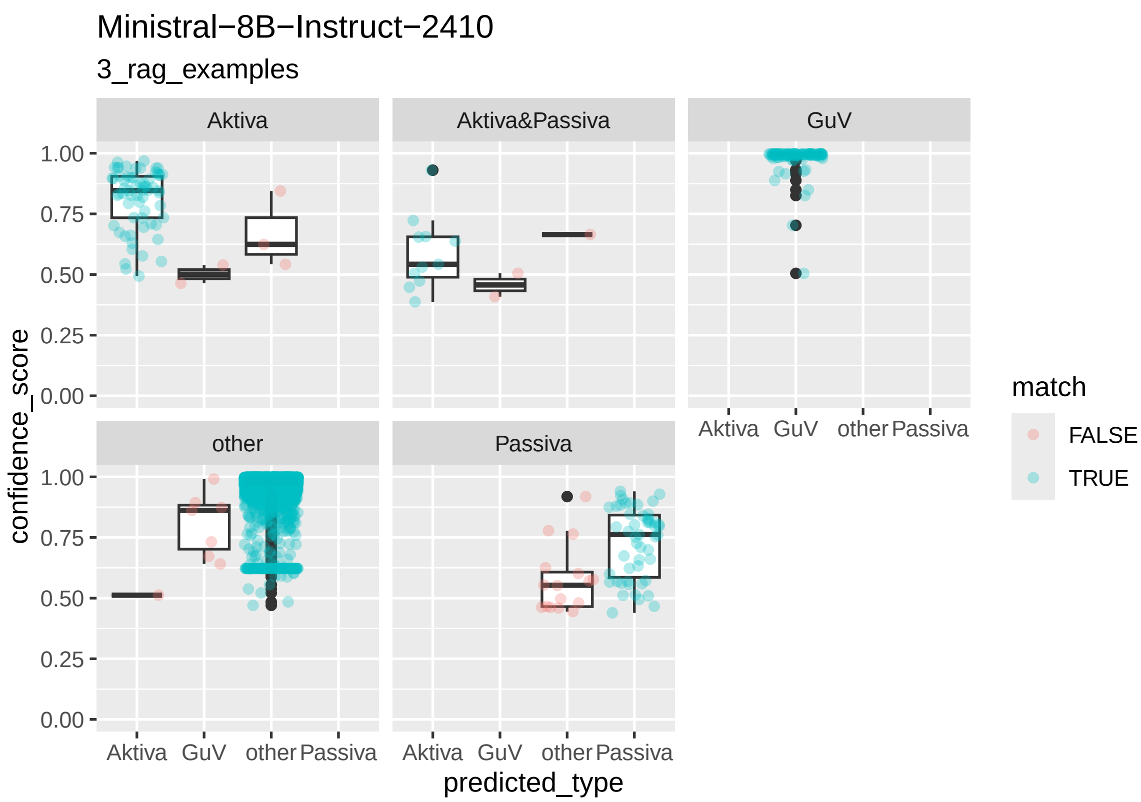


Figure C.7: Showing the reported confidence scores for all predictions of Ministral 8B grouped by the true target type. Errors have only been made within the majority class.

C

The AUC is highest for **GuV** again. But for the multi-class classification **Passiva** shows the lowest AUC, not **Aktiva** as it was in the binary classification task. The precision-recall-curve for **Aktiva** and **Passiva*+ show a "step" in the area of high recall. This has a strong effect on the threshold one should choose, if one wants to guarantee a high recall. The corresponding precision values of 0.2 and 0.13 mean that a human has to check five to eight pages in average to get a correct classified page of type **Aktiva** and **Passiva**.

The corresponding plot for the best performing model, Llama-4-Scout-17B-16E-Instruct, can be found in Figure G.8. Here the precision-recall-curve for **Passiva** and **GuV** is almost perfect. Just the single prediction for **Aktiva** wit a lower confidence shows an influence on the precision-recall-curve.

Figure C.9 summarizes the relation between reported confidence and correctness of the classification for all target types and compares it among the best performing model-strategy combinations for Llama-4-Scout-17B-16E-Instruct, Ministral-8B-Instruct-2410 and Qwen3-8B. It seems, as the reported confidence for correct and incorrect classifications are separable in most cases for Mistral-8B. For Llama 4 Scout this seems not true for the target type **GuV**. For Qwen3-8B there is almost no separation at all.

Figure C.10 shows, that there is almost no area, where the empirical rate of wrong classifications is zero[6]. Only for Ministral-8B we find intervals, where a human don't has to double check the classification for target types **Aktiva** and **GuV**. These intervals include 90 % of all predictions. If error rates of 1 % are accepted almost all predictions by Llama Scout 4 and about 96 % of the predictions by Ministral-8B are included in the corresponding intervals. For Qwen3-8B we find no interval without an error rate below 1 %.

C.2 Extraction with LLMs

C.2.1 Real tables

Confidence Figure C.11 shows, that the distrubution of the models reported confidence is heterogeneous. Again, Qwen3-8B reports very high confidence values no matter if the results are correct or not. Qwen3-235B-A22B-Instruct (and Qwen3-14B) report some lower confidence scores for predictions, where they are wrong. The Mistral model again reports a wider range of confidences and for wrong results the reporetd confidence is lower. But no real separation can be observed for any of the models.

Figure C.12 helps answering the question, if the reported confidence score of the responses can be used, to alert a human that certain predictions might be wrong. In contrast to the page identification task, we find no confidence intervals where the mistake rate is equal 0 or less than 1 %. The majority of the predictions has a very high reported confidence. For the best performing model Qwen3-235B-A22B-Instruct we find error rates of 3.3 % for numeric predictions and 1.3 % for predicting a missing value.

Thus, we can inform the human about the empirical found error rates but do not flag some values to be realy trustworthy. In defense for the model: with manual transcription the error rate is not lower. But we can inform the human about values that have shown a higher rate of mistakes, especially for the Ministral model.

C

C.2.2 Synthetic tables

Confidence Figure C.13 shows, that we do not find a high confidence interval containing a majority of the predictions with 0 % error rate. But for Qwen3-235B we find, that the error rate is below 1 %, except for predicting numeric values, while ignoring their currency units.

Figure G.26 groups the responses additonally by the *input_format* of the documents. It shows, that with HTML documents Qwen3-235B achieves 0 % error rate for the prediction of missing values and predicting numeric values, if currency units get respected.

C.2.3 Hybrid approach

Confidence Figure C.14 shows the rate of wrong predictions for given confidence intervals. Again, the confidence for predicting a missing value is higher than for predicting a numeric value. One can't see much

[6] The size of intervals has been narrowed down to 0.1 % and still there was no range without wrong classification for Llama 4 Scout.

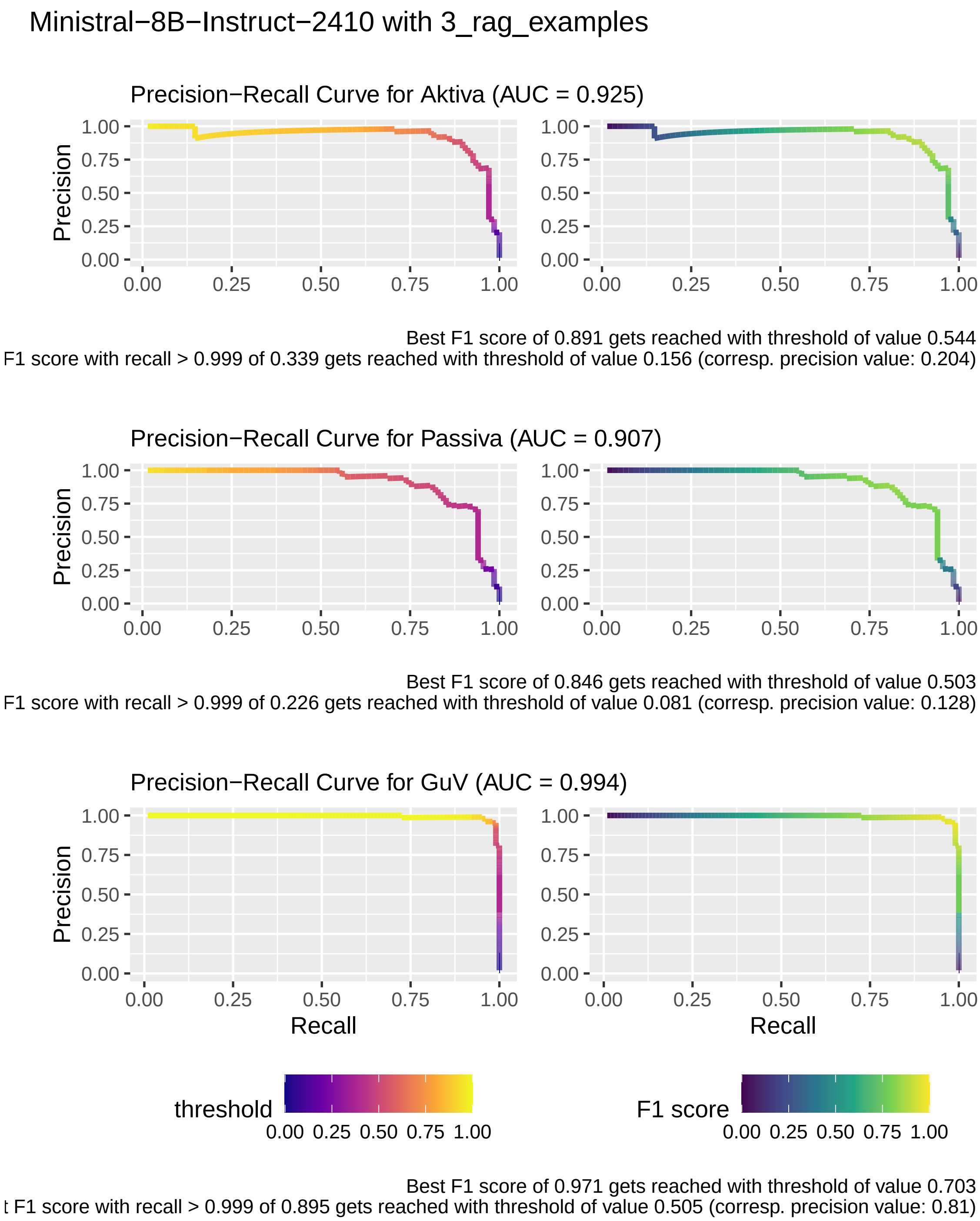


Figure C.8: Showing the precsion-recall-curve for Ministral-8B-Instruct-2410.

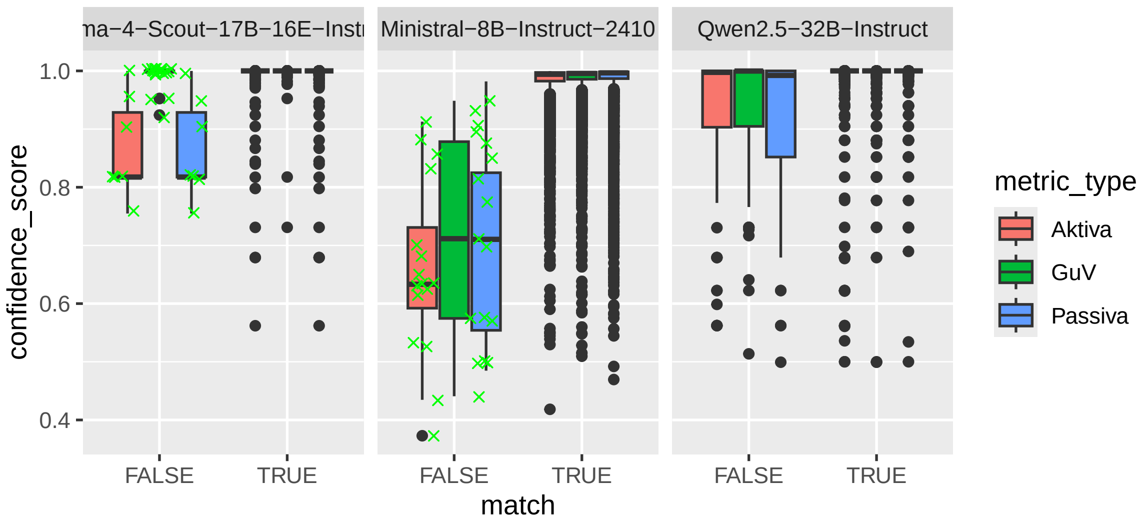


Figure C.9: Comparing the reported confidence scores for the multi-class page identification task for the Mistral and Qwen 3 with 8B parameters. Showing individual scores for groups with less than 20 observations.

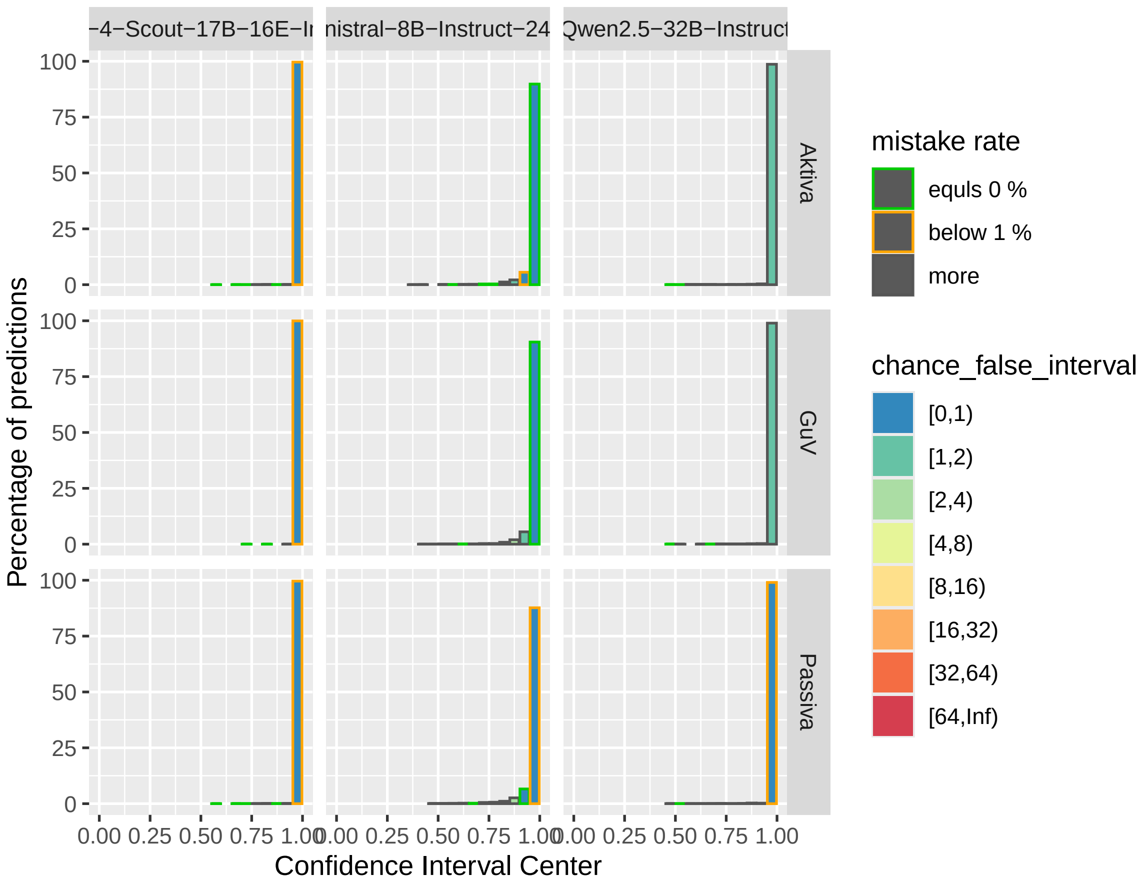


Figure C.10: Estimating the relative frequency to find a wrong classification over different confidence intervals for the multi-class classification task.

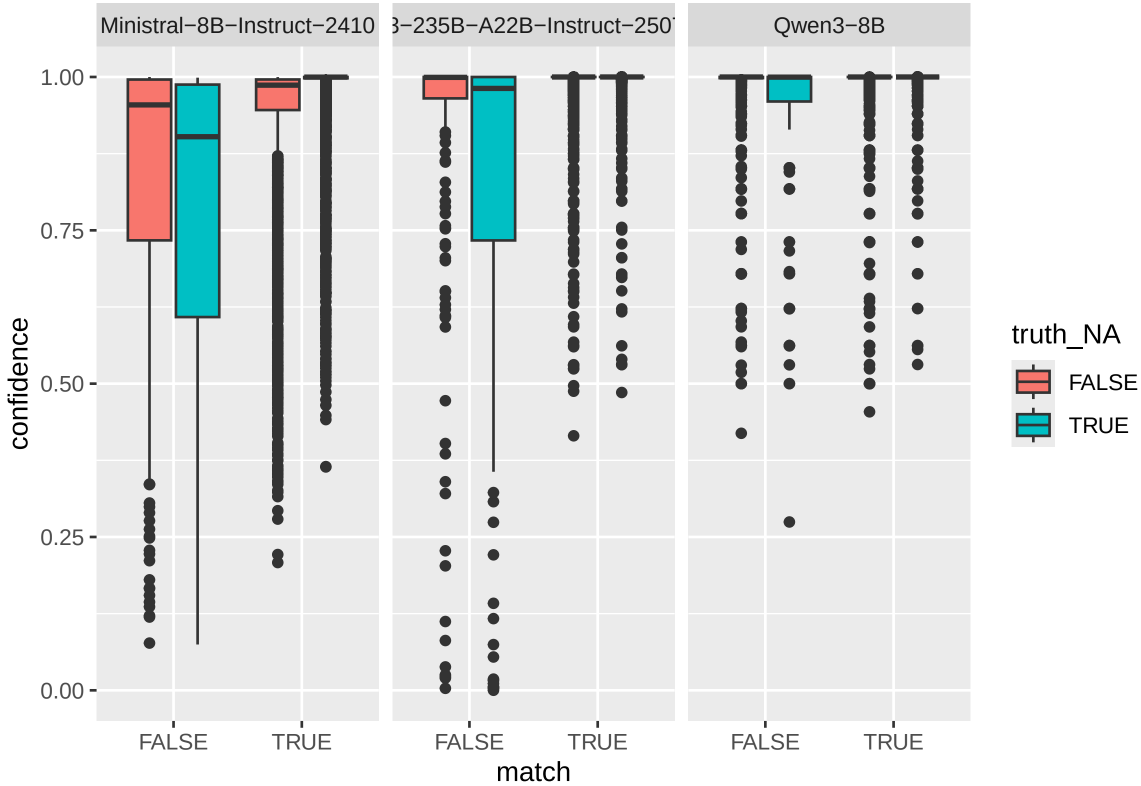


Figure C.11: Comparing the reported confidence scores for the table extraction task on real dataset for the Mistral and Qwen 3 with 8B parameters.

difference, but for the best performing model Qwen3-235B the error rate for numeric values is lower, when currency units are respected (20 % vs 26 %). But the error rate is still to high to mark any numeric value as trustful.

C

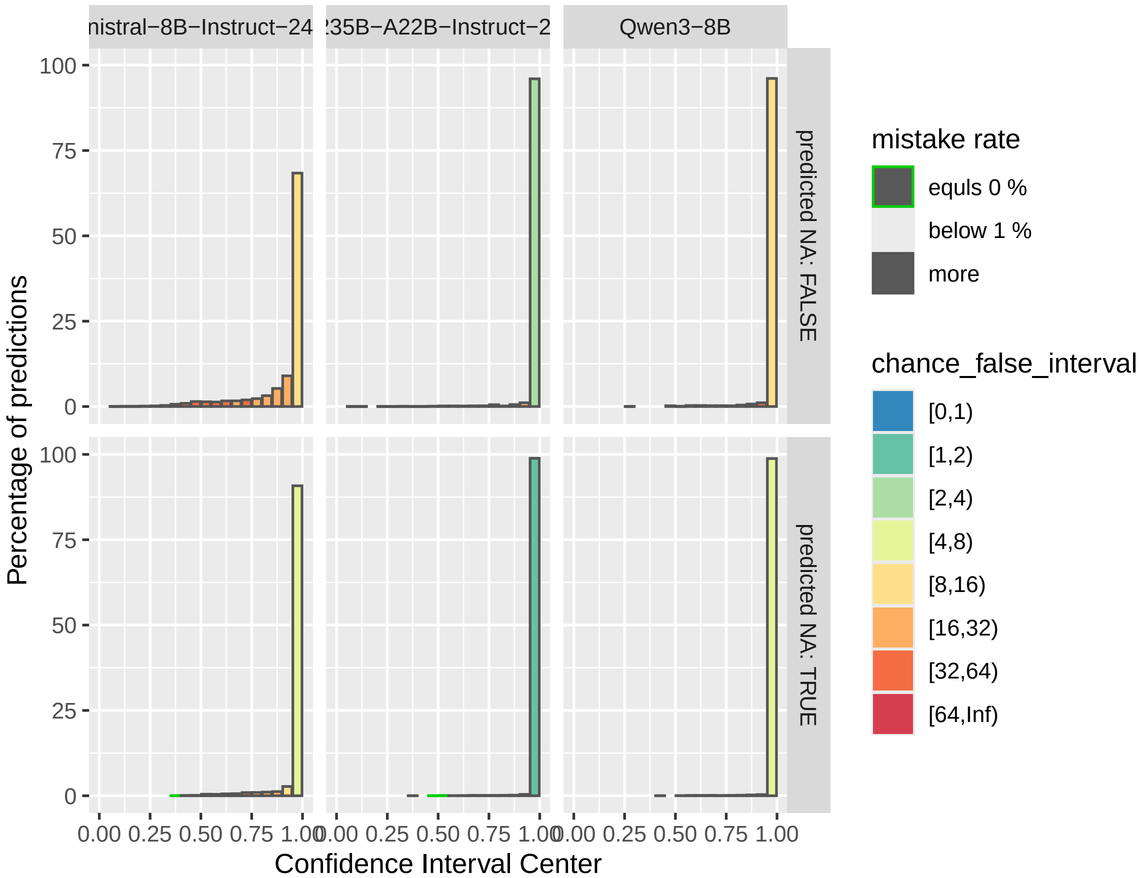


Figure C.12: Estimating the relative frequency to find a wrong classification over different confidence intervals.

C

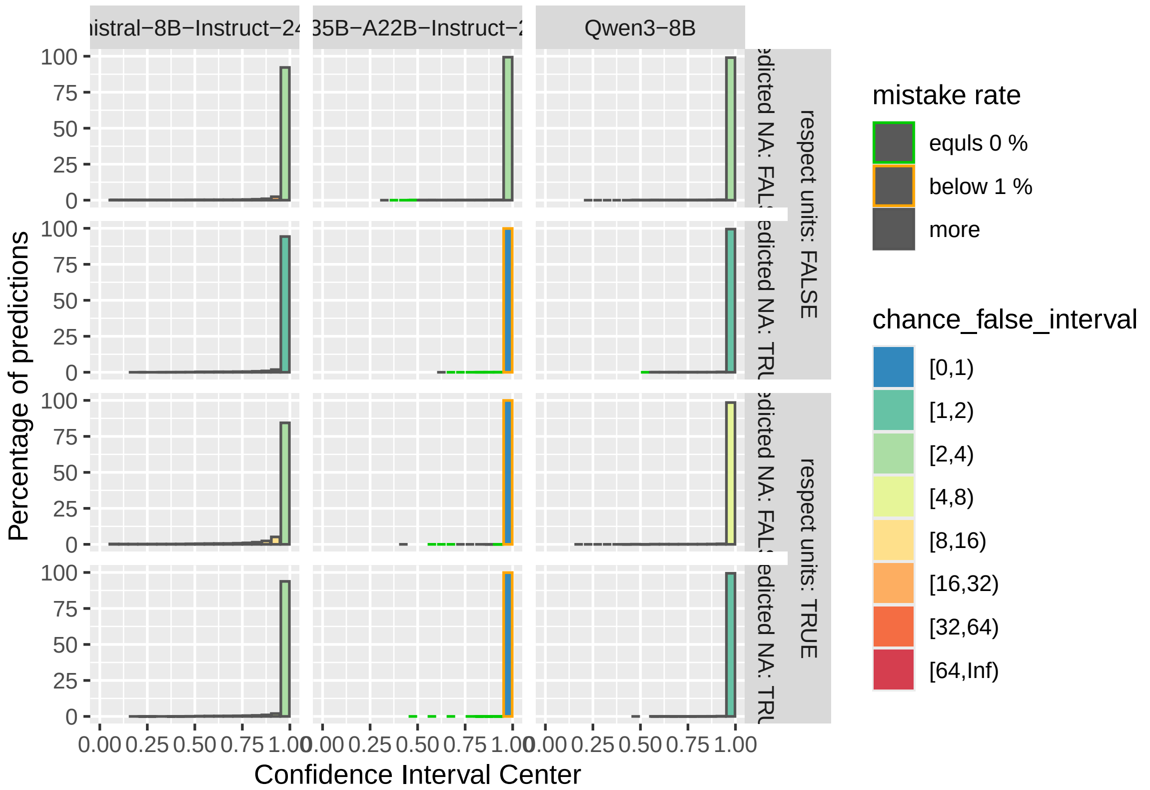


Figure C.13: Estimating the relative frequency to find a wrong extraction result over different confidence intervals for predictions for the synthetic table extraction task.

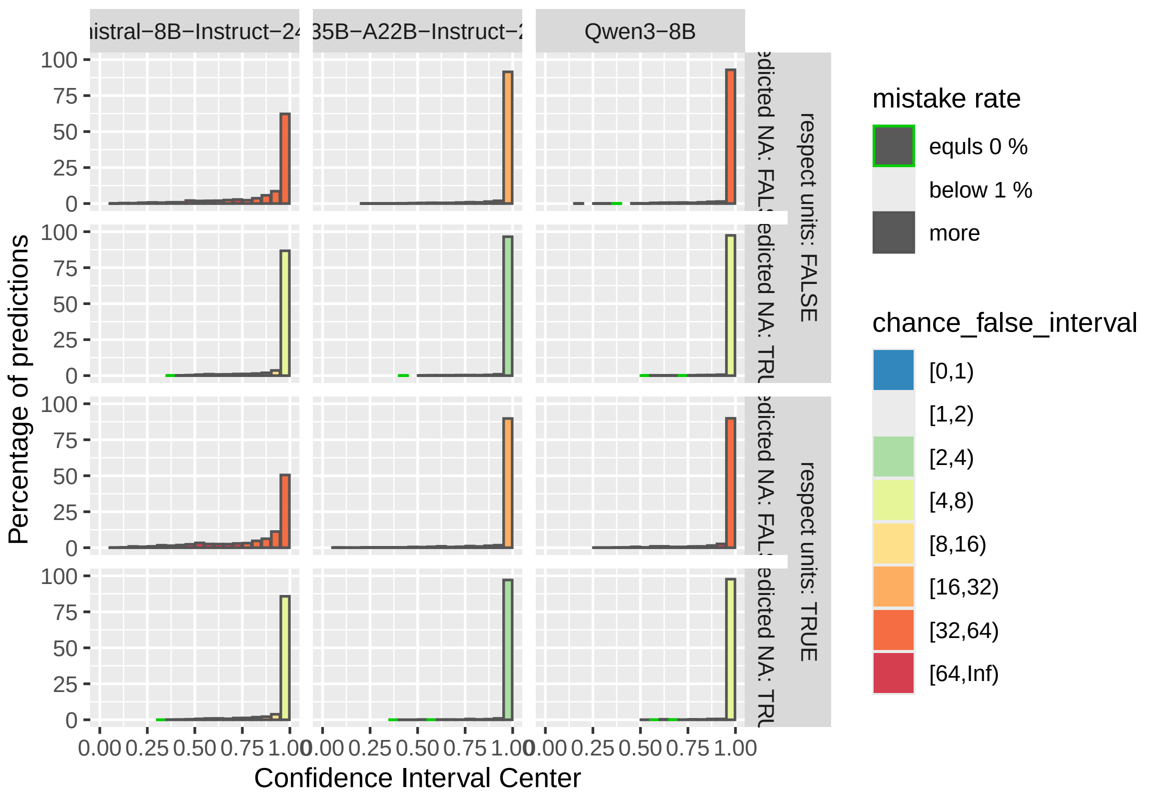


Figure C.14: Estimating the relative frequency to find a wrong extraction result over different confidence intervals for predictions based on synthetic examples for in-context learning.

C

Chapter D

Appendix D - Feature effect analysis

D.1 Regular expressions

Hypotheses The formulated hypotheses have been evaluated visually using the dependence and beeswarm plots from the *shapviz* library based on the SHAP values calculated for a random forest.

Real dataset The formulated hypotheses have been evaluated visually using the dependence and beeswarm plots from the shapviz library based on the SHAP values calculated with a random forest.

Table D.1 shows in the first column the predictors included in the random forests. Subsequent groups of two columns show the hypotheses and the found effects of those predictors for two aggregated measures (F1 score and percentage of correct numeric predictions) and one value based measure (binomial correctness rating).

Predictors that are marked with an asterisk only have five or less representatives. Thus, those results are not reliable. Bold set hypotheses show the predictors, that showed the highest mean SHAP values. For all measures but the binomial this means the effect is at least 0.025. For the binomial measure the effect of a predictor with bold hypothesis is at least 0.05. Results with red text highlight hypotheses that are not supported by the visual evaluation.

Table D.1 shows many red results, meaning the corresponding hypothesis is getting no support. Since most of these findings show only minor effect strength we don't interpret them as strongly challenging those hypotheses. Only three findings regarding the F1 score show a strong effect and findings that do not align with our hypotheses. First, it seems, that finding a sum in the same row, has a negative effect of finding any valid number there. Second, it has negative effect, if the previous year column is given as *T€*. Third, it has negative effect, if the columns are visually separated.

see Figure G.10

Synthetic dataset Table D.2 shows, many red results as well, meaning the corresponding hypothesis is getting no support. We find more predictors with a strong effect compared to the real **Aktiva** table extraction task. The results are based on 24_576 extracted tables and the SHAP values have been calculated on 2_000 examples each.

Contrary to our assumption, does the *extraction_backend* have a strong effect on all measures. We find, that *pdfium* is struggling with some of the table characteristics while *pymupdf* is not influenced by them. Figure D.1 A shows this exemplary for the characteristic *header_span*. An example for a erroneous text extraction with *pdfium* can be found in section @ref(#regex-extraction-mistakes). Actually, all results that are marked with an asterisk are showing this effect if *pdfium* is used as extraction backend. This can be inspected in Figure G.12.

Furthermore, does the number of columns is have a positive effect overall. Figure D.1 B shows, that this effect has inverse direction for the two libraries.

Table D.1: Comparing the formulated hypotheses and the found results for the table extraction on real Aktiva tables with the regular expression approach.

	F1		% correct numeric		binomial	
predictor	Hypothesis	Result	Hypothesis	Result	Hypothesis	Result
extraction_backend	neutral	neutral	neutral	pymupdf better	neutral	neutral
n_columns	neutral	2 is better	neutral	neutral	neutral	2 is better
sum_same_line	**neutral**	**negative**	**negative**	negative	negative	neutral
sum_in_header*	**neutral**	positive	neutral		neutral	neutral
header_span	neutral	negative	neutral	negative	neutral	negative
unit_first_cell*	neutral		negative	neutral	negative	neutral
T_in_previous_year	**neutral**	**negative**	**negative**	negative	negative	negative
T_in_year*	neutral	negative	**negative**	negative	negative	negative
passiva_same_page	negative	positive	negative	positive	negative	neutral
vorjahr	neutral	negative	neutral	negative	neutral	negative
vis_separated_cols	**neutral**	**negative**	neutral	negative	neutral	negative
vis_separated_rows	neutral	neutral	neutral	neutral	neutral	neutral
label_length					negative	negative
label					**unknown**	
missing					**positive**	positive

It might be worth noting, that the row for *Anteile an verbundenen Unternehmen* was rated to have a clear negative effect on the chance to extract the correct value.

The question, if visual separation of columns is having an effect, as found for the real data, is not studied here, because in the synthetic tables all columns are visually separated. But this could be investigated in future work. It is possible, that the visual separation is causing the faulty text extractions of *pdfium*.

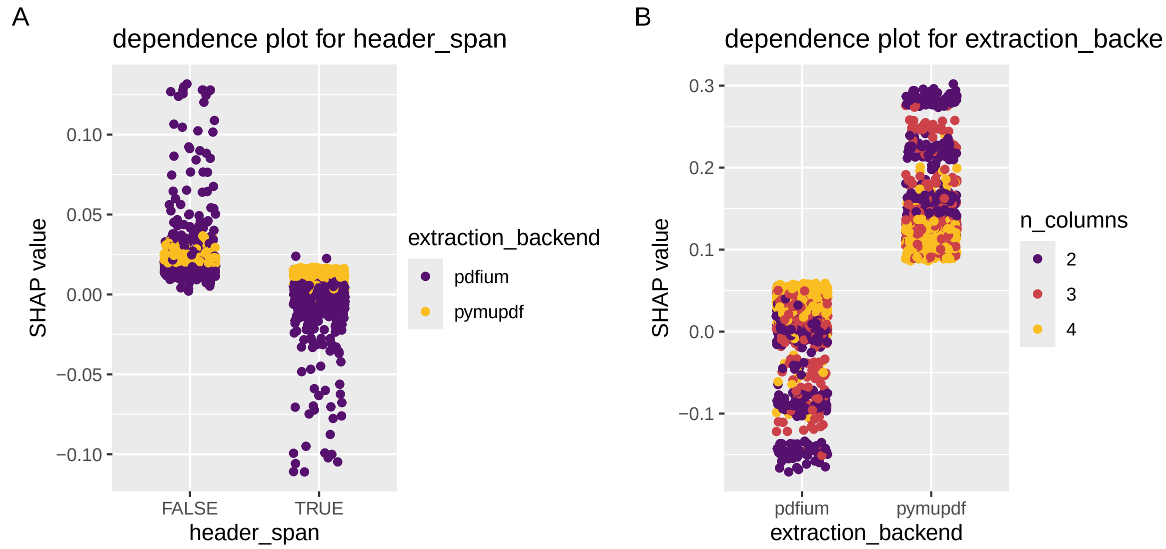


Figure D.1: Showing the influence of the extraxtion library on the numeric text extraction task with synthetic data for the percentage of correct numeric predictions.

D.2 Real tables

Hypotheses The formulated hypotheses have been evaluated visually using the dependence and beeswarm plots from the shapviz library based on the SHAP values calculated with a random forest.

Table D.2: Comparing the formulated hypotheses and the found results for the table extraction on synthetic Aktiva tables with the regular expression approach.

	F1		% correct numeric		binomial	
predictor	Hypothesis	Result	Hypothesis	Result	Hypothesis	Result
extraction_backend	**neutral**	pymupdf better	**neutral**	pymupdf better	**neutral**	pymupdf better
n_columns	**neutral**	**positive**	**neutral**	**positive**	**neutral**	**positive**
sum_same_line	neutral	neutral	negative	negative*	negative	neutral
header_span	neutral	negative*	neutral	negative*	neutral	negative*
thin	negative		neutral	positive*	neutral	neutral
year_as	**neutral**	positive*	neutral	positive*	neutral	positive*
unit_in_first_cell	negative	negative*	negative	negative*	negative	negative*
log10_unit_multiplier	**neutral**	negative*	**positive**	negative*	positive	negative*
enumeration	**positive**	positive*	**neutral**	positive*	**neutral**	positive*
shuffle_rows	neutral	neutral	neutral	neutral	neutral	neutral
text_around	neutral	neutral	neutral	neutral	neutral	neutral
many_line_breaks	negative	neutral	neutral	neutral	neutral	neutral
label_length					negative	neutral
label					**unknown**	
missing					positive	positive

Table D.3 shows in the first column the predictors included in the random forests. Subsequent groups of two columns show the hypotheses and the found effects of those predictors for two aggregated measures (F1 score and percentage of correct numeric predictions) and two value based measures (binomial correctness rating and reported confidence for the prediction).

Predictors that are marked with an asterix only have five or less representatives. Thus, those results are not reliable. Bold set hypotheses show the predictors, that showed the highest mean SHAP values. For all measures but the binomial this means the effect is at least 0.025. For the binomial measure the effect of a predictor with bold hypothesis is at least 0.05. Results with red text highlight hypotheses that are not supported by the visual evaluation.

For most measures the model and method related predictors (*model_family*, *parameter_count*, *method_family* and *n_examples*) show the strongest effects. Worth mentioning is, that *method_family* and *n_examples* show no strong effect on the reported confidence score. From the table related characteristics most strong effects show the hypothesized direction. Not predicted was the negative effect of label length on the reported confidence score. The visual separation of columns and rows shows small effects. We find no support for a negative effect of the fact that the **Passiva** table is on the same page as the **Aktiva** table.

see Figure G.22

D.3 Synthetic tables

Hypotheses Table D.4 shows some unsupported hypotheses for predictors with strong effects. The observations suggest, that prompting the model to respect the currency units is decreasing its performance to predict the correct numeric values. This is understandable, if the task is reflected properly and we made a mistake there, when we formulated our hypothesis.

If the model is not prompted to respect the currency units, it is presented with examples that just copy over the numeric values. And it gets evaluated if it copied the values correctly. If the model is prompted to respect the currency units, it is presented with examples, where the values not only get copied but also transformed. And they get evaluated if they do the transformation correct as well. Thus the task is harder, if numeric values should be respected and the effect is having a negative direction.

Figure D.2 shows that the transformation task is handled best, if the examples are provided with the *top_n_rag* strategy. It does not work with the *zero_shot* strategy. It also shows, that the performance is lower

Table D.3: Comparing the formulated hypotheses and the found results for the table extraction on real Aktiva tables the LLM approach.

	F1		% correc	
predictor	Hypothesis	Result	Hypothesis	Res
model_family	**unknown**	google worst	**unknown**	goo
parameter_count	**positive**	positive	**positive**	pos
method_family	**top_n_rag & n_random best**	zero shot worst	**top_n_rag & n_random best**	top
n_examples	**positive**	1 and 3 best (five bad for Llama4)	**positive**	1 ar
n_columns	neutral	neutral	neutral	neu
sum_same_lin	neutral	negative (i.e. if prev year not T€)	negative	neg
sum_in_header*	neutral	neutral	neutral	neu
header_span	neutral	neutral	neutral	neu
unit_first_cell*	neutral	neutral	neutral	neu
T_in_previous_year	neutral	neutral	**negative**	neg
T_in_year*	neutral	negative	**negative**	neg
passiva_same_page	negative	neutral	negative	neu
vorjahr	neutral	neutral	neutral	neu
vis_separated_cols	neutral	negative	neutral	neg
vis_separated_rows	neutral	neutral	neutral	pos
label_length				
label				
missing				
confidence				

with the PDF *input_format* and that the models have difficulties, if the *unit_multiplier* is one million. This also shows a strong effect and is strongest for the PDF *input_format*. This is a general effect. We see in C.13 that it can be different for single models like Qwen3-235B.

old:

HTML and Markdown better but expected interaction effects mostly not found - except: - columns help pdf - thinning least bad for pdf - pdf worst with numbers that have currency units (short numbers, maybe no 1000er delimiter) - enumeration positive for pdf (and interaction with log10 mult)

line breaks are no problem

zero shot gets confused by text around

Markdown might be even better than HTML

respecting units was bad - except for: Top n rag finds examples with same currency units (shorter numbers more important than currnency in header?)

log10 multiplier has many interaction effects

LLama 4 Maverick again problem with five examples

Positive column count effect (different for real data)

header span not reflected in html and markdown

D

D.4 Hybrid approach

Hypotheses Table D.5 shows only one unsupported hypothesis for a predictor with a strong effect: *method_family*. Figure D.3 shows the dependence plot for the this predictor. The prompting strategy *static_example* shows the highest SHAP values. This is surprising, because the the *static example* is equivalent to providing a single random example.

Table D.4: Comparing the formulated hypotheses and the found results for the table extraction on synthetic Aktiva tables with the LLM approach.

	F1		
predictor	Hypothesis	Result	Hypothesis
model_family	**unknown**	google worst	**unknown**
parameter_count	**positive**	positive	**positive**
method_family	**top_n_rag & n_random best**	zero shot worst	**top_n_rag**
n_examples	**positive**	positive (except for Llama 4 Maverick)	**positive**
n_columns	3 is worse	positive	neutral
n_columns:input_format	less for html and md	neutral	less for htm
sum_same_line	neutral	neutral	negative
sum_same_line:input_format	neutral	neutral	better for h
header_span	neutral	neutral	neutral
header_span:input_format	Can't be evaluated		Can't be ev
header_span:respect_units	neutral	neutral	negative
thin	Can't be evaluated		neutral
respect_units	neutral	negative	**positive**
respect_units:input_format	neutral	neutral	better for h
input_format	**md and html better**	md and html better	**neutral**
year_as	neutral	neutral	neutral
unit_in_first_cell	neutral	neutral	negative
unit_in_first_cell:input_format	neutral	neutral	neutral
log10_unit_multiplier	neutral	neutral	**positive**
log10_unit_multiplier:input_format	neutral	negative for pdf	neutral
enumeration	positive	neutral	neutral
shuffle_rows	neutral	neutral	neutral
text_around	neutral	neutral	neutral
many_line_breaks	negative	neutral	neutral
many_line_breaks:input_format	better for html and md	neutral	neutral
label_length			
label			
missing			
confidence			

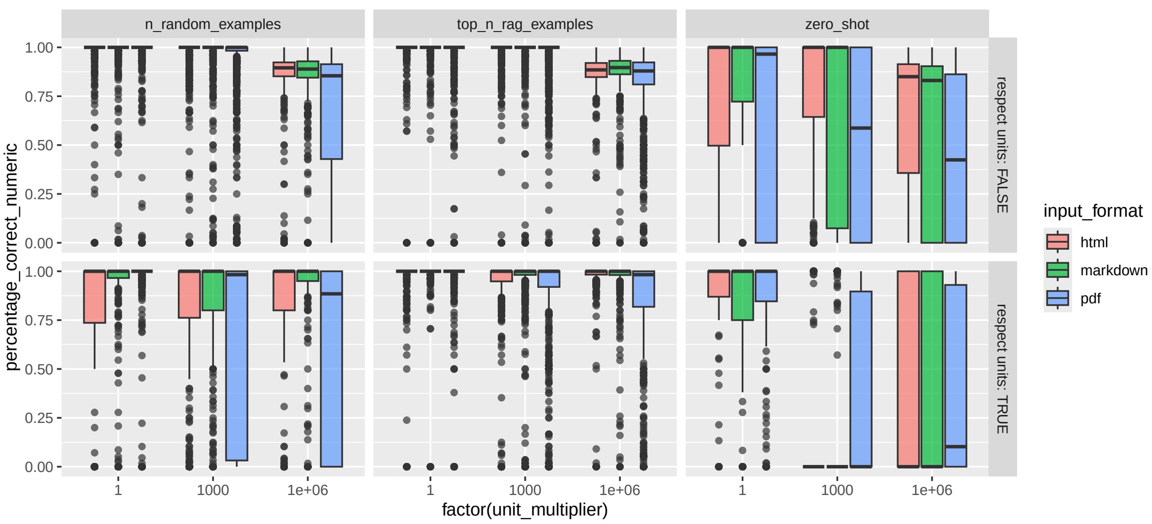


Figure D.2: Comparing the percentage of correct extracted numeric values grouped by input format, method family and the fact, if currency should be respected.

Table D.5: Comparing the formulated hypotheses and the found results for the table extraction on real Aktiva tables with the hybrid LLM approach.

	F1		% correc
predictor	Hypothesis	Result	Hypothesis
model_family	**unknown**	Google & Qwen3 worst	**unknown**
parameter_count	**positive**	positive	**positive**
method_family	**top_n_rag & n_random best**	static_example best	**top_n_rag & n_random best**
n_examples	**positive**	positive	positive
n_columns	neutral	interaction with passiva_same_page	neutral
sum_same_lin	neutral	negative if header_span	negative
sum_in_header*	neutral	neutral	neutral
header_span	neutral	neutral	neutral
unit_first_cell*	neutral	neutral	neutral
T_in_previous_year	neutral	neutral	**negative**
T_in_year*	neutral	negative	**negative**
passiva_same_page	negative	neutral	negative
vorjahr	neutral	neutral	neutral
vis_separated_cols	neutral	negative (if T_in_prev year)	neutral
vis_separated_rows	neutral	positive (if header_span)	neutral
respect_units	neutral	neutral	positive
label_length			
label			
missing			
confidence			

Figure D.3: Estimating the relative frequency to find a wrong extraction result over different confidence intervals for predictions based on synthetic examples for in-context learning.

Chapter E

Appendix E - Miscellaneous

E.1 Human in the loop application

Using LLMs has become a lot easier in the past years. *Python* frameworks as vLLM make it easy to deploy open-source models for developers. *Open WebUI* allows us to set up a sophisticated chat bot UI with ease. *Dify* promises to create no-code AI agents via a graphical flowchart. But the regular employee at RHvB will just use a provided application. And to solve the task, to extract (tabular) information for loading it into a relational database, a chat UI seems not the perfect match.

For the planned application we have two requirements: First, it should be able to accurately extract information. Second, it has to provide a good user experience.

Extracting the information of the assets table accurately means:

- Numeric values have to be extracted and transformed according to potentially given currency units, e.g. *T€*.
- The row labels should be matched with a limited set of labels in the target database.
- Unknown row labels and their values have to treated in a appropriate way.

To ensure, that there are no mistakes, a HITL approach should be implemented. Figure E.1 shows that the employee should initiate the extraction, by providing the document to extract information from and potentially choose, what to extract. The employee should also double check the results, before they are saved for future usage in down stream tasks.

Why do we aim to build an application that assists the process instead of fully automating it? On one hand, building a system that makes no mistakes might be impossible. And it would be more efficient, to ensure machine-readable data provision in the first place.

On the other hand, building a HITL application, could increase the employees acceptance for the product and trust in the results. An assisting application is not as threatening as an application, that fully automates a process, making the human redundant and potentially leading to job cuts. Programms building on the machine learning paradigm are rarely making decisions that are not seen as correct by a human. And humans make errors as well. Teaming up with an AI to reduce the error rate, should increase the trust in the results, if this story is sold right. (change management)

We believe, that the UX would improve, if we can guide the user, which values to check and which he can trust. This would reduce the work load and might increase the feeling to spend the time meaningful, by finding the same amount of mistakes, while checking less values. Thus, we formulate our side research question:

Q4 Can we use additional information from the extraction process, to guide the user which values need to be checked and which can be trusted as they are?

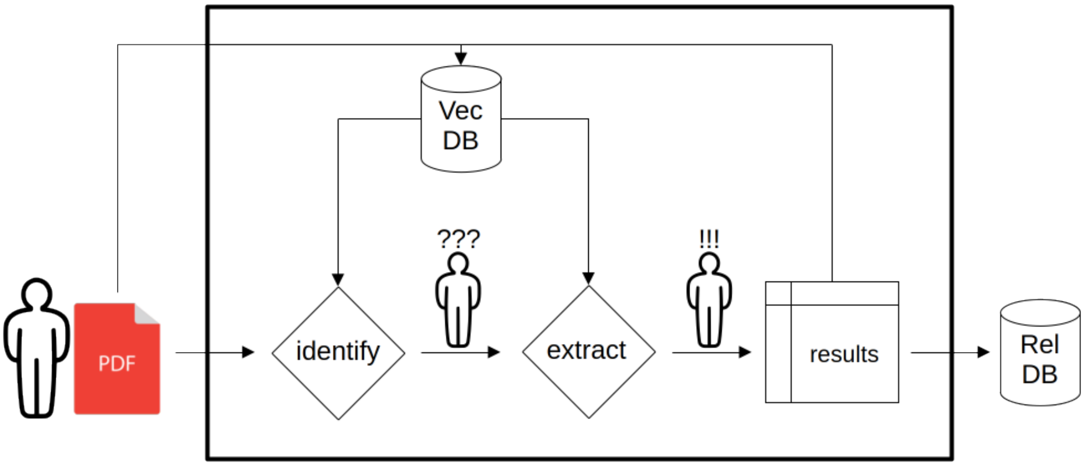


Figure E.1: Showing the information extraction process in a HITL application. The fed in document and results are saved in a database, that is used for the in-context learning RAG approaches for future extractions. The results are saved in the relational company database as well, e.g. as information to present in dashboards.

Of course, mistakes not only happen by extracting a wrong value, but by matching the values row with the wrong row labels from the predefined set. This can happen, because they are similar but not equal or because the row label at hand is not present in the predefined set yet. Having a human in the loop, seems to be a good way to handle such unknown row labels as well.

The UI (user interface) should assist in all cases. Comparing numeric values, checking the row label matching and showing, which rows have not been handled yet and need human decisions. Figure E.2 shows a first mockup, about what has to be compared focussing on a single years numeric values and the row lable matching. A UX study should answer questions, how exactly the UI should look like and if the employee prefers to check the identified page, before the extraction begins (see the human shape with question marks in Figure E.1).

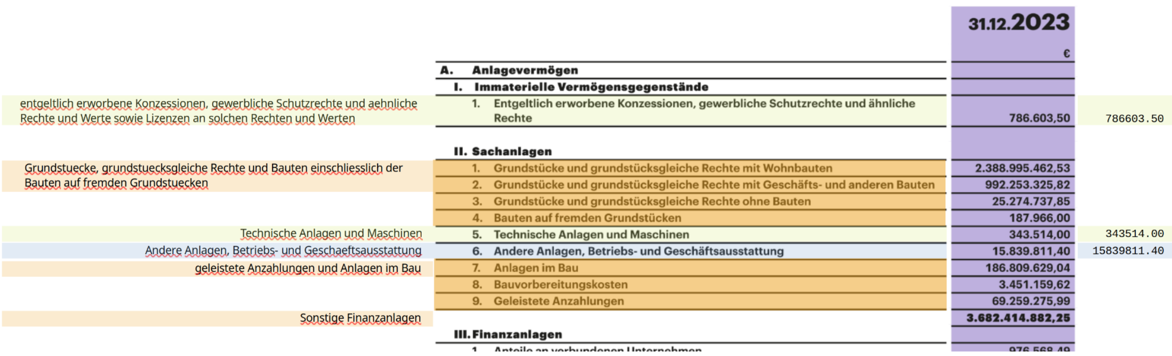


Figure E.2: Showing the information that need to be compared by the user after the information extraction. Unmatched rows could be highlighted in another color.

human in the loop (Mosqueira-Rey et al., 2023; Natarajan et al., 2024; Wu et al., 2022)

- allowing in place adjustments to the extracted data.

E.2 Local machine

One can find the specifications of the local machine used to run the tasks that do not require a GPU below. It is a lightweight laptop device. Its performance cores support hyper-threading and have a clock range between 2.1 and 4.7 GHz. Due to its slim design, there is little active cooling. Thus, thermal throttling starts quickly. It is a reasonable assumption that most local benchmarks are running at 2.1 GHz. Despite this handicap, it has a sufficiently large RAM of 32 GB and 3 TB of NVMe disk space.

System Details Report

Report details

- **Date generated:** 2025-07-19 13:56:16

Hardware Information:

- **Hardware Model:** LG Electronics 17ZB90Q-G.AD79G
- **Memory:** 32.0 GiB
- **Processor:** 12th Gen Intel® Core™ i7-1260P × 16
- **Graphics:** Intel® Graphics (ADL GT2)
- **Disk Capacity:** 3.0 TB

Software Information:

- **Firmware Version:** A2ZG0150 X64
- **OS Name:** Ubuntu 24.04.2 LTS
- **OS Build:** (null)
- **OS Type:** 64-bit
- **GNOME Version:** 46
- **Windowing System:** Wayland
- **Kernel Version:** Linux 6.11.0-29-generic

E.3 Benchmarks

E.3.1 Text extraction

All our experiments use the text extracted from PDF files. The available open-source libraries differ in their speed, quality of results and restrictiveness of licensing (Auer et al., 2024). We have tested multiple libraries in this thesis, because Auer et al. (2024) published no quantitative results. The benchmark runs on the local machine described in section E.2. There are 5256 pages to extract the text from.

Table E.1 shows, that *pdfium* and *pymupdf* extract the text fastest. For implementation in a system where the text has to get extracted live or frequently the speed of the library might be paramount. Since the AGPL license of *pymupdf* might not be met with the application, that will be created for RHvB, *pdfium* is an interesting candidate for the PDF parsing library to use.

Auer et al. (2024) reports, that *pdfium* occasionally merges text cells that are not close o each other, resulting in unrecoverable quality issues. Thus we checked some of the extracted texts manually and include the PDF extraction backend as a variable in our experiments. The page identification experiment, using the regex approach, shows no effect of the text extraction library. In contrast, we find an effect in the later performed information extraction experiment on synthetic **Aktiva** tables with the regex approach.

Some examples for errorneous extracted texts with *pdfium* and *pdfminer* can be found in section E.7.

time to ocr 107 images with pdf2image and pytesseract: 1:28+11:41

Table E.1: Comparing extraction time (in seconds) for different Python package

package	runtime in s
pdfium	**14**
pymupdf	22
pypdf	218
pdfplumber	675
pdfminer	752
docling-parse	1621

E.3.2 Table detection

E.3.2.1 old classification with llm

First experiments for the page identification task ran on a smaller dataset. The pages used for this setup are the pages, that are classified as target class by the regular expression approach. Without batch processing and the vLLM framework classifying these 727 pages already takes very long. This motivated the test, if image detection models can be used for the page refinement by detecting tables. Subsection E.3.2.2 shows the results of this attempt.

benchmark and n-shot base for llm classification (contrasts)

E.3.2.2 yolo benchmark and table transformer

We test three visual models, trained for table detection. One is based on Microsofts table transformer. The other two are based on Ultralytics Yolo 12 and differ in their parameter size:

- microsoft/table-transformer-detection
- yolo12l-doclaynet
- yolo12n-doclaynet

The Yolo models performed much better. Up to a threshold of 0.5 they show a recall of around 1 and a precision over 0.95. Thus, they can be used to refine the page range, by identifying all pages that have a table. The table transformer model has a worse precision of around 0.7

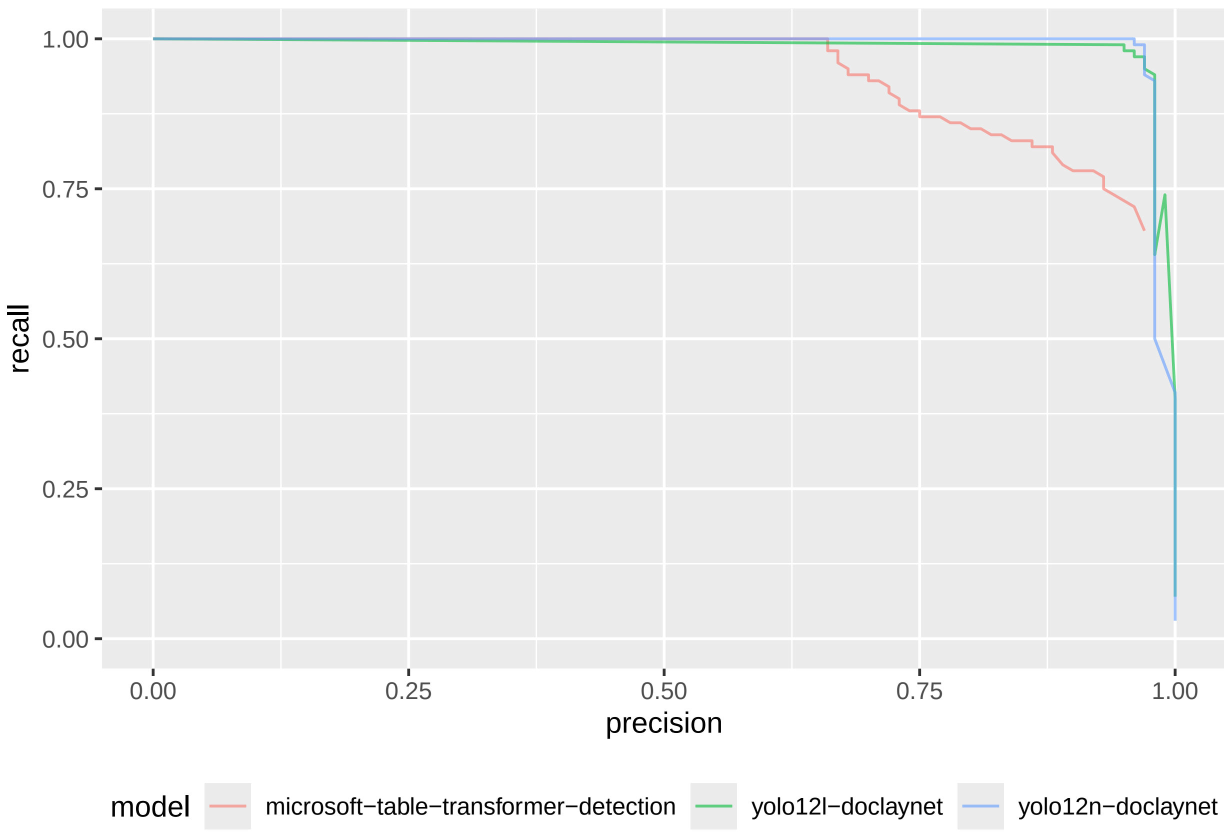
microsoft 5:55 min yolo large 20:6 min yolo nano 3:20 min

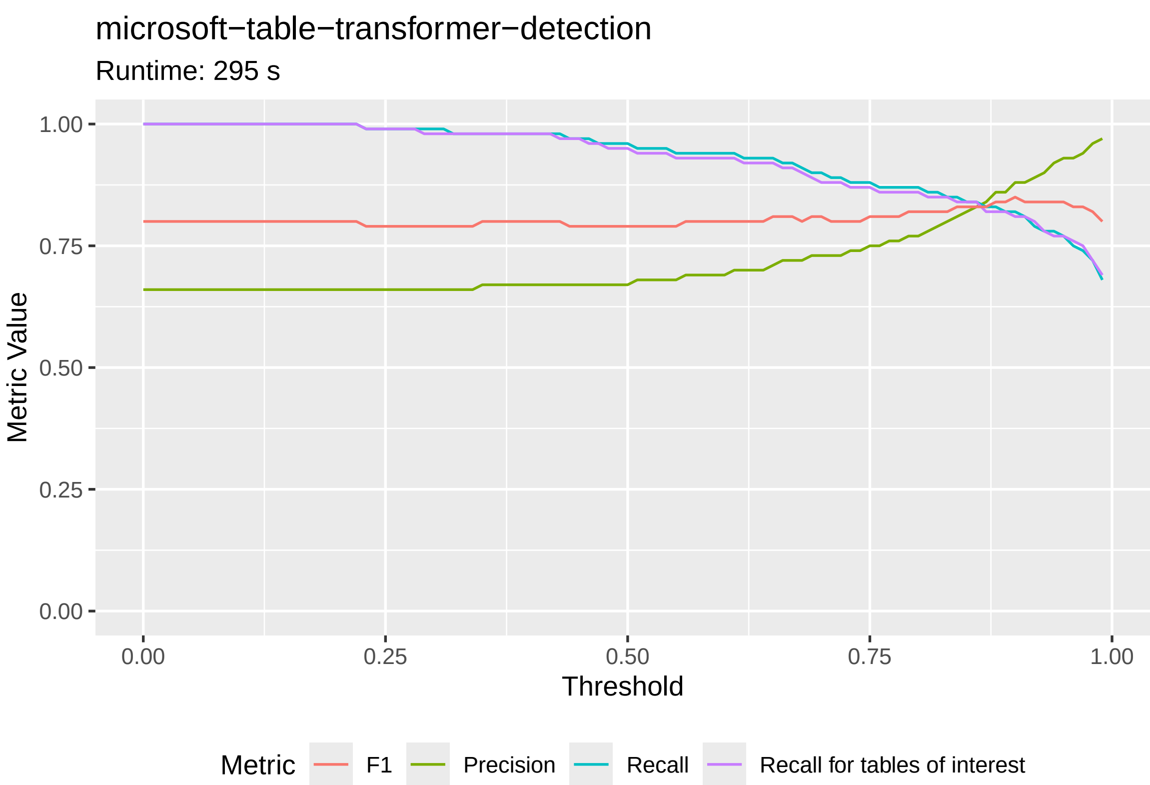
```
df_table_detection_result %>%
  select(model, runtime, pr_auc) %>%
  mutate(pr_auc = format_floats(pr_auc, 3))
```

```
## # A tibble: 3 x 3
##   model                                runtime pr_auc
##   <chr>                                <chr>   <chr>
## 1 microsoft-table-transformer-detection 5:55    0.920
## 2 yolo12l-doclaynet                     20:6    0.986
## 3 yolo12n-doclaynet                     3:20    0.989
```

```
table_detection_result_list %>% rowwise() %>%
  mutate(metrics = list(bind_rows(metrics, c(precision = 0, recall = 1)))) %>%
  unnest(metrics) %>%
  ggplot() +
  geom_line(aes(x = precision, y = recall, color = model), alpha = 0.6) +
  theme(
    legend.position = "bottom"
  )
```

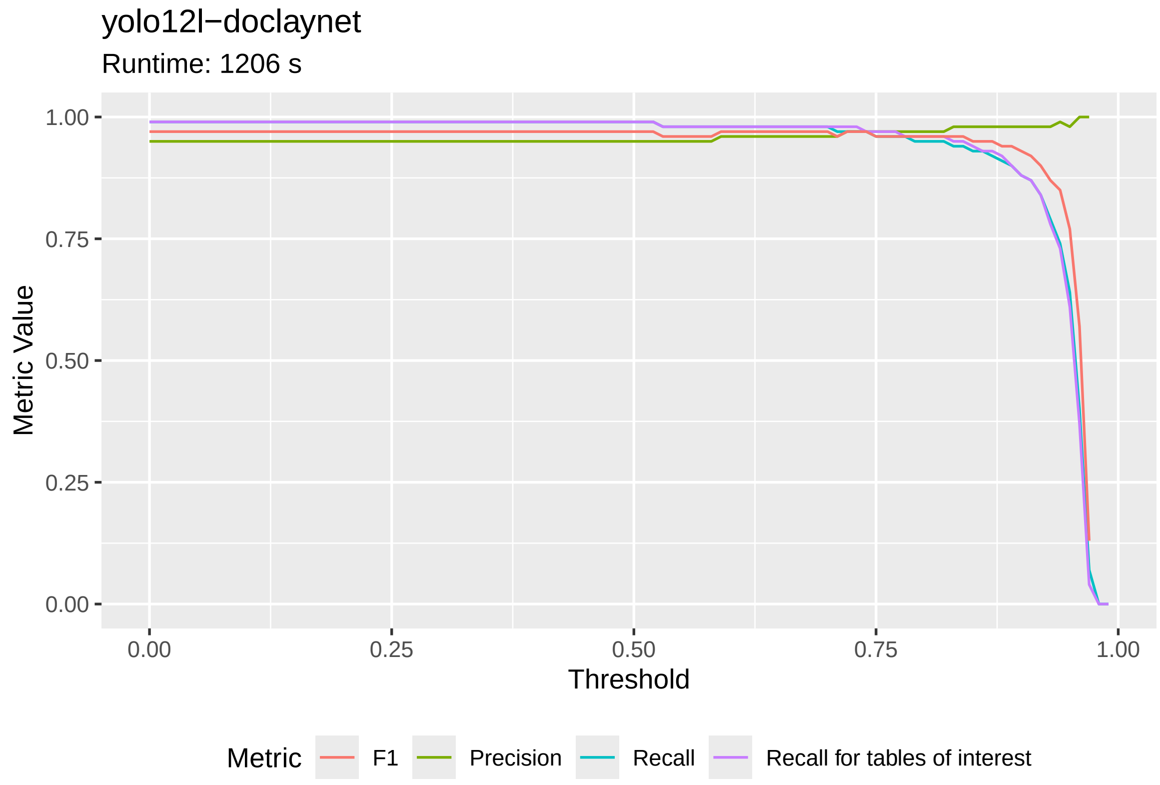
```
## Warning: Removed 5 rows containing missing values or values
## outside the scale range (`geom_line()`).
```



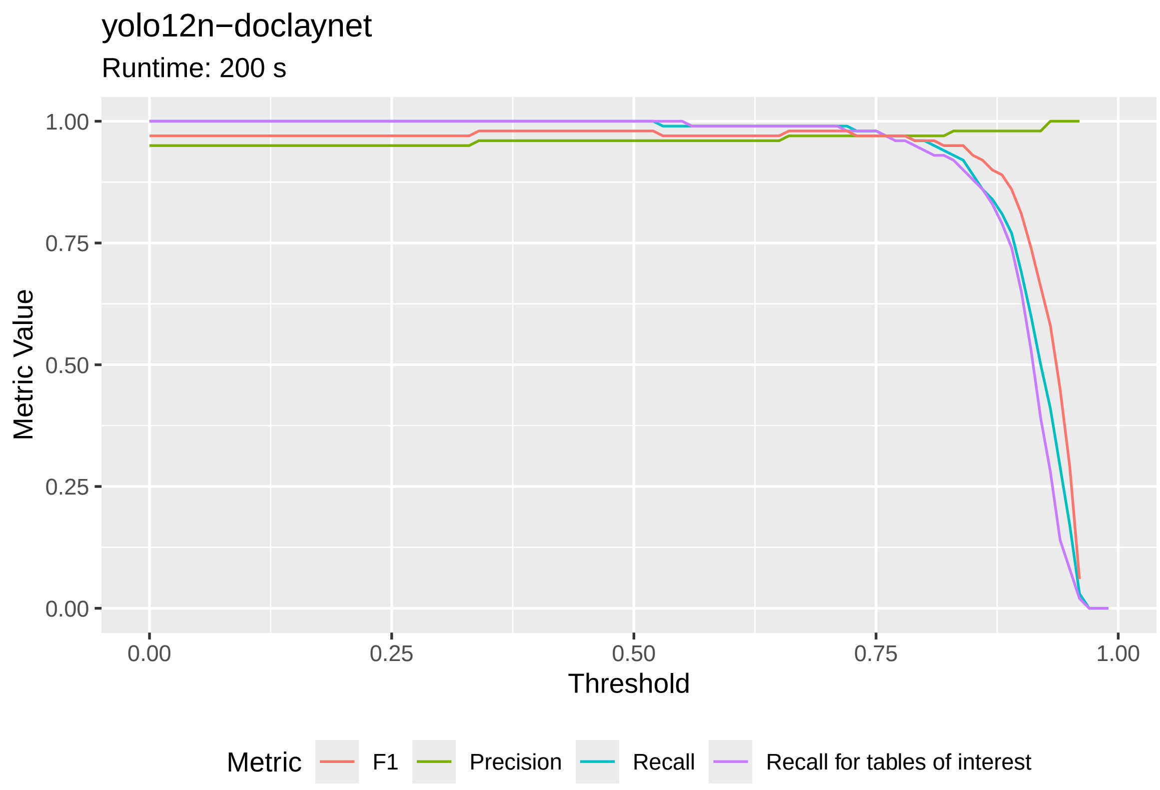


E

You see the plot for: microsoft-table-transformer-detection. (Click to stop automatic rotation.)

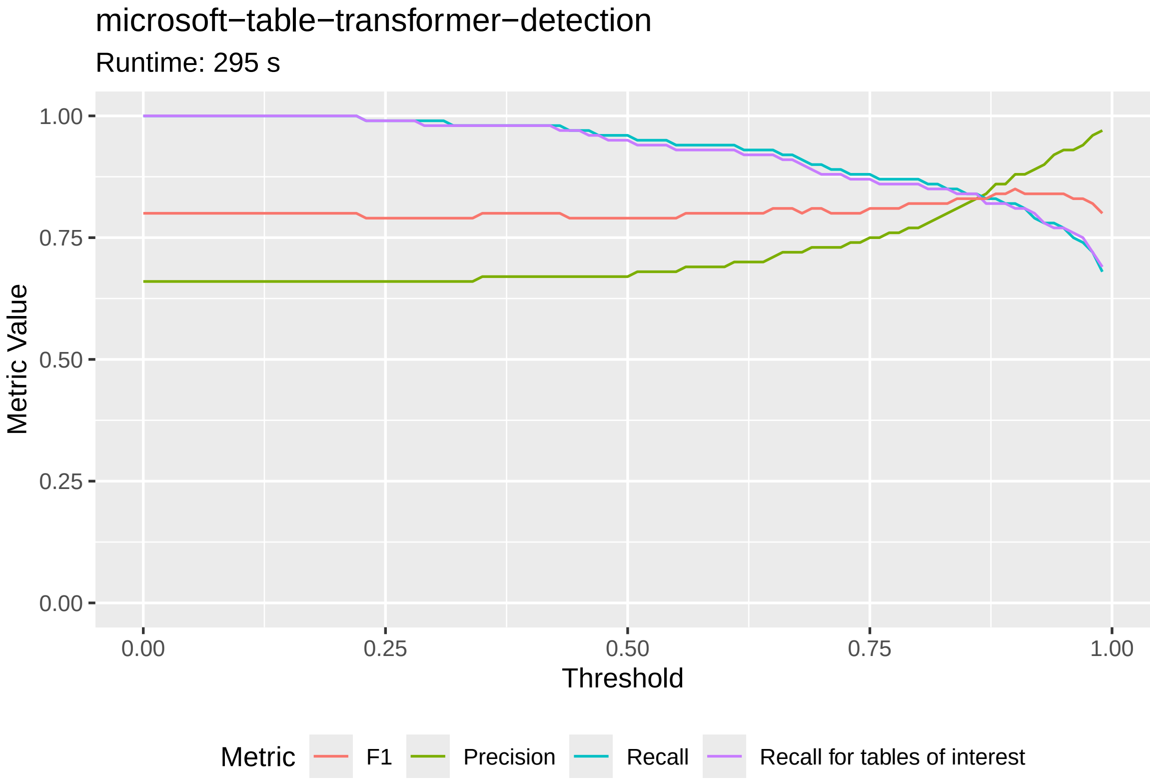


You see the plot for: yolo12l-doclaynet. (Click to stop automatic rotation.)

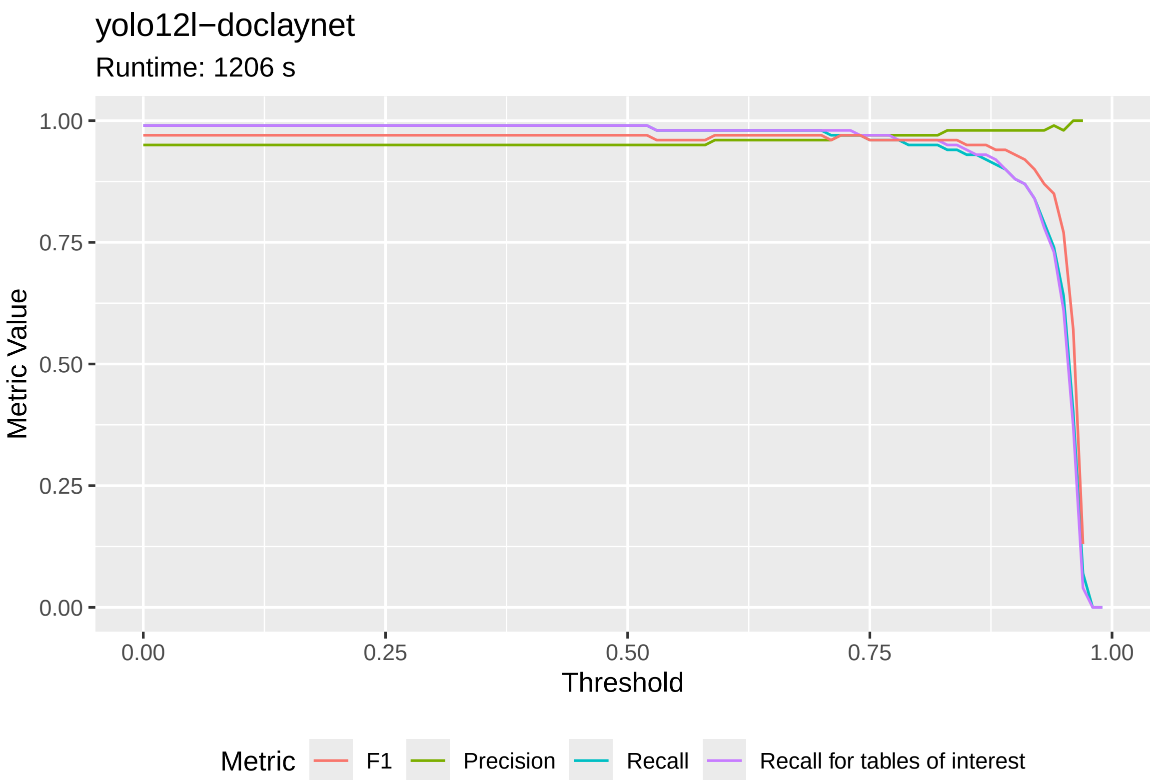


You see the plot for: yolo12n-doclaynet. (Click to stop automatic rotation.)

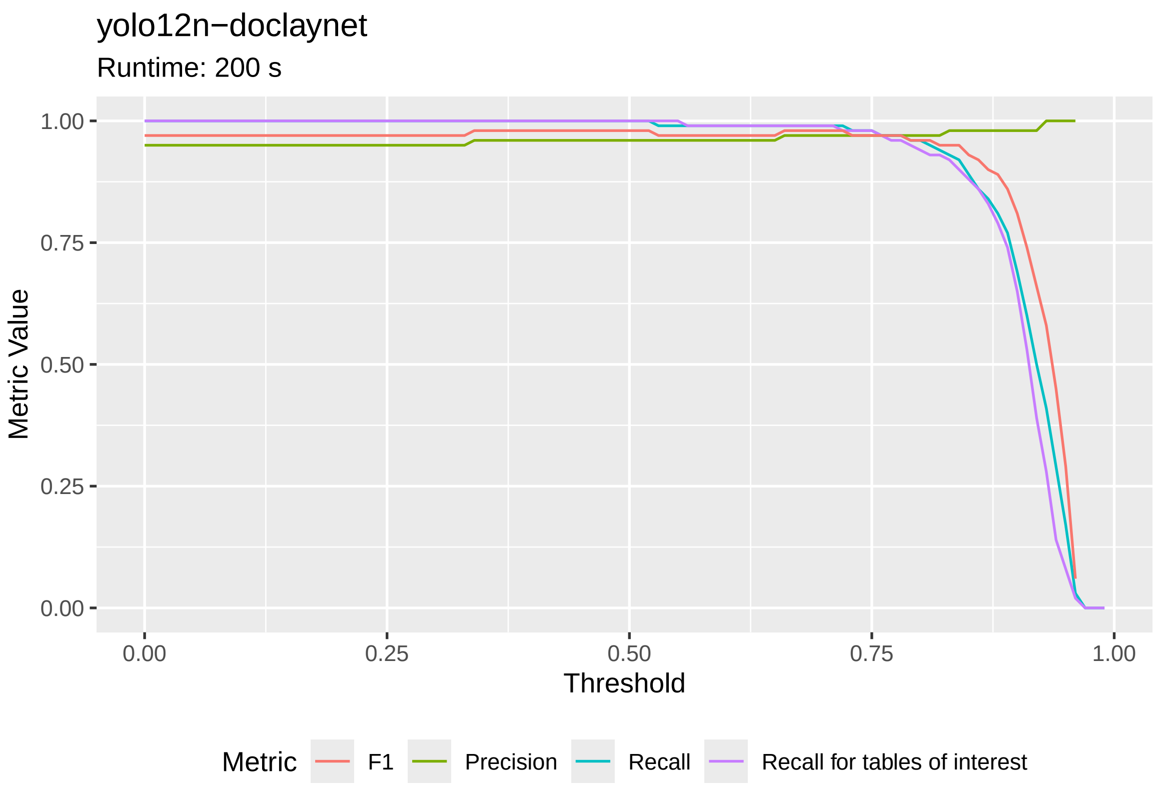
E



microsoft-table-transformer-detection



yolo12l-doclaynet



yolo12n-doclaynet

E.3.3 Large language model process speed

In April 2025 there have been issues with running vLLM within the Python framework. Thus, the first experiments are conducted, using the *transformers* library. When we managed to build a working vLLM based docker image for the experiments, we measured, how long the same task takes with the *transformers* and the vLLM library and how the batched processing competes versus a loop approach. The model family used is Qwen2.5-Instruct. The task is to extract the information from ten real **Aktiva** tables.

Table E.2 shows that the experiments with vLLM run around four to five times faster. Processing the messages in a batched mode is six to seven times faster than using a looped approach. Thus, the change of the experimental setup from a *transformers* powered loop-based approach to a vLLM powered batched processing approach, increased the speed by 2500 %.

This allows us to run the page identification benchmark on whole an nual reports, giving a sound estimate of the false positive rate (see section A.3). Previous experiments were only conducted on a subset of pages, that were selected based on the results of the *simple regex* approach (see section A.1).

E

Table E.2: Comparing time (in seconds) for extract the information from ten Aktiva tables using different libraries and approaches.

Model parameters (in B)	Transformers	vLLM	vLLM batched
0.5	330	65	NA
3.0	628	130	20
7.0	940	217	30

E.4 Prompts

E.4.1 TOC understanding

Base prompt:

```
    messages = [
        {"role": "system", "content": "You are a helpful assistant that can determine
↪   the page range information in a German financial report can be found at based on the
↪   documents table of contents."},
        {"role": "user", "content": f"This is the table of contents:\n\n{toc_string}"},
        {"role": "user", "content": f"On which pages might the win and loss statement
↪   (in German: Gewinn- und Verlustrechnung; GuV) and the balance sheets (German:
↪   Bilanz) be located? Give seperate answers for:\n\n1) the assets (German: Aktiva)
↪   table.\n2) the liabilities (German: Passiva) table.\n3) the win and loss
↪   statement."},
        specific_prompt,
        {"role": "user", "content": f"Answer in JSON format with keys 'GuV', 'Aktiva',
↪   and 'Passiva' and the page range as values."},
    ]
```

First attempt:

```
specific_prompt = {"role": "user", "content": f"The assets and liabilities tables often
↪   are on separate pages. They are often located directly before the win and loss
↪   statement. Rarely the tables for any of the three can span multiple pages."}
```

Given hint that assets and liabilities are part of the balance sheet:

```
specific_prompt = {"role": "user", "content": f"The assets and liabilities are part of
↪   the balance sheet (in German: Bilanz). The assets and liabilities tables often are
↪   on separate pages. They are often located directly before the win and loss
↪   statement. Rarely the tables for any of the three can span multiple pages."}
```

Stating, that liabilities are on next page:

```
specific_prompt = {"role": "user", "content": f"The assets and liabilities are part of
↪   the balance sheet (in German: Bilanz). The liabilities table is often on the page
↪   after the assets table. They are often located directly before the win and loss
↪   statement. Rarely the tables for any of the three can span multiple pages."}
```

TOC extraction from text prompt:

```
    messages = [
        {"role": "system", "content": "[Role] You are a helpful assistant that can
↪   identify table of contents in a German financial report."},
        {"role": "system", "content": f"[Context] These are the text lines of the first
↪   {i} pages:\n\n{start_pages}"},
        {"role": "user", "content": f"[Tasks] 1. Please identify if there is a table of
↪   contents in the text."},
        {"role": "user", "content": f"2. If there is a table of contents, please extract
↪   its text."},
        {"role": "user", "content": f"3. Answer as JSON with the table of contents text
↪   as string in the key 'toc'."},
        {"role": "user", "content": f"If there is no table of contents, return an empty
↪   string."},
    ]
```

E

E.4.2 Classification

binary classification prompt factory

```
messages = [{"role": "system", "content": "[Role and Context]: You are a helpful
↪  assistant that can classify texts extracted from PDFs."}]

if law_context:
    if classification_type == "GuV":
        messages.append({"role": "system", "content": f"You know the laws about how to
↪  structure the 'Gewinn- und Verlustrechnung' (profit and loss statement) table:'
↪  \n\n'''\n{hgb_guv}\n'''."})
    elif classification_type == "Aktiva":
        messages.append({"role": "system", "content": f"You know the laws about how to
↪  structure the 'Aktiva' (assets) table for a 'Bilanz' (balance sheet):'
↪  \n\n'''\n{hgb_aktiva}\n'''."})
    elif classification_type == "Passiva":
        messages.append({"role": "system", "content": f"You know the laws about how to
↪  structure the 'Passiva' (liabilities) table for a 'Bilanz' (balance sheet):'
↪  \n\n'''\n{hgb_passiva}\n'''."})
    else:
        raise ValueError(f"Unknown classification type: {classification_type}. Expected
        ↪  'GuV', 'Aktiva', or 'Passiva'.")

if random_examples:
    system_messages = self.__get_random_example_message(classification_type, **kwargs)
    for msg in system_messages:
        messages.append({"role": "system", "content": msg})

if rag_examples:
    system_messages = self.__get_rag_example_message(text, classification_type,
↪  **kwargs)
    for msg in system_messages:
        messages.append({"role": "system", "content": msg})

if top_n_rag_examples:
    system_messages = self.__get_top_n_rag_example_message(text, classification_type,
↪  **kwargs)
    for msg in system_messages:
        messages.append({"role": "system", "content": msg})

messages.append({"role": "user", "content": f"[Task]: Decide if the given text contains
↪  {phrase_dict[classification_type]}.\n\n[Rule]: Answer with 'yes' if it does.
↪  Otherwise answer with 'no'.\n\n[Text]: Here is the text to classify:
↪  \n\n'''\n{text}\n'''"})
return messages
```

example for binary classification with 1 random example with Qwen 3

```
<|im_start|>system
/no_think [Role and Context]: You are a helpful assistant that can classify texts
↪  extracted from PDFs.<|im_end|>
<|im_start|>system
You know this example for a \'Gewinn- und Verlustrechnung\' (profit and loss statement)
↪  table and for this example you should answer with "no":

\'\'\'
28
```

```
2023
EUR
2022
EUR
EUR EUR
1. Umsatzerlöse 1.315.073,26 1.507.621,05
2. Sonstige betriebliche Erträge 562.644,72 631.803,96
3. Materialaufwand -388.989,26 -98.471,89
4. Abschreibungen -447.356,00 -460.923,00
5. Sonstige betriebliche Aufwendungen -907.414,53 -2.304.390,53
6. Sonstige Zinsen und ähnliche Erträge 95.260,94 -2.533,45
7. Ergebnis nach Steuern 229.219,13 -726.893,86
8. Sonstige Steuern -857.535,62 -879.289,10
9. Jahresfehlbetrag -628.316,49 -1.606.182,96
Gewinn- und Verlustrechnung
für die Zeit vom 01. Januar bis 31. Dezember 2023
\'\'\'.<|im_end|>
<|im_start|>system
You know this example for a \'Aktiva\' (assets) table and for this example you should
 ↪  answer with "yes":

\'\'\'
BEN Berlin Energie und Netzholding GmbH (vormals: Berlin Energie Rekom 2 GmbH)
Berlin
Bilanz zum 31.12.2021
Aktivseite 31.12.2021 31.12.2020 31.12.2021 31.12.2020
T€ T€ T€ T€
A. Anlagevermögen A. Eigenkapital
imv I. Immaterielle Vermögensgegenstände 0,8 - ek I. Gezeichnetes Kapital 25,0 25,0
bga II. Sachanlagen 73,1 - kr II. Kapitalrücklage 6,9 6,9
III. Finanzanlagen 2.094.146,0 - vv III. Verlustvortrag - 6,9 - 6,9
IV. Jahresüberschuss 1.326,7 -
 2.094.219,9 -
 1.351,7 25,0
B. Umlaufvermögen sor
unf I. Forderungen und sonstige B. Rückstellungen
 Vermögensgegenstände Sonstige Rückstellungen 265,1 6,7
Forderungen gegen verbundene
Unternehmen 423,1 - anzC. Verbindlichkeiten
1. Verbindlichkeiten gegenüber
fll II. Guthaben bei Kreditinstituten 166.662,0 39,2 vll Kreditinstituten 2.180.051,3 -
2. Verbindlichkeiten aus
 167.085,1 39,2 Lieferungen und Leistungen 91,9 1,9
3. Verbindlichkeiten gegenüber
 verbundenen Unternehmen 81.286,7 -
C. Rechnungsabgrenzungsposten 2.471,2 - vvu 4. Verbindlichkeiten gegenüber
 Gesellschaftern 713,9 5,6
5. Sonstige Verbindlichkeiten 15,6 -
 2.262.159,4 7,5
 2.263.776,2 39,2 2.263.776,2 39,2
Passivseite
 21-006917
\'\'\'.<|im_end|>
<|im_start|>system
You know this example for a \'Passiva\' (liabilities) table and for this example you
 ↪  should answer with "no":

\'\'\'
4
```

```
 Bilanz Elektrizitätsverteilung
Aktiva 31.12.2022
T€
Anlagevermögen
imv Immaterielle Vermögensgegenstände -
bga Sachanlagen -
Finanzanlagen -
 -
Umlaufvermögen
unf Forderungen und sonstige Vermögensgegenstände 329,6
davon Verrechnungsposten gegenüber anderen Aktivitäten 289,9
fll Guthaben bei Kreditinstituten -
 329,6
Rechnungsabgrenzungsposten 17,9
 347,6
Passiva 31.12.2022
T€
Eigenkapital
ek Gezeichnetes Kapital -
kr Kapitalrücklage -
vv Gewinnrücklage/Verlustvortrag -
Jahresüberschuss 0,1
 0,1
Rückstellungen
Sonstige Rückstellungen 258,4
Verbindlichkeiten
anz Verbindlichkeiten gegenüber Kreditinstituten -
vll Verbindlichkeiten aus Lieferungen und Leistungen 89,0
Verbindlichkeiten gegenüber Gesellschaftern -
Sonstige Verbindlichkeiten -
 89,0
 347,6
\'\'\'.<|im_end|>
<|im_start|>system
You know this example for a text that does not suit the categories of interest and for
 ↪  this example you should answer with "no":

\'\'\'
Bericht des
Aufsichtsrates
Sehr geehrte Damen,
sehr geehrte Herren,
mit diesem Bericht informieren wir über unsere Tätigkeit im Geschäftsjahr 2016
und das Ergebnis der Prüfung des Jahresabschlusses. Die uns nach Gesetz, Satzung
und Geschäftsordnung obliegenden Kontroll- und Beratungsaufgaben haben
wir verantwortungsvoll und mit der gebührenden Sorgfalt wahrgenommen. Dabei
haben wir den Vorstand bei der Leitung der GESOBAU beratend begleitet, seine
Tätigkeit überwacht und waren in alle für die Gesellschaft grundlegend bedeutenden
Entscheidungen unmittelbar eingebunden. Der Vorstand ist seinen
 ↪  Informations\x02pflichten uneingeschränkt nachgekommen und hat uns regelmäßig sowohl
 ↪  schrift\x02lich als auch mündlich informiert. Dies geschah zeitnah und umfassend zu
 ↪  allen
Aspekten der Unternehmensplanung, dem Verlauf der Geschäfte, der strategischen
Weiterentwicklung sowie der aktuellen Lage des Unternehmens. Planabweichungen
beim Geschäftsverlauf wurden uns im Einzelnen erläutert und mit schlüssigen
Argumenten begründet. Der Vorstand stimmte die strategische Ausrichtung des
Unternehmens vertrauensvoll mit uns ab. Die für das Unternehmen bedeutenden
Geschäftsvorgänge haben wir auf der Basis der Berichte des Vorstandes ausführlich
erörtert und seinen Beschlussvorschlägen nach gründlicher Prüfung und Beratung
```

```
zugestimmt.
Sitzungen
Im Berichtsjahr fanden vier turnusgemäße und eine außerordentliche Sitzung statt.
Die Sitzungen des Aufsichtsrates sind von einem intensiven und offenen Austausch
geprägt. Ein Mitglied des Aufsichtsrates hat im abgelaufenen Geschäftsjahr an
weniger als der Hälfte der Sitzungen teilgenommen. Aufgrund besonderer
 ↪  Eilbe\x02dürftigkeit erfolgten in Abstimmung mit der Vorsitzenden des Aufsichtsrates
 ↪  vier
Beschlussfassungen im Umlaufverfahren.
Die Mitglieder des Aufsichtsrates bereiten sich auf anstehende Beschlüsse regelmäßig
auch anhand von Unterlagen vor, die der Vorstand vorab zur Verfügung stellt. Dabei
wurden sie von den jeweils zuständigen Ausschüssen unterstützt. Die
 ↪  Aufsichtsrats\x02sitzungen werden zudem von den Arbeitnehmervertretern in Gesprächen
 ↪  mit dem
Vorstand vorbereitet.
Information durch den Vorstand
Über die wichtigsten Indikatoren der Geschäftsentwicklung und bestehende Risiken
unterrichtet der Vorstand den Aufsichtsrat anhand schriftlicher Quartalsberichte.
Zwischen den Sitzungsterminen des Aufsichtsrates und seiner Ausschüsse wurde
die Aufsichtsratsvorsitzende ausführlich unterrichtet. Hierbei wurde die Strategie
des Unternehmens besprochen, wie auch die aktuelle Geschäftsentwicklung und
-lage, das Risikomanagement, Fragen der Compliance sowie wesentliche Einzel\x02themen
 ↪  und bevorstehende bedeutsame Entscheidungen erörtert.
16 Perspektiven Bericht des Aufsichtsrates
\'\'\'.<|im_end|>
<|im_start|>user
[Task]: Decide if the given text contains a \'Aktiva\' (assets) table.

[Rule]: Answer with \'yes\' if it does. Otherwise answer with \'no\'.

[Text]: Here is the text to classify:

\'\'\'
22 Amt für Statistik Berlin-Brandenburg | Geschäftsbericht 2014
Amt für Statistik Berlin-Brandenburg Anstalt des öffentlichen Rechts, Potsdam
Bilanz zum 31. Dezember 2014
A K T I V S E I T E 31.12.2014 Vorjahr
EUR EUR TEUR
A. ANLAGEVERMÖGEN
I. Immaterielle Vermögensgegenstände
1. Entgeltlich erworbene Konzessionen, gewerbliche
Schutzrechte und ähnliche Rechte und Werte
sowie Lizenzen an solchen Rechten und Werten 81.480,00 146
II. Sachanlagen
1. Grundstücke, grundstücksgleiche Rechte und Bauten
einschließlich der Bauten auf fremden Grundstücken 68.386,00 93
2. Andere Anlagen, Betriebs- und Geschäftsausstattung 140.186,00 174
208.572,00 267
III. Finanzanlagen
1. Wertpapiere des Anlagevermögens 2.000.000,00 2.000
2.000.000,00 2.000
2.290.052,00 2.413
B. UMLAUFVERMÖGEN
I. Forderungen und sonstige Vermögensgegenstände
1. Forderungen aus Lieferungen und Leistungen 36.617,86 14
2. Sonstige Vermögensgegenstände 297.982,42 267
334.600,28 281
II. Kassenbestand, Bundesbankguthaben, Guthaben bei
Kreditinstituten und Schecks 5.560.638,85 7.783
```

E

```
5.895.239,13 8.064
C. RECHNUNGSABGRENZUNGSPOSTEN 216.321,49 213
8.401.612,62 10.690
Bestätigungsvermerk
des Abschlussprüfers
Anhang
\'\'\'<|im_end|>
<|im_start|>assistant
```

multi-class classification prompt factory

```
messages = [
  {"role": "system", "content": "[Role and Context]: You are a helpful assistant that
 ↪  can classify texts extracted from PDFs."},
]

if law_context:
  messages.append({"role": "system", "content": f"You know the laws about how to
 ↪  structure the 'Gewinn- und Verlustrechnung' (profit and loss statement) table:'
 ↪  \n\n'''\n{hgb_guv}\n'''."})
  messages.append({"role": "system", "content": f"You also know the laws about how to
 ↪  structure the 'Aktiva' (assets) and 'Passiva' (liabilities) table for a 'Bilanz'
 ↪  (balance sheet):' \n\n'''\n{hgb_bilanz}\n'''."})

if random_examples:
  system_messages = self.__get_random_example_message(**kwargs)
  for msg in system_messages:
      messages.append({"role": "system", "content": msg})

if rag_examples:
  system_messages = self.__get_rag_example_message(text, **kwargs)
  for msg in system_messages:
      messages.append({"role": "system", "content": msg})

if top_n_rag_examples:
  system_messages = self.__get_top_n_rag_example_message(text, **kwargs)
  for msg in system_messages:
      messages.append({"role": "system", "content": msg})

messages.append({"role": "user", "content": f"""
[Task]: Decide of what type the given text is. You can differentiate between four types
 ↪  of pages: 'Aktiva', 'GuV', 'Passiva' and 'other'.\n\n
[Rules]:\n
  1) If the text contains a 'Gewinn- und Verlustrechnung' (profit and loss statement)
 ↪  table, answer with 'GuV'.\n\n
  2) If the text contains an 'Aktiva' (assets) table, answer with 'Aktiva'.\n\n
  3) If the text contains a 'Passiva' (liabilities) table, answer with 'Passiva'.\n\n
  4) If the text contains something else, answer with 'other'.\n\n
[Text]: Here is the text to classify: \n\n'''\n{text}\n'''
"""})
```

E

example for multi-class classification with 1 rag example with Qwen 3

```
<|im_start|>system
/no_think [Role and Context]: You are a helpful assistant that can classify texts
 ↪  extracted from PDFs.<|im_end|>
<|im_start|>system
```

```
You know this example for a \'Gewinn- und Verlustrechnung\' (profit and loss statement)
 ↪  table and for this example you should answer with "GuV":

"""
74
Gewinn- und Verlustrechnung für die Zeit vom 01.01.2014 bis 31.12.2014
Aufwendungen in TEUR Vorjahr
1. Zinsaufwendungen 302.081 314.077
2. Provisionsaufwendungen 714 656
4. Allgemeine Verwaltungsaufwendungen
a) Personalaufwand
 aa) Löhne und Gehälter
 ab) Soziale Abgaben und Aufwendungen
 für Altersversorgung und für Unterstützung
 darunter: für Altersversorgung
b) andere Verwaltungsaufwendungen

39.535
9.009
2.417
48.544
31.161
79.705
39.310
11.020
4.651
50.330
24.983
75.313
5. Abschreibungen und Wertberichtigungen auf immaterielle
Anlagewerte und Sachanlagen 3.647 3.707
6. Sonstige betriebliche Aufwendungen 25.803 26.412
7. Abschreibungen und Wertberichtigungen auf Forderungen und
bestimmte Wertpapiere sowie Zuführungen zu
Rückstellungen im Kreditgeschäft 25.366 14.666
8. Abschreibungen und Wertberichtigungen auf Beteiligungen,
Anteile an verbundenen Unternehmen
und wie Anlagevermögen behandelte Wertpapiere 421 0
9. Aufwendungen aus Verlustübernahme 1.268 0
13. Sonstige Steuern, soweit nicht unter Posten 6 ausgewiesen 65 80
15. Jahresüberschuss 25.863 36.897
Summe der Aufwendungen 464.933 471.808
Jahresüberschuss 25.863 36.897
Gewinnvortrag aus dem Vorjahr 0 0
Bilanzgewinn 25.863 36.897
An unsere Geschäftspartner | Grußwort der Vorsitzenden des Verwaltungsrats | Bericht des
 ↪  Verwaltungsrats
Wohnungsbauförderung | Wirtschaftsförderung | Beteiligungen | Immobilien- und
 ↪  Stadtentwicklung | Personalbericht | Nachhaltigkeit
Lagebericht | Jahresabschluss | Anhang | Bestätigungsvermerk |
 ↪  Corporate-Governance-Bericht | Organigramm
""". (The L2 distance of this example text is: 0.562)<|im_end|>
<|im_start|>system
You know this example for a \'Aktiva\' (assets) table and for this example you should
 ↪  answer with "Aktiva":

"""
52 Gruppenbilanz
Gruppenbilanz zum 31. Dezember 2016
```

```
A K T I V A 31. 12. 2016 31. 12. 2015
€ € €
A. ANLAGEVERMÖGEN
I. Immaterielle Vermögensgegenstände
Entgeltlich erworbene Konzessionen, gewerbliche
Schutzrechte und ähnliche Rechte 122.148,00 185.602,00
II. Sachanlagen
1. Anlageimmobilien 3.423.064.255,69 3.338.758.481,04
2. übrige Grundstücke und Bauten 4.143.376,87 1.087.406,00
3. technische Anlagen und Maschinen 120.700,00 149.667,00
4. andere Anlagen, Betriebs- und Geschäftsausstattung 5.143.477,51 4.555.161,48
5. geleistete Anzahlungen und Anlagen im Bau 1.007.468,36 180.543,58
3.433.479.278,43 3.344.731.259,10
III. Finanzanlagen
1. Anteile an verbundenen Unternehmen 1.026.647,27 1.027.646,27
2. Ausleihungen an verbundene Unternehmen 157.645,00 214.395,00
3. Beteiligungen 284.138,88 40.073,02
4. sonstige Ausleihungen 120.966,91 120.966,91
1.589.398,06 1.403.081,20
3.435.190.824,49 3.346.319.942,30
B. UMLAUFVERMÖGEN
I. Vorräte
1. unfertige Leistungen 48.642.315,18 52.057.422,25
2. andere Vorräte 13.053,63 21.315,99
48.655.368,81 52.078.738,24
II. Forderungen und sonstige Vermögensgegenstände
1. Forderungen aus Lieferungen und Leistungen 32.107.301,91 35.679.035,16
2. Forderungen gegen verbundene Unternehmen 74.457,55 554.130,28
3. Forderungen gegen Unternehmen,
mit denen ein Beteiligungsverhältnis besteht 108.647,73 23.698,71
4. sonstige Vermögensgegenstände 100.560.866,41 100.896.144,88
132.851.273,60 137.153.009,03
III. Wertpapiere
sonstige Wertpapiere 1.700,00 1.700,00
IV. Kassenbestand, Guthaben bei Kreditinstituten 893.140.123,18 689.887.519,98
1.074.648.465,59 879.120.967,25
C. RECHNUNGSABGRENZUNGSPOSTEN 9.245.284,80 9.917.197,12
D. AKTIVER UNTERSCHIEDSBETRAG AUS DER
VERMÖGENSVERRECHNUNG 68.523,69 0,00
4.519.153.098,57 4.235.358.106,67

""". (The L2 distance of this example text is: 0.421)<|im_end|>
<|im_start|>system
You know this example for a \'Passiva\' (liabilities) table and for this example you
 ↪ should answer with "Passiva":

"""
Anlage 1
BEN Berlin Energie und Netzholding GmbH
Berlin
Bilanz zum 31.12.2023
Aktivseite 31.12.2023 31.12.2022 31.12.2023 31.12.2022
T€ T€ T€ T€
A. Anlagevermögen A. Eigenkapital
imv I. Immaterielle Vermögensgegenstände 58,0 20,2 ek I. Gezeichnetes Kapital 25,0
 ↪ 25,0bga II. Sachanlagen 106,7 70,7 kr II. Kapitalrücklage 6,9 6,9
III. Finanzanlagen 2.194.146,0 2.094.146,0 vv III. Gewinnrücklage/Verlustvortrag
 ↪ 41.023,4 1.319,8
IV. Jahresüberschuss 51.158,5 39.703,6
```

```
 2.194.310,6 2.094.236,9
 92.213,8 41.055,3
B. Umlaufvermögen sor
unf I. Forderungen und sonstige B. Rückstellungen
 Vermögensgegenstände Sonstige Rückstellungen 4.759,3 460,0
1. Forderungen aus Lieferungen und Leistungen 73,1 70,72. Forderungen gegen verbundene
 ↪  C. Verbindlichkeiten Unternehmen 96.998,2 60.960,4 anz 1. Verbindlichkeiten
 ↪  gegenüber 3. Sonstige Vermögensgegenstände 988,6 923,3 Kreditinstituten 2.317.498,9
 ↪  2.148.050,6
fll II. Guthaben bei Kreditinstituten 226.047,2 160.535,8 vll 2. Verbindlichkeiten aus
 Lieferungen und Leistungen 272,0 158,4
 324.107,1 222.490,2 3. Verbindlichkeiten gegenüber
 verbundenen Unternehmen 104.704,9 128.407,54. Verbindlichkeiten gegenüber
C. Rechnungsabgrenzungsposten 1.969,9 2.207,9 vvu Gesellschaftern 695,8 706,1
5. Sonstige Verbindlichkeiten 242,9 97,1
 2.423.414,4 2.277.419,7
 2.520.387,6 2.318.935,0 2.520.387,6 2.318.935,0
Passivseite
3
""". (The L2 distance of this example text is: 0.481)<|im_end|>
<|im_start|>system
You know this example for a text that does not suit the categories of interest and for
 ↪  this example you should answer with "other":

"""
46 Konzernbilanz
Konzernbilanz zum 31. Dezember 2013
A K T I V A 31. 12. 2013 31. 12. 2012
€ € €
A. ANLAGEVERMÖGEN
I. Immaterielle Vermögensgegenstände
Konzessionen, gewerbliche Schutzrechte und ähnliche
Rechte
344.384,00 461.417,00
II. Sachanlagen
1. Grundstücke und Bauten 1.242.921,00 1.272.566,00
2. Technische Anlagen und Maschinen 122.769,00 62.405,00
3. Andere Anlagen, Betriebs- und Geschäftsausstattung 2.339.362,51 1.562.893,45
4. Geleistete Anzahlungen 704,76 33.483,89
3.705.757,27 2.931.348,34
III. Finanzanlagen
1. Anteile an verbundenen Unternehmen 3.201.349,87 3.201.436,42
2. Ausleihungen an verbundene Unternehmen 217.680,00 223.395,00
3. Beteiligungen 42.171.545,24 54.585.174,81
4. Sonstige Ausleihungen 76.015.926,17 99.994.824,65
121.606.501,28 158.004.830,88
125.656.642,55 161.397.596,22
B. UMLAUFVERMÖGEN
I. Vorräte
1. Unfertige Leistungen 12.885.172,94 8.843.369,97
2. Zum Verkauf bestimmte Grundstücke und Gebäude 139.000,00 139.002,00
3. Andere Vorräte 61.319,05 93.039,06
13.085.491,99 9.075.411,03
II. Forderungen und sonstige Vermögensgegenstände
1. Forderungen aus Lieferungen und Leistungen 8.666.340,95 12.099.596,63
2. Forderungen gegen verbundene Unternehmen 1.409.363,51 7.573.168,86
3. Forderungen gegen Unternehmen,
mit denen ein Beteiligungsverhältnis besteht
555.093,06 1.651.573,06
```

```
4. Sonstige Vermögensgegenstände 345.991.815,13 163.003.969,98
356.622.612,65 184.328.308,53
III. Wertpapiere
Sonstige Wertpapiere 52.252.850,00 59.329.212,00
IV. Kassenbestand, Guthaben bei Kreditinstituten 152.594.976,48 248.363.122,67
574.555.931,12 501.096.054,23
C. RECHNUNGSABGRENZUNGSPOSTEN 7.545.702,82 7.957.871,65
707.758.276,49 670.451.522,10
Treuhandvermögen 1.943.915.141,66 1.953.309.522,69
""". (The L2 distance of this example text is: 0.434)<|im_end|>
<|im_start|>user

[Task]: Decide of what type the given text is. You can differentiate between four types
↪  of pages: \'Aktiva\', \'GuV\', \'Passiva\' and \'other\'.


[Rules]:

1) If the text contains a \'Gewinn- und Verlustrechnung\' (profit and loss statement)
↪  table, answer with \'GuV\'.


2) If the text contains an \'Aktiva\' (assets) table, answer with \'Aktiva\'.


3) If the text contains a \'Passiva\' (liabilities) table, answer with \'Passiva\'.


4) If the text contains something else, answer with \'other\'.


[Text]: Here is the text to classify:

\'\'\'
22 Amt für Statistik Berlin-Brandenburg | Geschäftsbericht 2014
Amt für Statistik Berlin-Brandenburg Anstalt des öffentlichen Rechts, Potsdam
Bilanz zum 31. Dezember 2014
A K T I V S E I T E 31.12.2014 Vorjahr
EUR EUR TEUR
A. ANLAGEVERMÖGEN
I. Immaterielle Vermögensgegenstände
1. Entgeltlich erworbene Konzessionen, gewerbliche
Schutzrechte und ähnliche Rechte und Werte
sowie Lizenzen an solchen Rechten und Werten 81.480,00 146
II. Sachanlagen
1. Grundstücke, grundstücksgleiche Rechte und Bauten
einschließlich der Bauten auf fremden Grundstücken 68.386,00 93
2. Andere Anlagen, Betriebs- und Geschäftsausstattung 140.186,00 174
208.572,00 267
III. Finanzanlagen
1. Wertpapiere des Anlagevermögens 2.000.000,00 2.000
2.000.000,00 2.000
2.290.052,00 2.413
B. UMLAUFVERMÖGEN
I. Forderungen und sonstige Vermögensgegenstände
1. Forderungen aus Lieferungen und Leistungen 36.617,86 14
2. Sonstige Vermögensgegenstände 297.982,42 267
334.600,28 281
II. Kassenbestand, Bundesbankguthaben, Guthaben bei
```

```
Kreditinstituten und Schecks 5.560.638,85 7.783
5.895.239,13 8.064
C. RECHNUNGSABGRENZUNGSPOSTEN 216.321,49 213
8.401.612,62 10.690
Bestätigungsvermerk
des Abschlussprüfers
Anhang
\'\'\'
        <|im_end|>
<|im_start|>assistant
```

E.5 Regular expressions

Here one can find the three regular expressions used for the benchmarks presented in section A.1.

```
simple_regex_patterns = {
    "Aktiva": [
        r"aktiva",
        r"((20\d{2}).*(20\d{2}))"
    ],
    "Passiva": [
        r"passiva",
        r"((20\d{2}).*(20\d{2}))"
    ],
    "GuV": [
        r"gewinn",
        r"verlust",
        r"rechnung",
        r"((20\d{2}).*(20\d{2}))"
    ]
}
```

```
regex_patterns_5 = {
    "Aktiva": [

        ↪  r"a\s*k\s*t\s*i\s*v\s*a|a\s*k\s*t\s*i\s*v\s*s\s*e\s*i\s*t\s*e|anlageverm.{1,2}gen",
        r"((20\d{2}).*(20\d{2}))|((20\d{2}).*vorjahr)|vorjahr",

        ↪  r"Umlaufverm.{1,2}gen|Anlageverm.{1,2}gen|Rechnungsabgrenzungsposten|Forderungen",
        r"\s([a-zA-Z]|[0-9]{1,2}|[iI]+)[\.\)]\s"
    ],
    "Passiva": [

        ↪  r"p\s*a\s*s\s*s\s*i\s*v\s*a|p\s*a\s*s\s*s\s*i\s*v\s*s\s*e\s*i\s*t\s*e|eigenkapital",
        r"((20\d{2}).*(20\d{2}))|((20\d{2}).*vorjahr)|vorjahr",

        ↪  r"Eigenkapital|R.{1,2}ckstellungen|Verbindlichkeiten|Rechnungsabgrenzungsposten",
        r"\s([a-zA-Z]|[0-9]{1,2}|[iI]+)[\.\)]\s"
    ],
    "GuV": [
        r"gewinn|guv",
        r"verlust|guv",
        r"rechnung|guv",
        r"((20\d{2}).*(20\d{2}))|vorjahr"

        ↪  r"Umsatzerl.{1,2}se|Materialaufwand|Personalaufwand|Abschreibungen|Jahres.{1,2}berschuss|Ja
```

E

```
        r"\s([a-zA-Z]|[0-9]{1,2}|[iI]+)[\.\)]\s"
    ]
}
```

```
regex_patterns_3 = {
    "Aktiva": [

         ↪  r"a\s*k\s*t\s*i\s*v\s*a|a\s*k\s*t\s*i\s*v\s*s\s*e\s*i\s*t\s*e|anlageverm.{1,2}gen",
        r"((20\d{2}).*(20\d{2}))|((20\d{2}).*vorjahr)|vorjahr"
    ],
    "Passiva": [

         ↪  r"p\s*a\s*s\s*s\s*i\s*v\s*a|p\s*a\s*s\s*s\s*i\s*v\s*s\s*e\s*i\s*t\s*e|eigenkapital",
        r"((20\d{2}).*(20\d{2}))|((20\d{2}).*vorjahr)|vorjahr"
    ],
    "GuV": [
        r"gewinn|guv",
        r"verlust|guv",
        r"rechnung|guv",
        r"((20\d{2}).*(20\d{2}))|vorjahr"
    ]
}
```

E.6 Annual Comprehensive Financial Report Balance Sheet

E.7 Table extraction with regular expressions

Extract by pdfium for '../../benchmark_truth/synthetic_tables/separate_files/final/aktiva_table__3_columns__span_False__thir
€__enumeration_False__shuffle_True__text_around_True__max_length_50__sum_in_same_row_False__0.pdf':

```
A

ktiva(inMio. €)GeschäftsjahrVorjahr

Anlagevermögen Immaterielle Verm

ögensgegenstände

SelbstgeschaffenegewerblicheSchutzrechteund

ähnlicheRechteundWerte

0,184,77

Geschäfts-oderFirmenwert4,426,78

geleisteteAnzahlungen1,780,65

entgeltlicherworbeneKonzessionen, gewerbliche

SchutzrechteundähnlicheRechteundWertesowie

LizenzenansolchenRechtenundWerten

4,646,71
```

State of California Annual Comprehensive Financial Report

Balance Sheet

Governmental Funds

June 30, 2023

(amounts in thousands)

		General		Federal
ASSETS				
Cash and pooled investments	$	71,968,861	$	6,986,275
Investments		—		—
Receivables (net)		46,621,774		2,076,598
Due from other funds		6,933,803		165,231
Due from other governments		4,075,837		37,069,188
Interfund receivables		3,914,413		—
Loans receivable		45,225		384,293
Other assets		6,244		601,252
Total assets	**$**	**133,566,157**	**$**	**47,282,837**
LIABILITIES				
Accounts payable	$	14,422,777	$	24,499,200
Due to other funds		3,911,973		3,865,533
Due to component units		264,995		—
Due to other governments		21,808,112		11,125,464
Interfund payables		2,692,941		—
Benefits payable		—		69,623
Revenues received in advance		25,891		6,675,956
Tax overpayments		21,740,974		—
Deposits		4,231		—
Unclaimed property liability		1,314,797		—
Other liabilities		522,844		46,256,400
Total liabilities		**66,709,535**		**92,492,176**
DEFERRED INFLOWS OF RESOURCES		2,852,934		10,709
Total liabilities and deferred inflows of resources		**69,562,469**		**92,502,885**
FUND BALANCES				
Nonspendable		3,950,919		—
Restricted		24,830,454		1,210,267
Committed		4,210,891		—
Assigned		20,714,283		—
Unassigned		10,297,141		(46,430,315)
Total fund balances (deficit)		**64,003,688**		**(45,220,048)**
Total liabilities, deferred inflows of resources, and fund balances	**$**	**133,566,157**	**$**	**47,282,837**

42 *The notes to the financial statements are an integral part of this statement.*

Figure E.3: Example balance sheet pagefom Californias Annual Comprehensive Financial Report 2023

E

```
11,0218,91

Sachanlagen

Grundstücke, grundstücksgleicheRechteundBauten

einschließlichderBautenauffremdenGrundstücken

2,802,55

TechnischeAnlagenundMaschinen5,205,53

AndereAnlagen, Betriebs-undGeschäftsausstattung1,601,93

geleisteteAnzahlungenundAnlagen imBau3,255,81

12,8615,83

Finanzanlagen

SonstigeFinanzanlagen7,446,51

AnteileanverbundenenUnternehmen0,499,83

AusleihungenanverbundeneUnternehmen0,573,49

Beteiligungen1,059,43

AusleihungenanUnternehmen, mitdenenein

Beteiligungsverhältnisbesteht

6,957,65

WertpapieredesAnlagevermögens2,002,71

SonstigeAusleihungen9,091,52

27,5841,13

51,4675,87

Umlaufvermögen

Vorräte

Roh-, Hilfs-undBetriebsstoffe0,382,98

UnfertigeErzeugnisse, unfertigeLeistungen3,236,19

FertigeErzeugnisseundWaren6,724,98

GeleisteteAnzahlungen4,024,83

14,3418,98

ForderungenundsonstigeVermögensgegenstände
```

E

```
ForderungenausLieferungenundLeistungen4,328,36

ForderungengegenverbundeneUnternehmen6,082,38

ForderungengegenUnternehmen, mitdenenein

Beteiligungsverhältnisbesteht

7,878,11

SonstigeVermögensgegenstände1,968,30

20,2227,15

Wertpapiere

AnteileanverbundenenUnternehmen2,383,24

SonstigeWertpapiere0,077,65

2,4410,88

Kassenbestand, Bundesbankguthaben, Guthabenbei Kreditinstituten und Schecks

4,144,00

41,1561,01

Rechnungsabgrenzungsposten2,746,78

Aktive latenteSteuern8,464,60

AktiverUnterschiedsbetragausder

Vermögensverrechnung

2,863,35

106,67151,61
```

Extract by pdfminer for '../../benchmark_truth/synthetic_tables/separate_files/final/aktiva_table__3_columns__span_False_
€__enumeration_False__shuffle_True__text_around_True__max_length_50__sum_in_same_row_False__0.pdf':

```
Aktiva (in Mio. €)

Anlagevermögen

Immaterielle Vermögensgegenstände

Selbst geschaffene gewerbliche Schutzrechte und
ähnliche Rechte und Werte

Geschäfts- oder Firmenwert

geleistete Anzahlungen

entgeltlich erworbene Konzessionen, gewerbliche
Schutzrechte und ähnliche Rechte und Werte sowie
```

```
Lizenzen an solchen Rechten und Werten

Sachanlagen

Grundstücke, grundstücksgleiche Rechte und Bauten
einschließlich der Bauten auf fremden Grundstücken

Technische Anlagen und Maschinen

Andere Anlagen, Betriebs- und Geschäftsausstattung

geleistete Anzahlungen und Anlagen im Bau

Finanzanlagen

Sonstige Finanzanlagen

Anteile an verbundenen Unternehmen

Ausleihungen an verbundene Unternehmen

Beteiligungen

Ausleihungen an Unternehmen, mit denen ein
Beteiligungsverhältnis besteht

Wertpapiere des Anlagevermögens

Sonstige Ausleihungen

Umlaufvermögen

Vorräte

Roh-, Hilfs- und Betriebsstoffe

Unfertige Erzeugnisse, unfertige Leistungen

Fertige Erzeugnisse und Waren

Geleistete Anzahlungen

Forderungen und sonstige Vermögensgegenstände

Forderungen aus Lieferungen und Leistungen

Forderungen gegen verbundene Unternehmen

Forderungen gegen Unternehmen, mit denen ein
Beteiligungsverhältnis besteht

Sonstige Vermögensgegenstände

Wertpapiere

Anteile an verbundenen Unternehmen

Sonstige Wertpapiere
```

```
Kassenbestand, Bundesbankguthaben, Guthaben bei
Kreditinstituten und Schecks

Rechnungsabgrenzungsposten

Aktive latente Steuern

Aktiver Unterschiedsbetrag aus der
Vermögensverrechnung

Geschäftsjahr

Vorjahr

0,18

4,42

1,78

4,64

11,02

2,80

5,20

1,60

3,25

12,86

7,44

0,49

0,57

1,05

6,95

2,00

9,09

27,58

51,46

0,38

3,23

6,72

4,02
```

```
14,34

4,32

6,08

7,87

1,96

20,22

2,38

0,07

2,44

4,14

41,15

2,74

8,46

2,86

4,77

6,78

0,65

6,71

18,91

2,55

5,53

1,93

5,81

15,83

6,51

9,83

3,49

9,43

7,65
```

E

```
2,71

1,52

41,13

75,87

2,98

6,19

4,98

4,83

18,98

8,36

2,38

8,11

8,30

27,15

3,24

7,65

10,88

4,00

61,01

6,78

4,60

3,35

106,67

151,61
```

Extract by PdfReader for '../Geschaeftsberichte/IBB/ibb_geschaeftsbericht_2006.pdf', p. 67:

```
'\x18\x18
Jahresbilanz zum 31. Dezember 2006
a ktivseite in te U r
31.12.2006 31.12.2005
1. Barreserve


b)
```

```
Guthaben

bei

Zentralnotenbanken
darunter:

bei

der

Deutschen

Bundesbank:
TEUR

19.823

(31.12.2005

:

TEUR

28.873)
3. Forderungen an
k
reditinstitute
     a)

täglich

fällig
     b)

andere

Forderungen
4. Forderungen an
k
unden


darunter:

durch

Grundpfandrechte

gesichert:
TEUR

9.496.661

(31.12.2005

:
```

```
TEUR

10.660.277)
Kommunalkredite:

TEUR

3.532.796

(31.12.2005

:

TEUR

2.338.961)
5. Schuldverschreibungen und andere festverzinsliche Wertpapiere
     a)

    Geldmarktpapiere

         ab)

von

anderen

Emittenten
     b)

    Anleihen

und

Schuldverschreibungen

         ba)

von

öffentlichen

Emittenten

         darunter:

beleihbar

bei

der

Deutschen

Bundesbank

         bb)

von
```

E

```
anderen

Emittenten

         darunter:

beleihbar

bei

der

Deutschen

Bundesbank
     c)

    eigene

Schuldverschreibungen
         Nennbetrag
7. Beteiligungen


darunter:

an

Kreditinstituten

TEUR

0

(31.12.2005

:

TEUR

0)
8.
a
nteile an verbundenen Unternehmen


darunter:

an

Kreditinstituten

TEUR

0

(31.12.2005
```

```
:

TEUR

0)
9.
t
reuhandvermögen

    darunter:

Treuhandkredite
11. Immaterielle
a
nlagewerte
12. Sachanlagen
15. Sonstige
v
ermögensgegenstände
16.
r
echnungsabgrenzungsposten
Summe der
a
ktiva
19.823


132.272
1.562.290
24.138
126.223
126.223
3.115.852
2.976.213
718
718
101.246
19.823
1.694.562
14.758.008
3.266.931
11.440
178.004
101.246
10.038
46.863
141.626
17.804
20.246.345
28.873
2.195.434
205.129
1.990.305
14.728.310
1.682.660
0
49.152
49.152
```

```
1.585.752
1.585.752
47.756
47.722
11.440
178.004
103.297
103.297
17.705
50.334
141.791
11.957
19.149.805
Jahresabschluss  | Jahresbilanz'
```

E.8 Term frequency missclassifications

Term counts for '../Geschaeftsberichte/GESOBAU AG/GESOBAU_Geschaeftsbericht_2012.pdf' page 36:

```
{'Vorjahr': 0,

 'Bau': 3,

 'Aktive': 1,

 'Immaterielle': 0,

 'Kassenbestand': 0,

 'Lieferungen': 0,

 'Hilfs': 0,

 'Anteile': 0,

 'Vermögensverrechnung': 0,

 'Rechten': 0,

 'Betriebsstoffe': 0,

 'Geschäfts': 19,

 'Beteiligungen': 0,

 'Wertpapiere': 0,

 'Betriebs': 0,

 'Sachanlagen': 0,

 'Lizenzen': 0,

 'Umlaufvermögen': 0,

 'Vorräte': 0,
```

E

```
'Kreditinstituten': 0,
'Grundstücke': 0,
'Schecks': 0,
'Unterschiedsbetrag': 0,
'Aktiver': 1,
'Werte': 0,
'Guthaben': 0,
'Konzessionen': 0,
'Unternehmen': 10,
'Leistungen': 1,
'Werten': 0,
'Aktiva': 0,
'Ausleihungen': 0,
'Anzahlungen': 0,
'Finanzanlagen': 0,
'Aktivseite': 0,
'Maschinen': 0,
'Anlagevermögens': 0,
'Forderungen': 0,
'Rechte': 0,
'Anlagevermögen': 0,
'Beteiligungsverhältnis': 0,
'Firmenwert': 0,
'Bundesbankguthaben': 0,
'III': 1,
'Erzeugnisse': 0,
'Geschäftsjahr': 1,
'Vermögensgegenstände': 0,
'Geschäftsausstattung': 0,
'Roh': 0,
```


E

```
'Rechnungsabgrenzungsposten': 0,

'Grundstücken': 0,

'Steuern': 0,

'Anlagen': 1,

'Bauten': 1,

'Schutzrechte': 0}
```

```
knitr::include_graphics("images/many_floats.png")
```

E

- Herr Dr. Eugen von Lackum

Bezüge	2017
in €	
Aufwandsentschädigung	3.850,81
Erstattung Reisekosten	582,63

- Frau Erika Jaeger

Bezüge	2017
in €	
Aufwandsentschädigung einschließlich Umsatzsteuer	7.675,50

- Herr Dr. Uwe Lissau

Bezüge	2017
in €	
Aufwandsentschädigung	6.000,00
Erstattung Reisekosten	3.700,18

- Herr Prof. Dr.-Ing. Engelbert Lütke Daldrup

Bezüge	2017
in €	
Aufwandsentschädigung	6.000,00

- Herr Sebastian Scheel

Bezüge	2017
in €	
Aufwandsentschädigung einschließlich Umsatzsteuer	3.645,16

- Frau Sabine Usinger

Bezüge	2017
in €	
Aufwandsentschädigung einschließlich Umsatzsteuer	7.140,00

3.4 Beschäftigte Arbeitnehmer

Im Berichtszeitraum waren bei den Unternehmen des **berlinovo**-Konzerns durchschnittlich 351 (Vorjahr: 354) Angestellte beschäftigt. Daneben waren im Konzern im Jahresdurchschnitt 10 (Vorjahr: 9) Auszubildende angestellt. Zum 31. Dezember 2017 sind 354 (Vorjahresstichtag: 344) Angestellte und 10 (Vorjahresstichtag: 10) Auszubildende beschäftigt.

Figure E.4: Showing page 48 of '../Geschaeftsberichte/Berlinovo/geschaeftsbericht_berlinovo_2017_0.pdf' as a page with a high float frequency.

Chapter F

Tables

F.1 Classification

F.2 Table extraction

F.2.1 Hybrid approach

Table F.1: Comparing the actual number of provided examples depending on the classification type, example selection strategy and chosen parameter n_examples.

approach	classification	n_example	target	other	sum
n_random_examples	binary	1	1	1	4
n_random_examples	binary	3	3	1	6
n_random_examples	binary	5	5	2	11
n_random_examples	multi	1	1	1	4
n_random_examples	multi	3	3	3	12
n_random_examples	multi	5	5	5	20
n_rag_examples	binary	1	1	1	4
n_rag_examples	binary	3	3	1	6
n_rag_examples	binary	5	5	2	11
n_rag_examples	multi	1	1	1	4
n_rag_examples	multi	3	3	3	12
n_rag_examples	multi	5	5	5	20
top_n_rag_examples	binary	1	NA	NA	1
top_n_rag_examples	binary	3	NA	NA	3
top_n_rag_examples	binary	5	NA	NA	5
top_n_rag_examples	binary	7	NA	NA	7
top_n_rag_examples	binary	9	NA	NA	9
top_n_rag_examples	binary	11	NA	NA	11
top_n_rag_examples	binary	13	NA	NA	13
top_n_rag_examples	multi	1	NA	NA	1
top_n_rag_examples	multi	3	NA	NA	3
top_n_rag_examples	multi	5	NA	NA	5
top_n_rag_examples	multi	7	NA	NA	7
top_n_rag_examples	multi	9	NA	NA	9
top_n_rag_examples	multi	11	NA	NA	11
top_n_rag_examples	multi	13	NA	NA	13

Table F.2: Comparing extraction performance for real Aktiva extraction task with synthetic and real examples for incontext learning with a zero shot approach averaged over all methods

model	median_real	median_synth	median_zero_shot	delta_rate_real_synth	delta_
Qwen3-235B-A22B-Instruct-2507-FP8	**0.983**	**0.966**	**0.897**	0.5	
Llama-4-Scout-17B-16E-Instruct	0.931	0.897	0.448	0.33	
Mistral-Large-Instruct-2411	0.966	0.897	0.776	0.67	
Llama-3.1-8B-Instruct	0.828	0.759	0.552	0.286	
Qwen3-8B	0.931	0.759	0.336	**0.714**	
Ministral-8B-Instruct-2410	0.862	0.741	0.552	0.467	
gemma-3-27b-it	0.862	0.672	0.207	0.579	
gemma-3-12b-it	0.793	0.5	0.543	0.586	

Chapter G

Figures

NA predicting

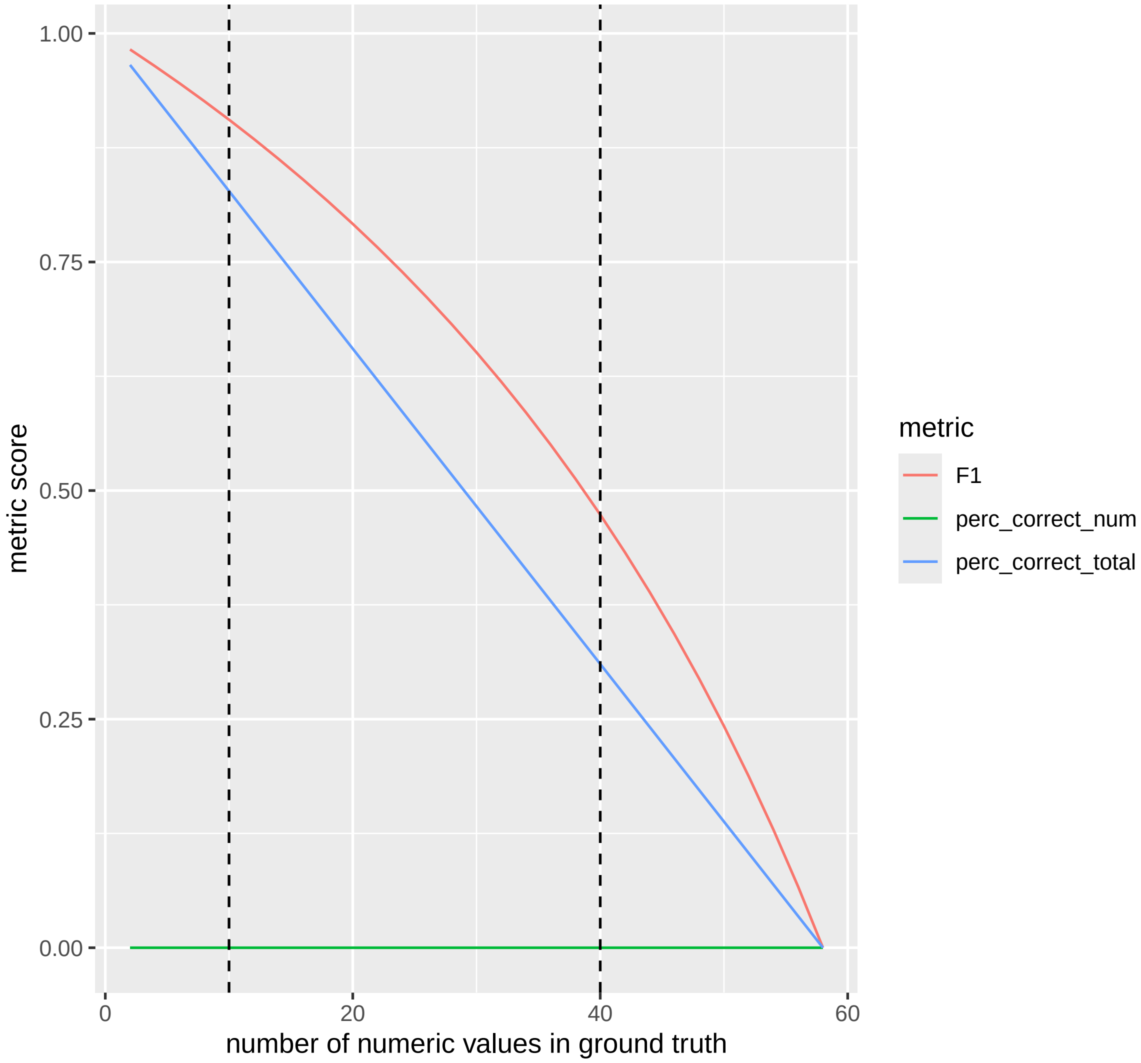


Figure G.1: Displaying the performance metrics a LLMs response would have, if all predictions are 'null'. The area between the two dashed lines shows the number of numeric values found in the real Aktiva tables.

G.1 Page identification

G.1.1 Regex baseline

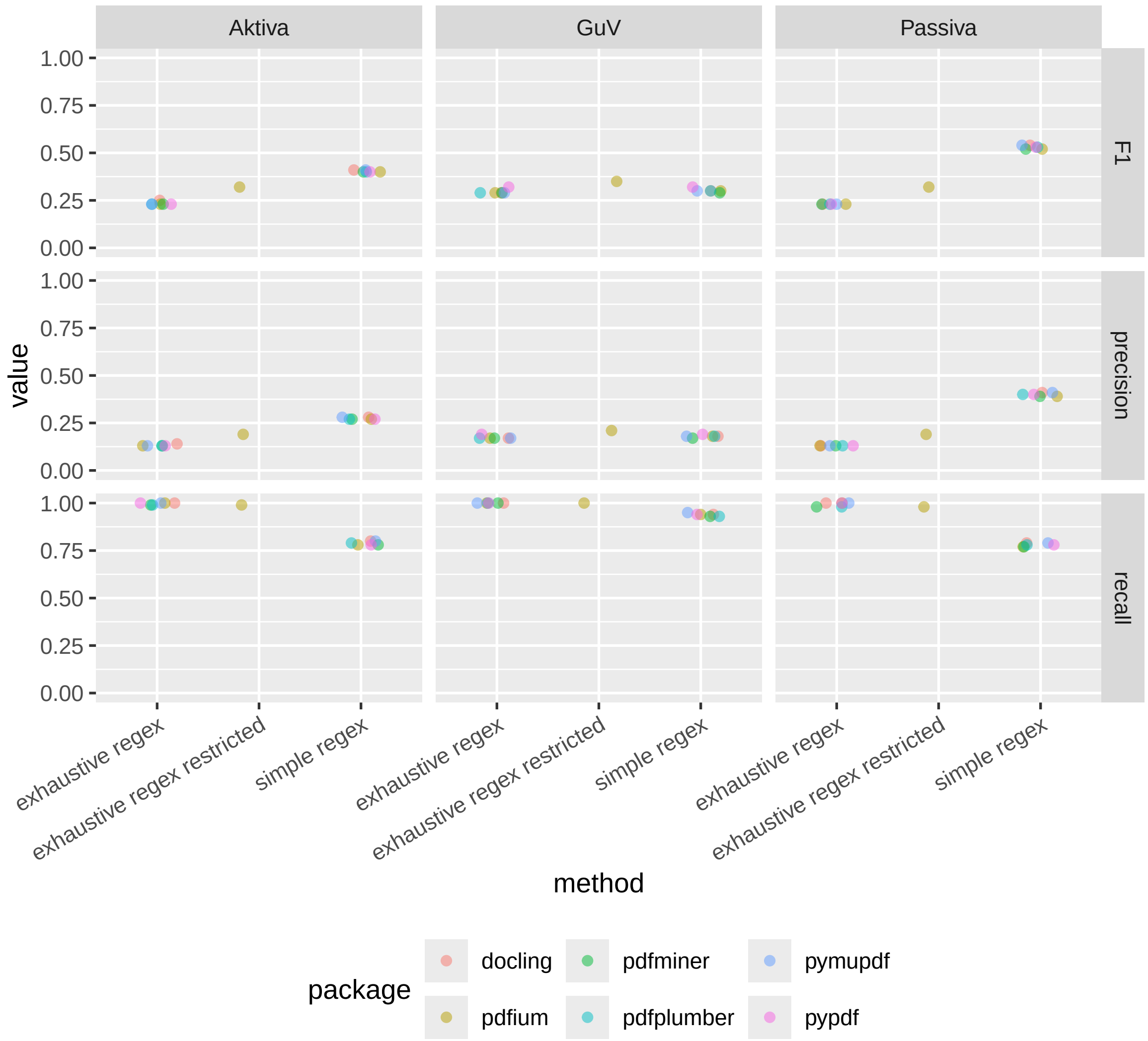


Figure G.2: Comparing page identification metrics for different regular expressions for each classification task by type of the target table.

G.1.2 TOC understanding

G.1.3 Classification

G.1.3.1 Binary

Binary classification F1 score over runtime limited to 60 minutes

Binary classification F1 score over runtime unlimited

G.1.3.2 Multi-class classification

Multi-class classification micro minorites F1 score over runtime limited to 60 minutes

Multi-class classification micro minorites F1 score over runtime

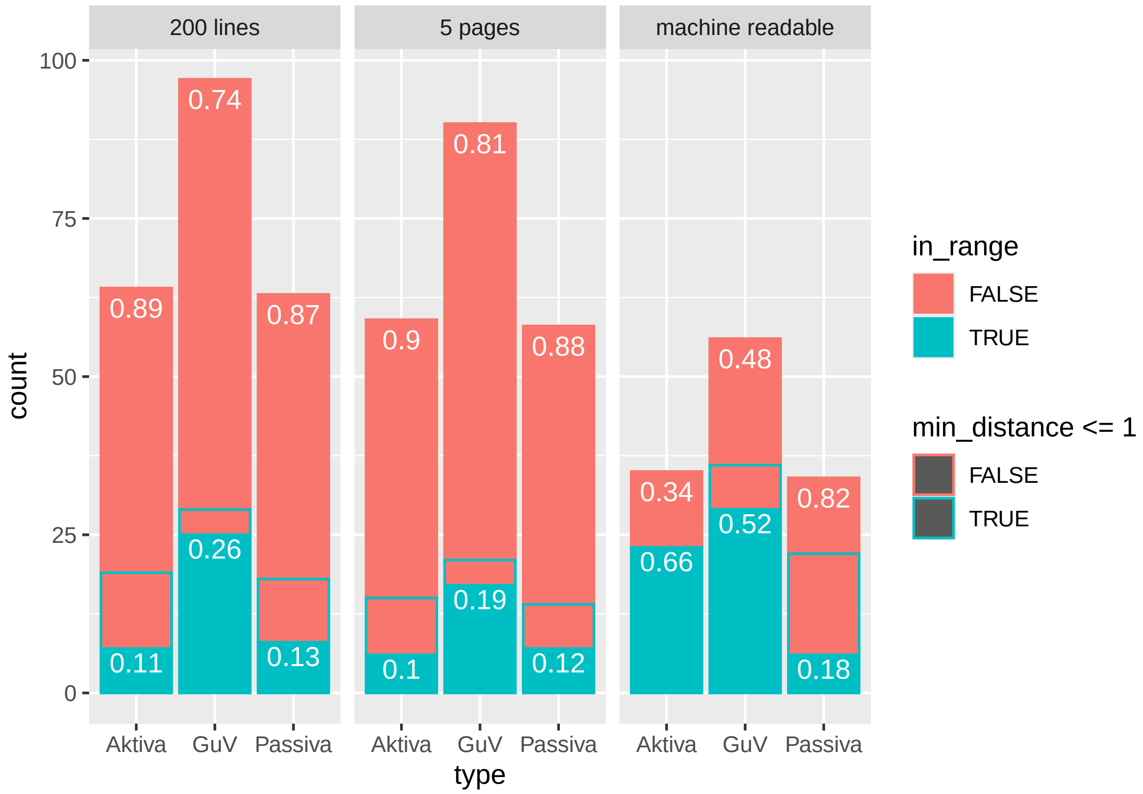


Figure G.3: Comparing number of fount TOC and amount of correct and incorrect predicted page ranges

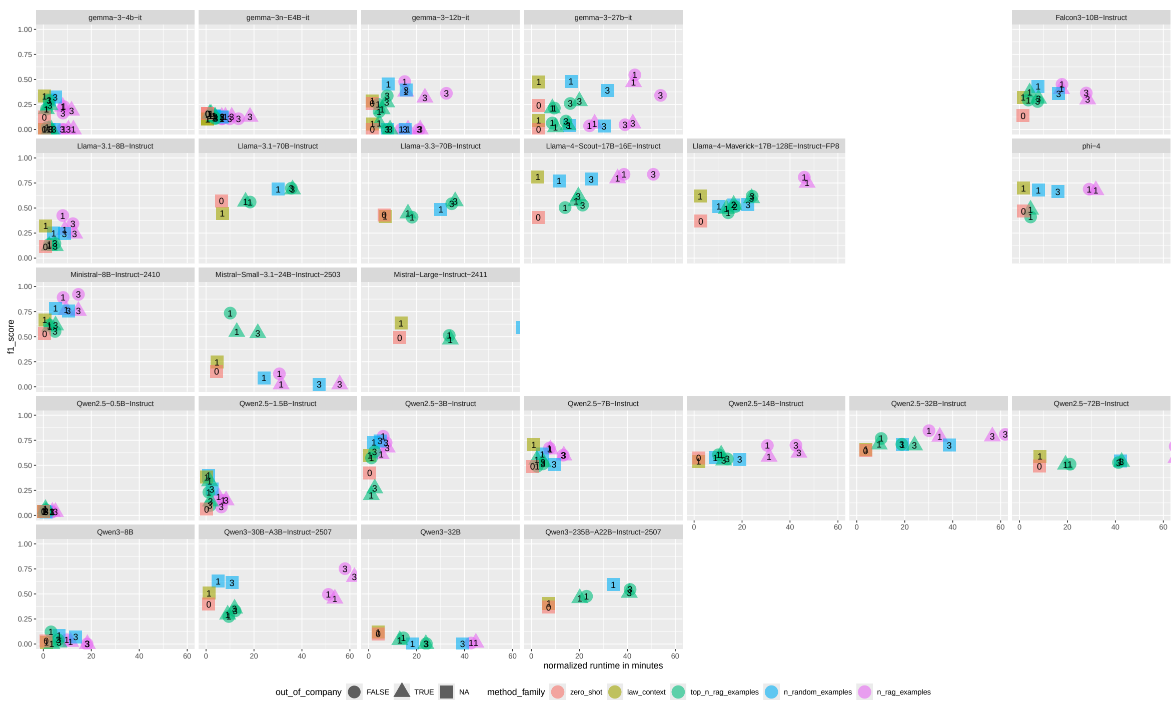


Figure G.4: Comparing F1 score over normalized runtime for binary classification task. The normalized runtime is given in minutes of processing on a single B200. The time to load the model into the VRAM is excluded. Focussing on small models showing only 60 minutes of runtime.

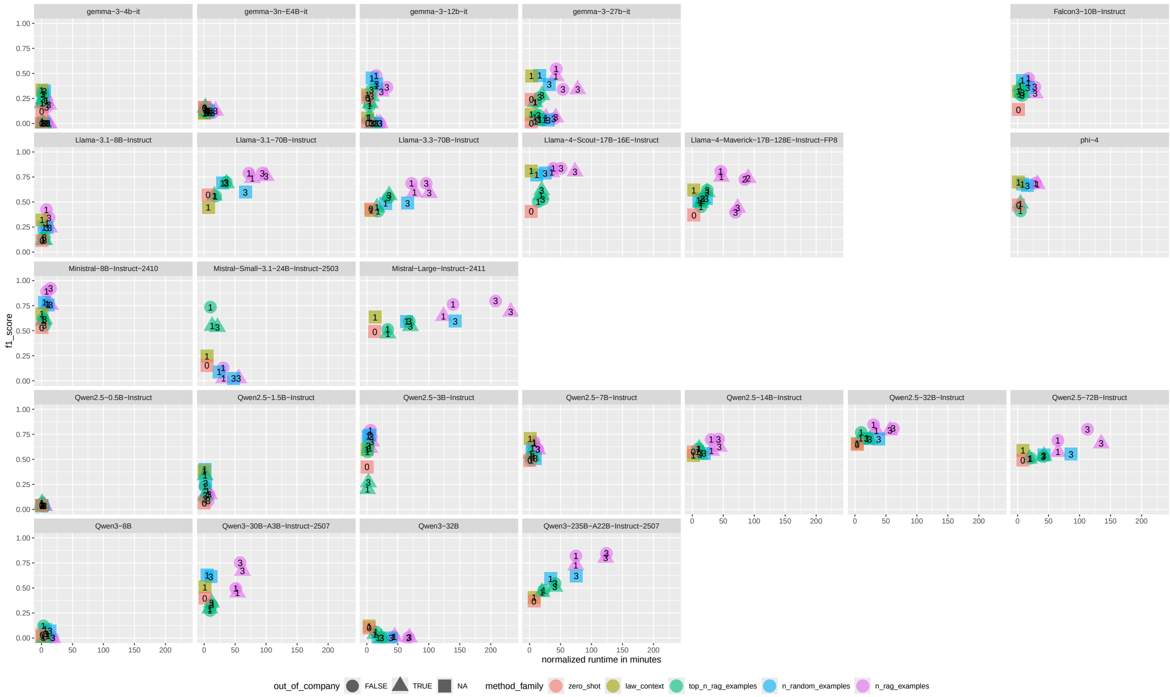


Figure G.5: Comparing F1 score over normalized runtime for binary classification task. The normalized runtime is given in minutes of processing on a single B200. The time to load the model into the VRAM is excluded.

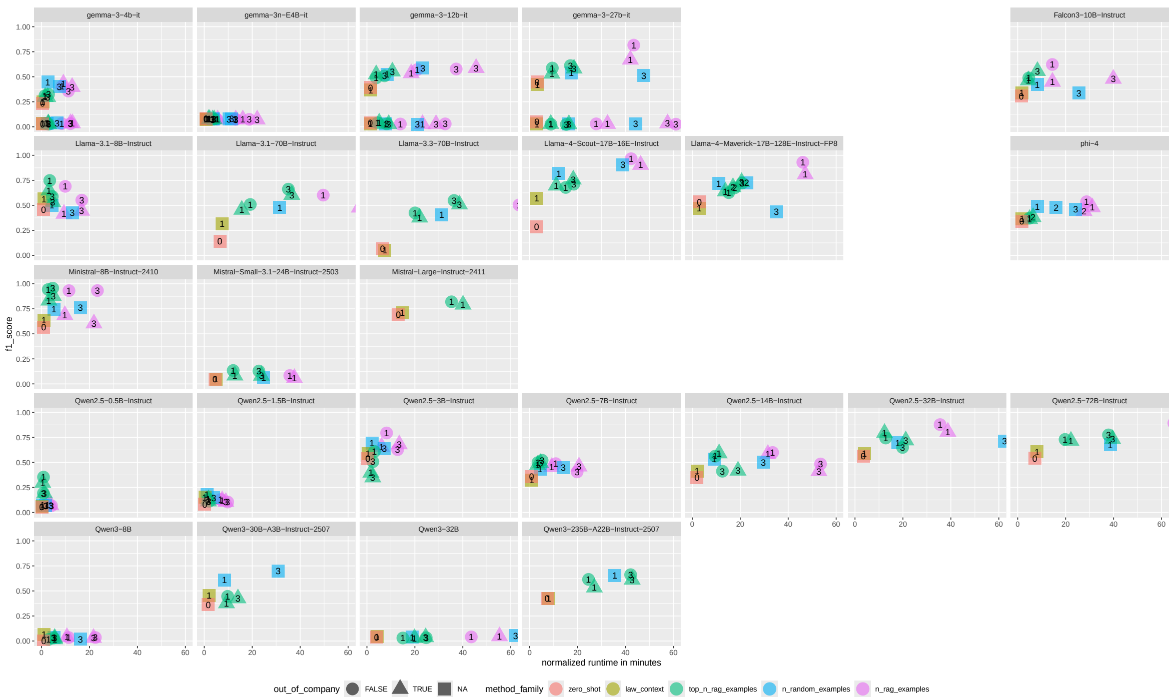


Figure G.6: Comparing F1 score over normalized runtime for multi-class classification task. The normalized runtime is given in minutes of processing on a single B200. The time to load the model into the VRAM is excluded. Focussing on small models showing only 60 minutes of runtime.

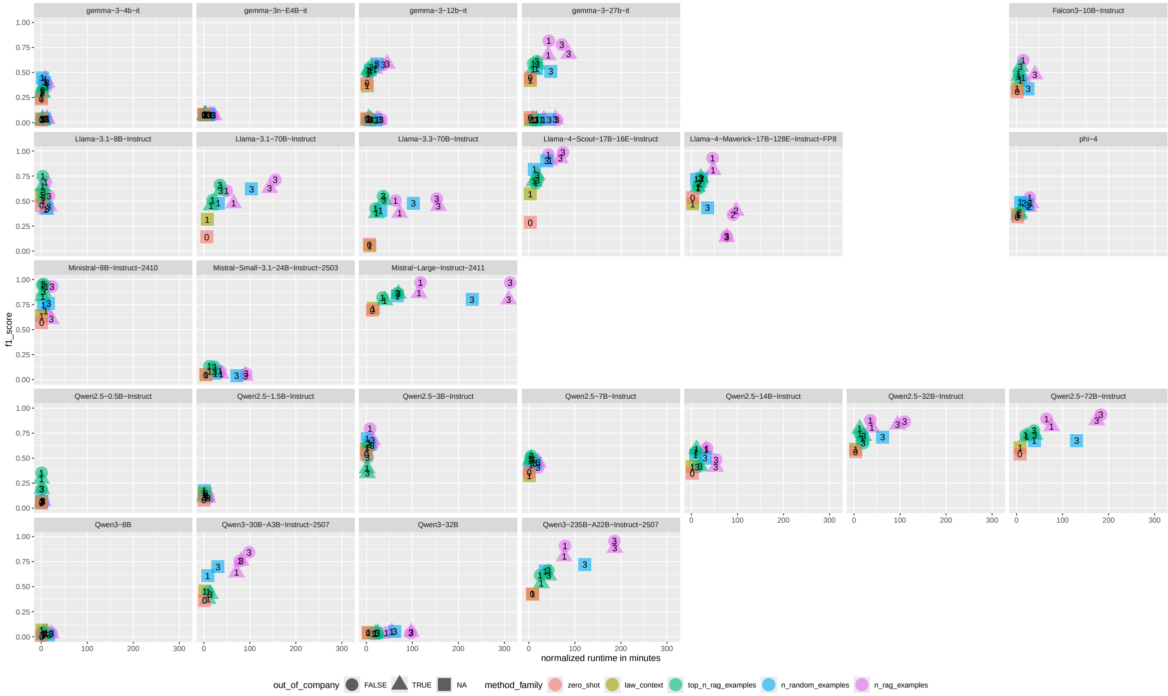


Figure G.7: Comparing F1 score over normalized runtime for multi-class classification task. The normalized runtime is given in minutes of processing on a single B200. The time to load the model into the VRAM is excluded.

G.2 Table extraction

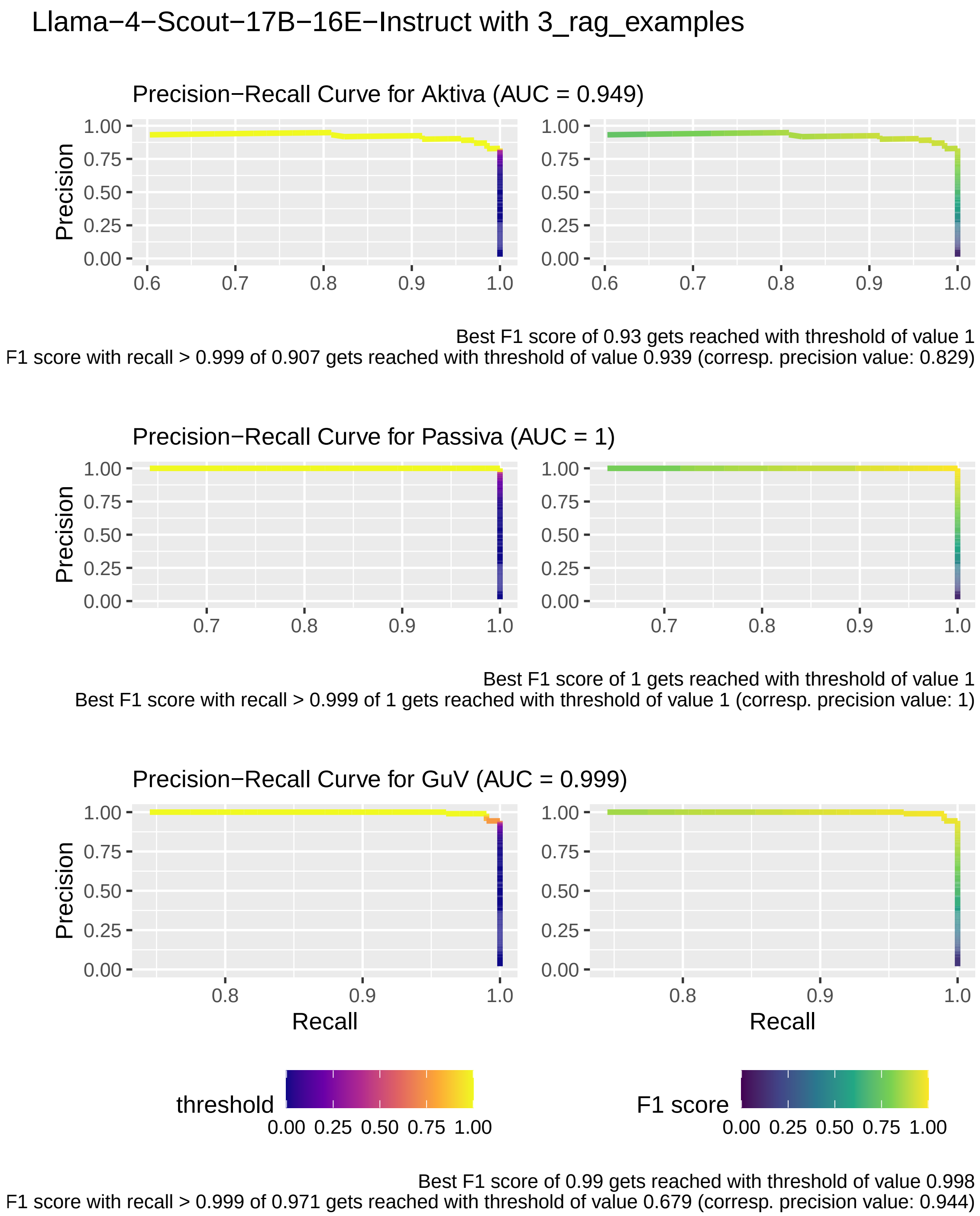


Figure G.8: Showing the precsion-recall-curve for Llama-4-Scout-17B-16E-Instruct.

G.2.1 Regex approach

G.2.1.1 Real tables

G.2.1.2 Synthetic tables

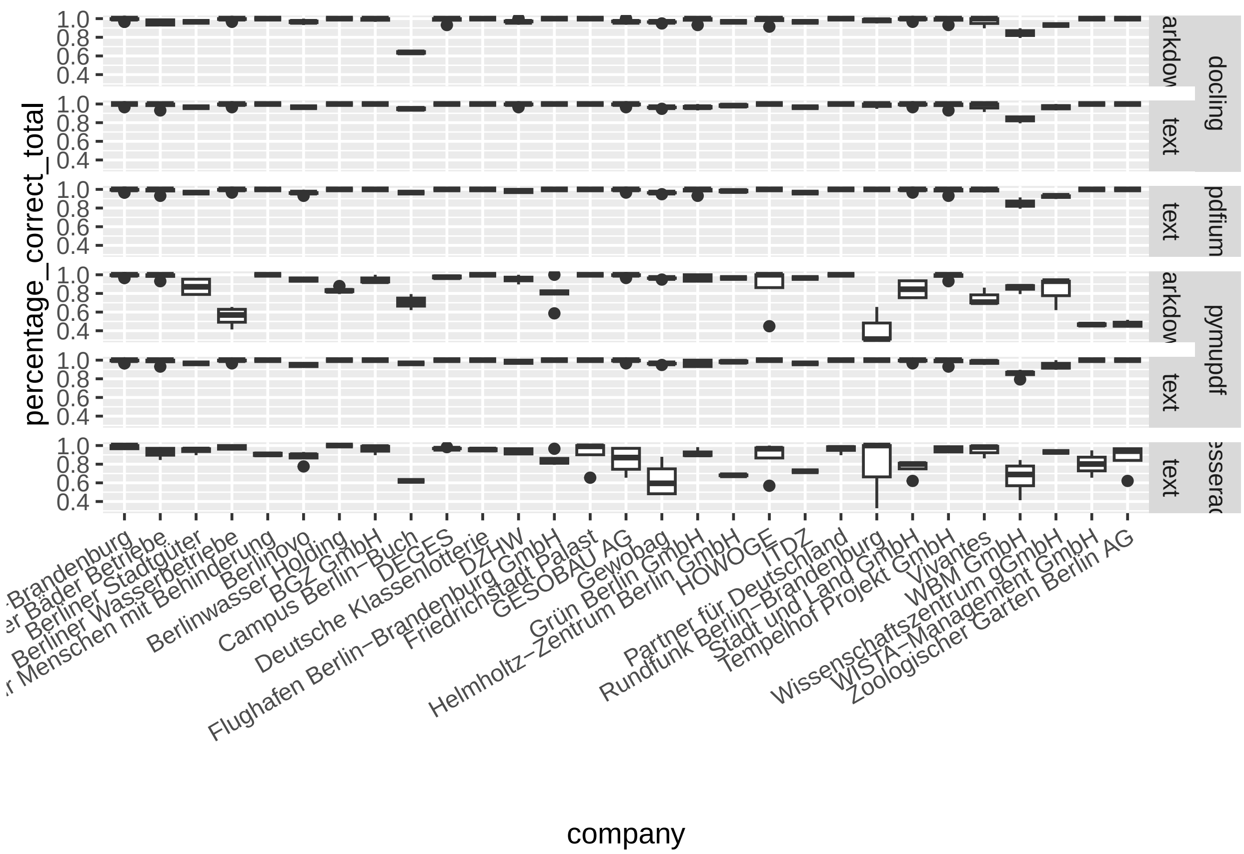
G.2.2 Real tables

G.2.2.1 Examples from same company

G.2.2.2 OpenAI models

G.2.2.3 Comparing input formats and text extration libraries

```
df_qwen235 %>% group_by(model, method, extractor, input_format) %>%
  mutate(mean_total = mean(percentage_correct_total)) %>%
  group_by(model, extractor, input_format, filepath) %>%
  slice_max(n = 1, mean_total, with_ties = FALSE) %>%
  # select(model, method, extractor, input_format, mean_total, filepath) %>%
  ggplot() +
  geom_boxplot(aes(x = company, y = percentage_correct_total)) +
  facet_nested(extractor+input_format~.) +
  scale_x_discrete(guide = guide_axis(angle = 30))
```



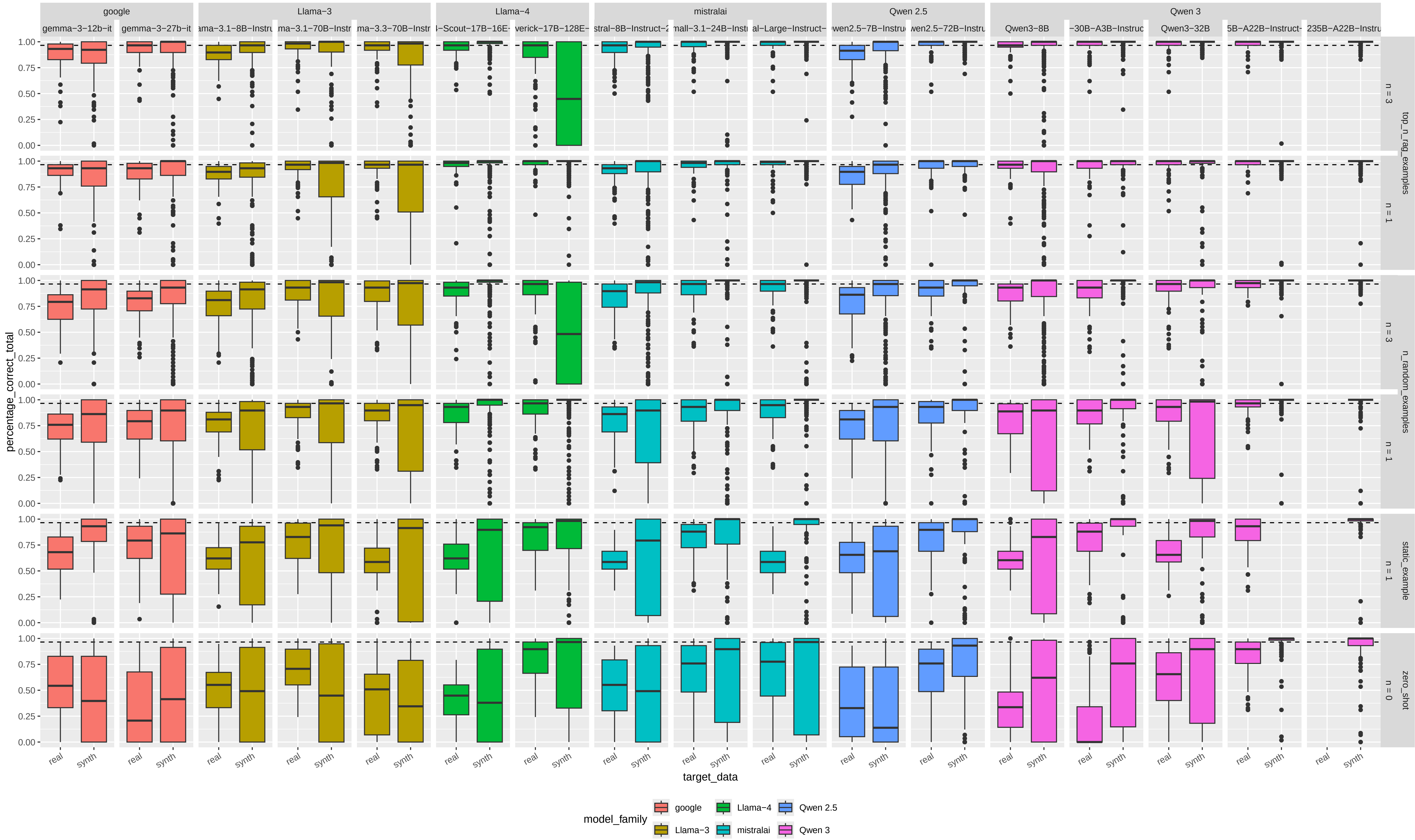


Figure G.9: Comparing the table extraction performance among real and synthetic Aktiva tables

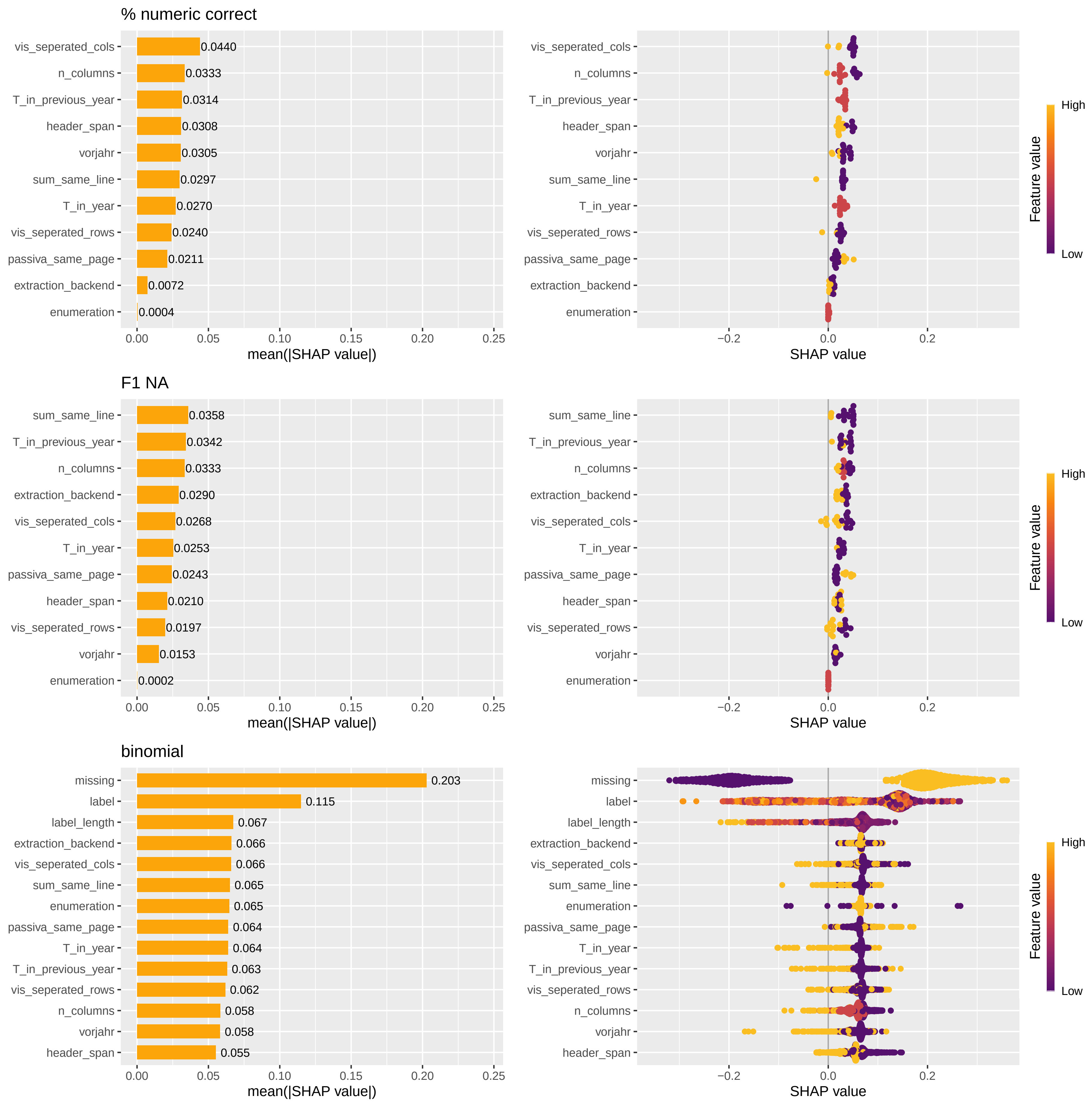


Figure G.10: Mean absolute SHAP values and beeswarm plots for real table extraction with regular expression approach

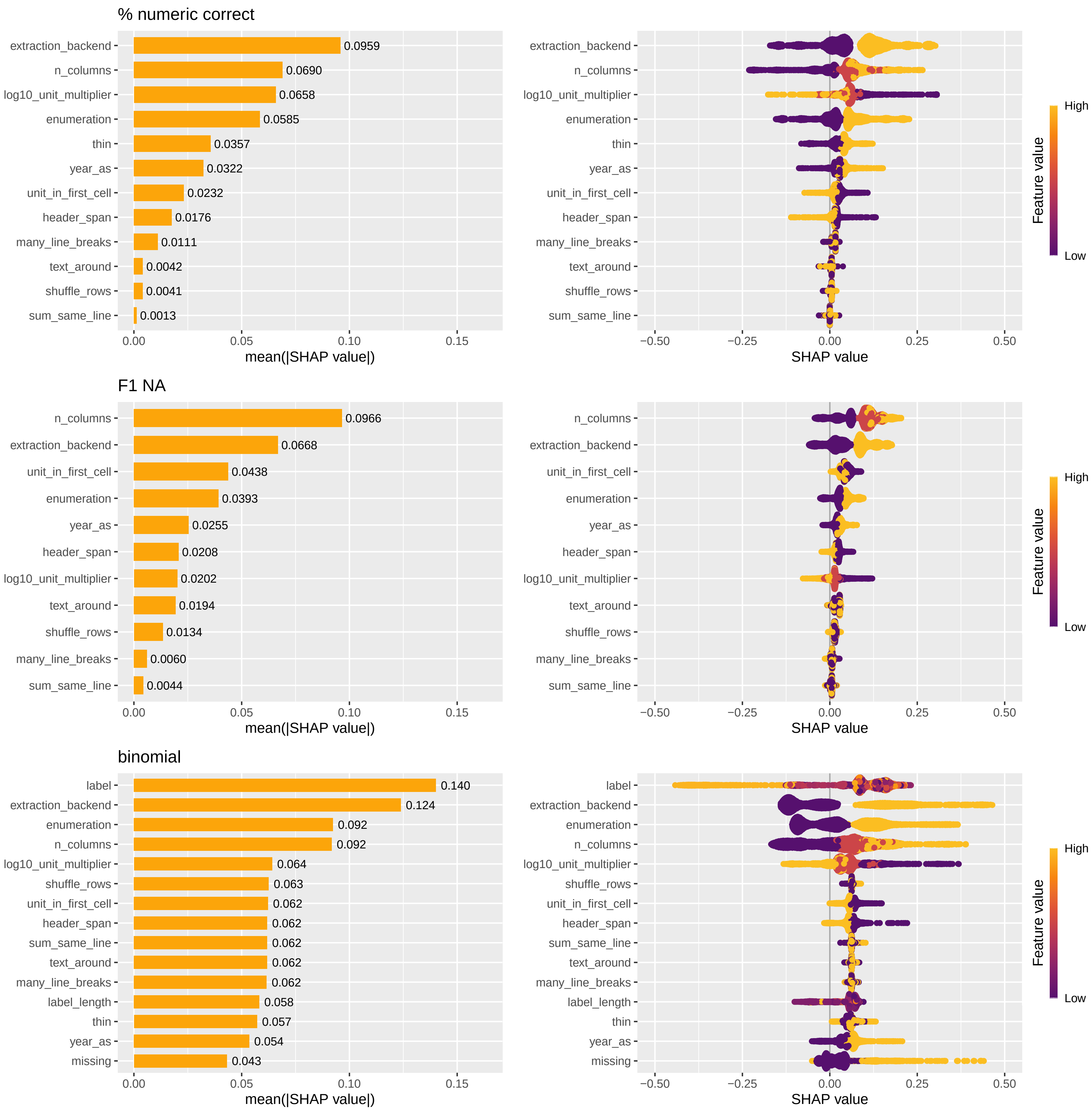


Figure G.11: Mean absolute SHAP values and beeswarm plots for synth table extraction with regular expression approach

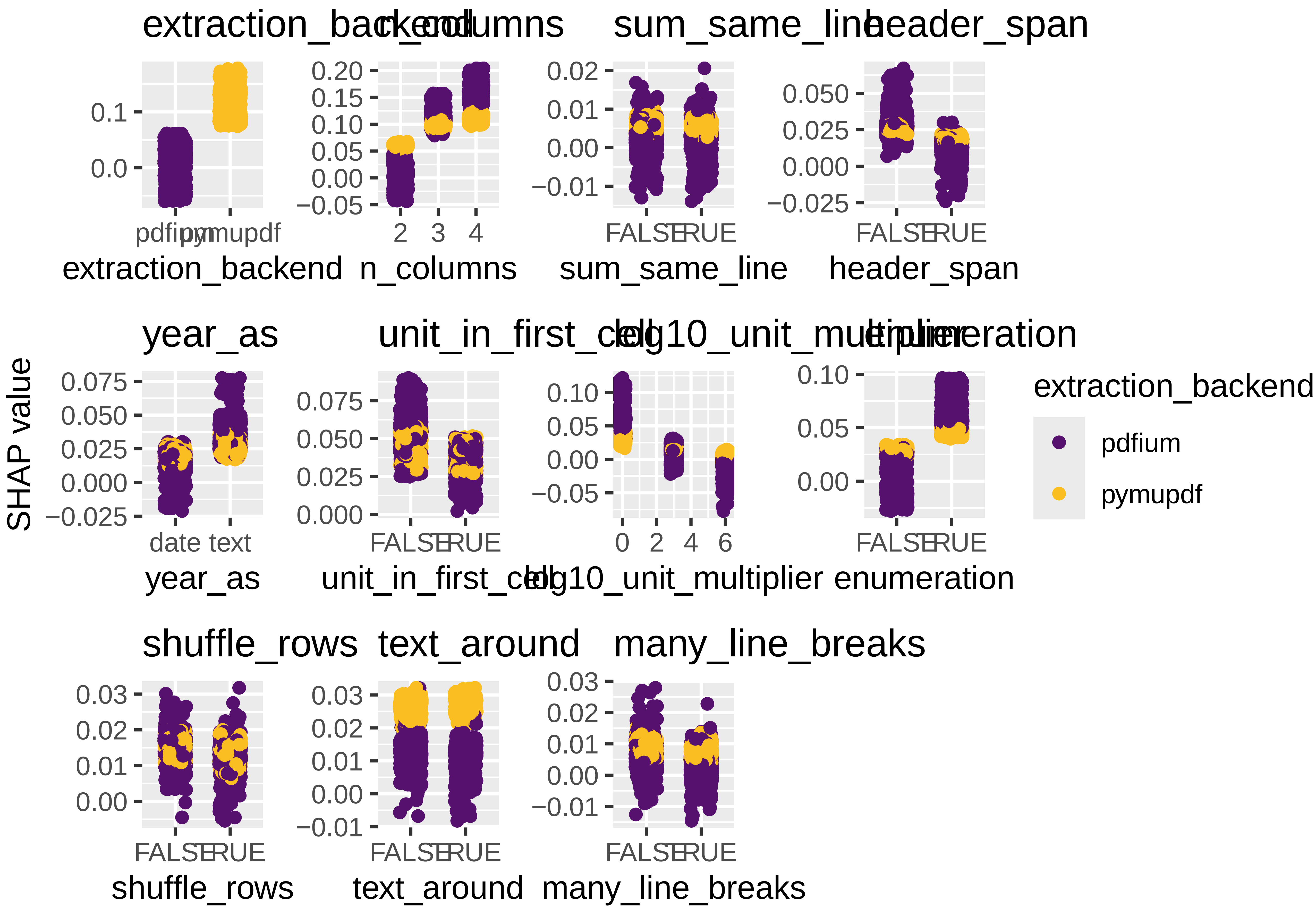


Figure G.12: Showing the interactions of the extraction backend pdfium with the table characteristics for F1 score.

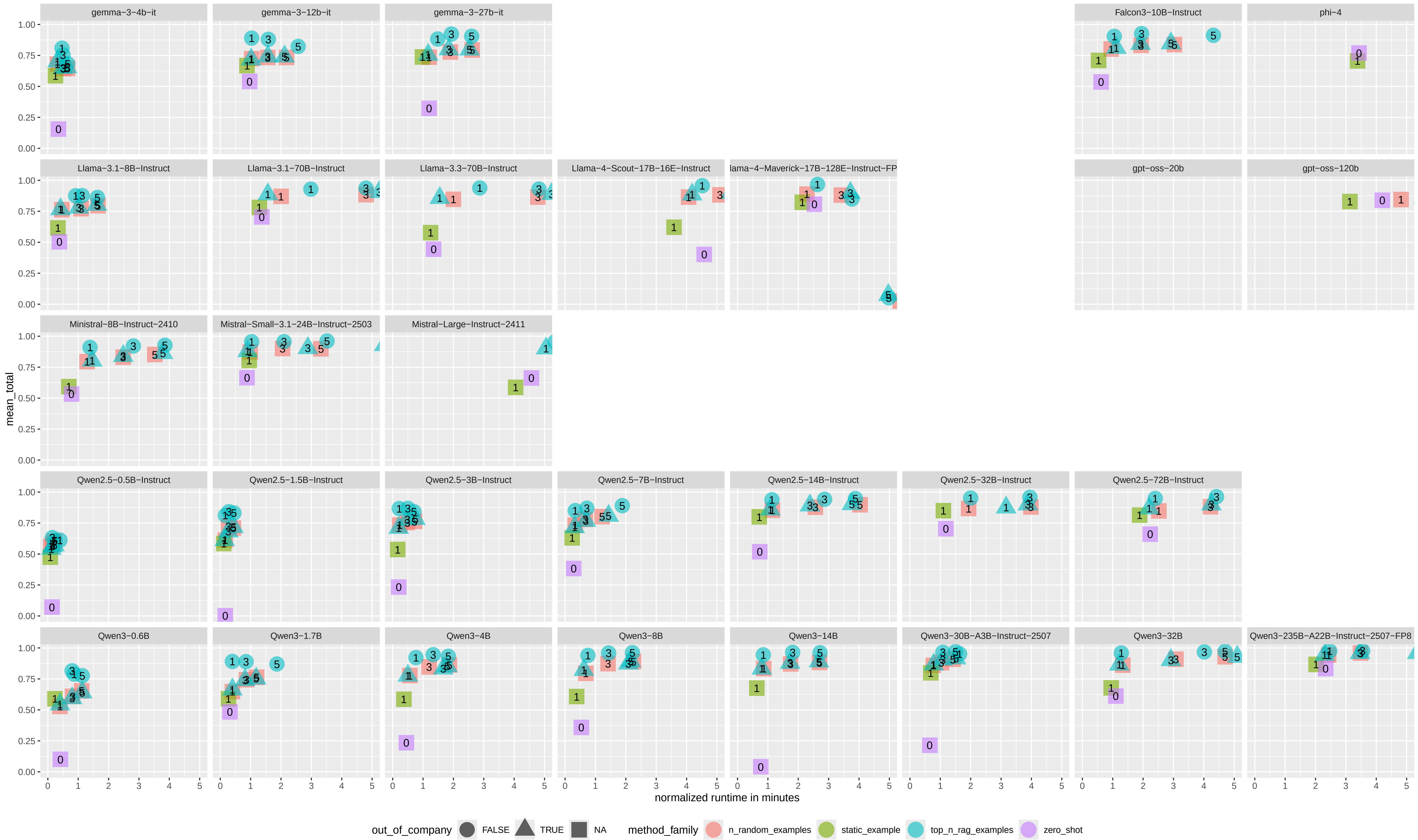


Figure G.13: Comparing percentage of correct predictions total over the normalized runtime. The normalized runtime is given in minutes of processing on a single B200. The time to load the model into the VRAM is excluded. Focussing on small models showing only 5 minutes of runtime.

Figure G.14: Comparing percentage of correct predictions total over the normalized runtime. The normalized runtime is given in minutes of processing on a single B200. The time to load the model into the VRAM is excluded. Showing the full runtime range.

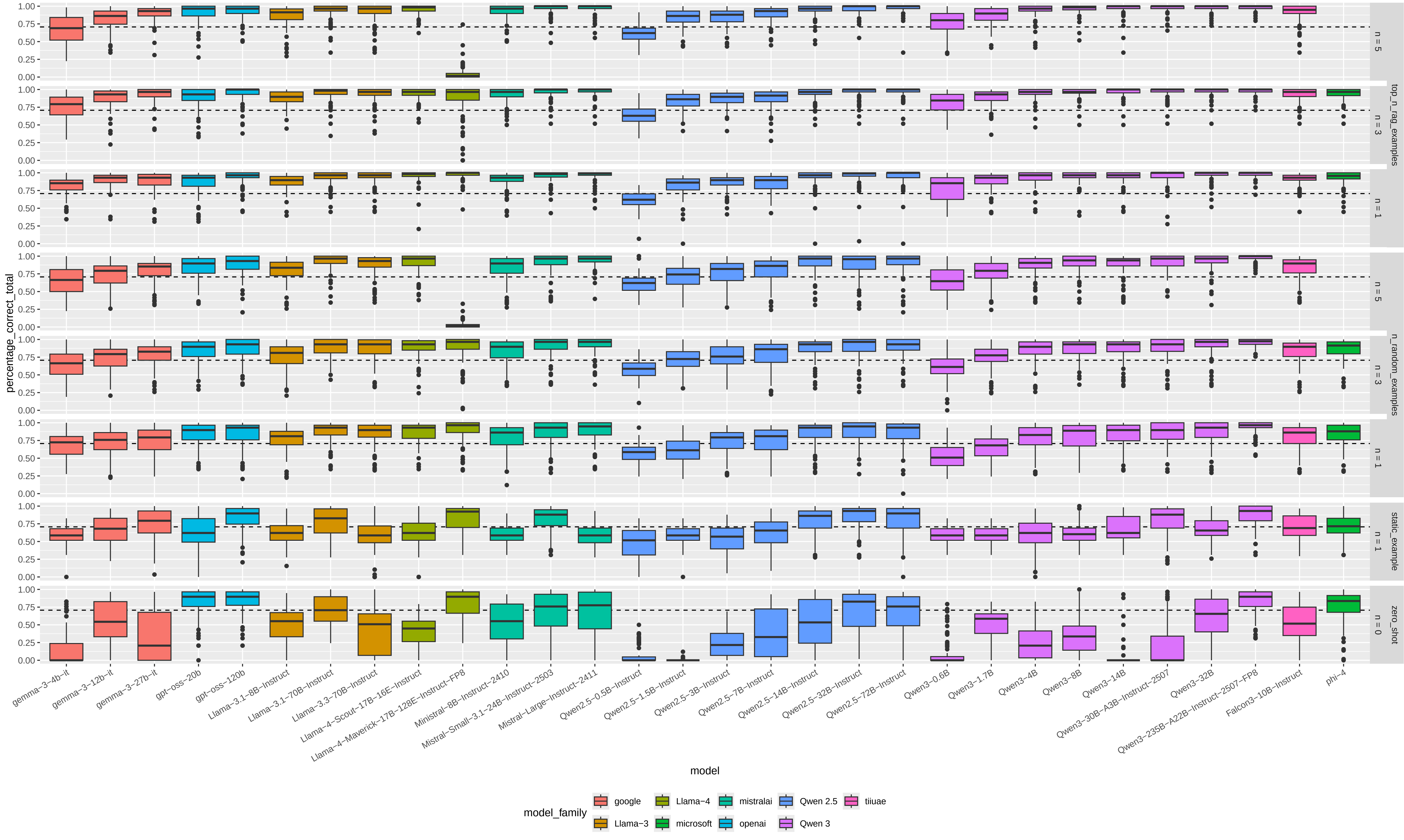


Figure G.15: Percentage of correct extracted or as missing categorized values for table extraction task on real Aktiva tables

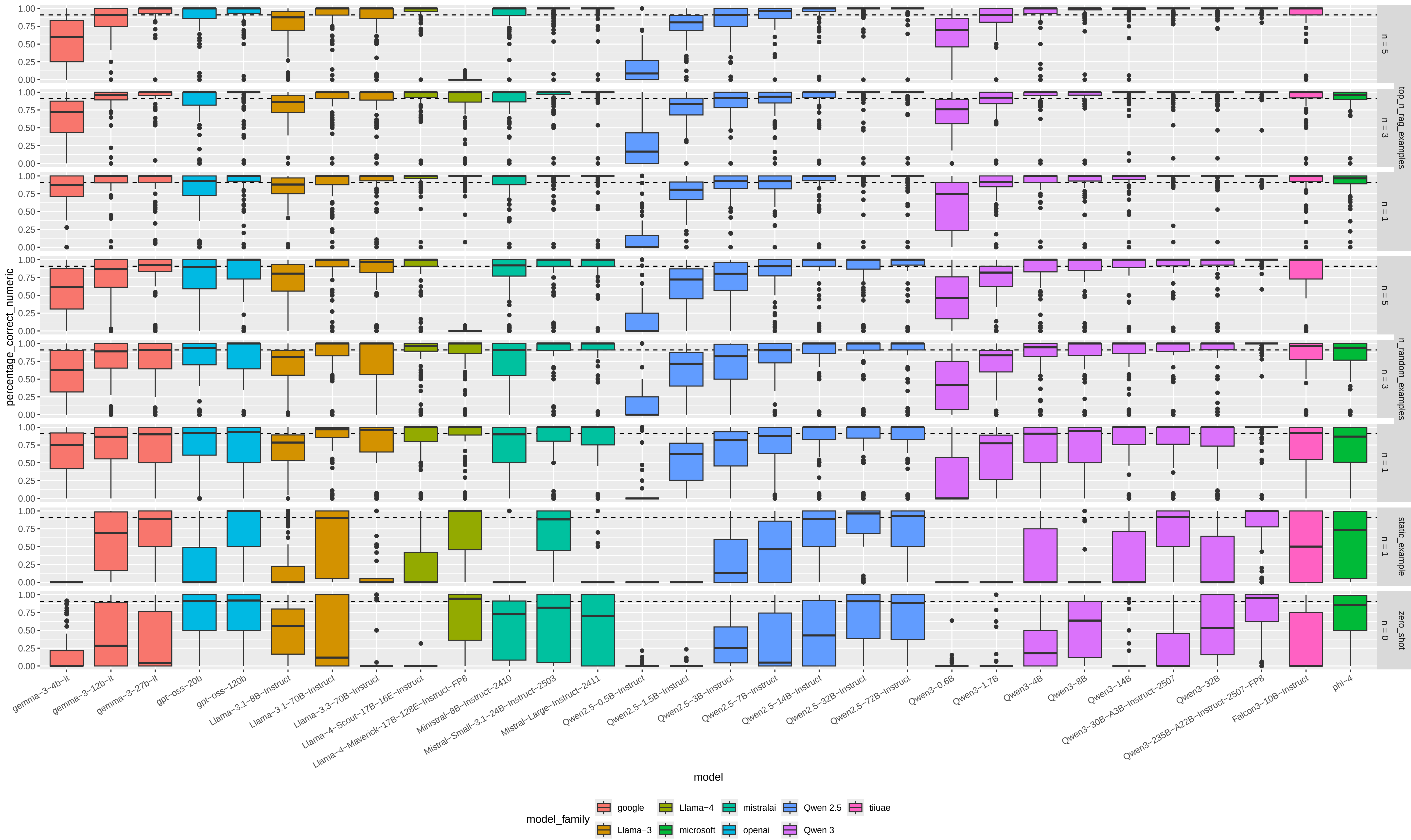


Figure G.16: Percentage of correct extracted numeric values for table extraction task on real Aktiva tables

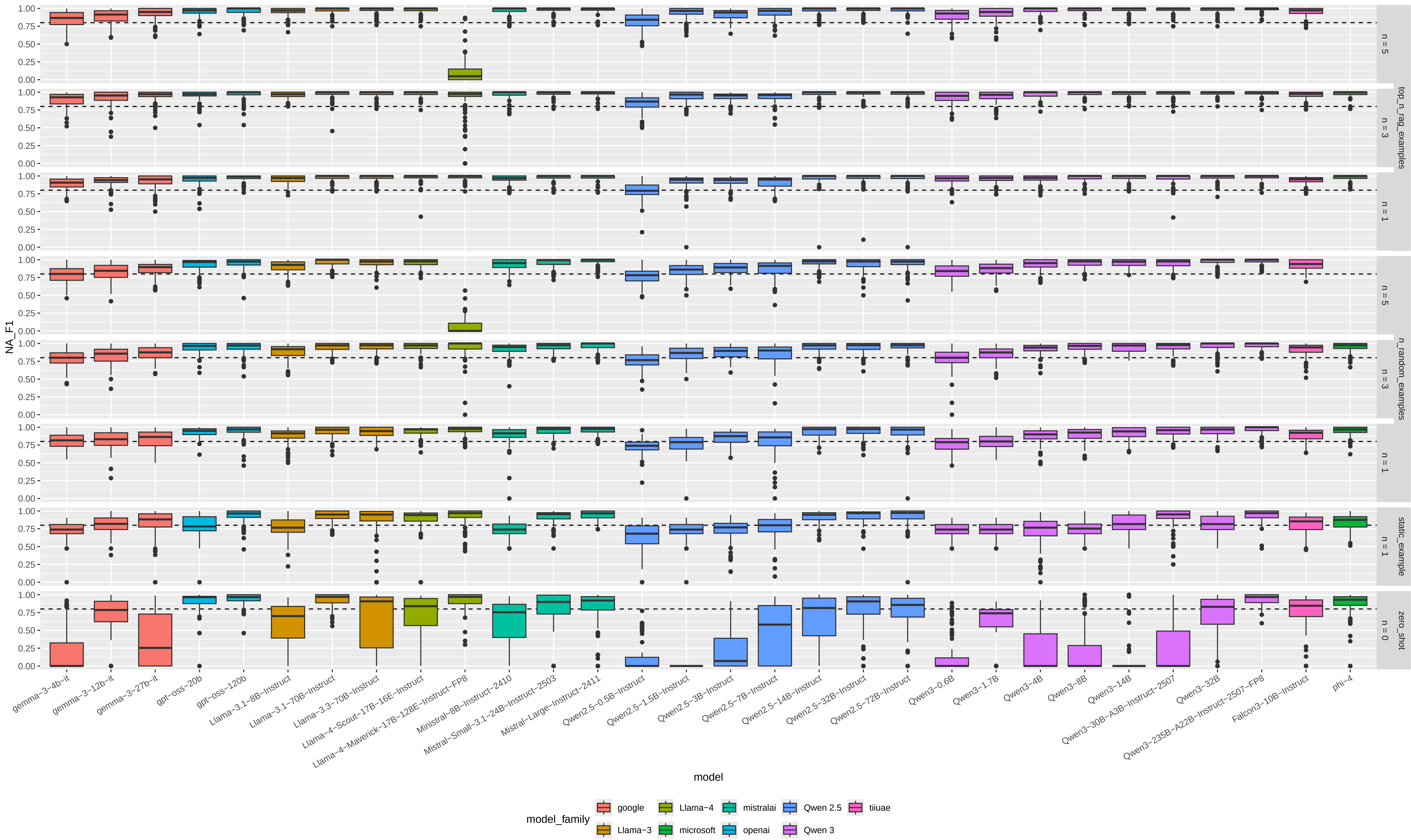


Figure G.17: F1 score for the missing classification if a value is missing for table extraction task on real Aktiva tables

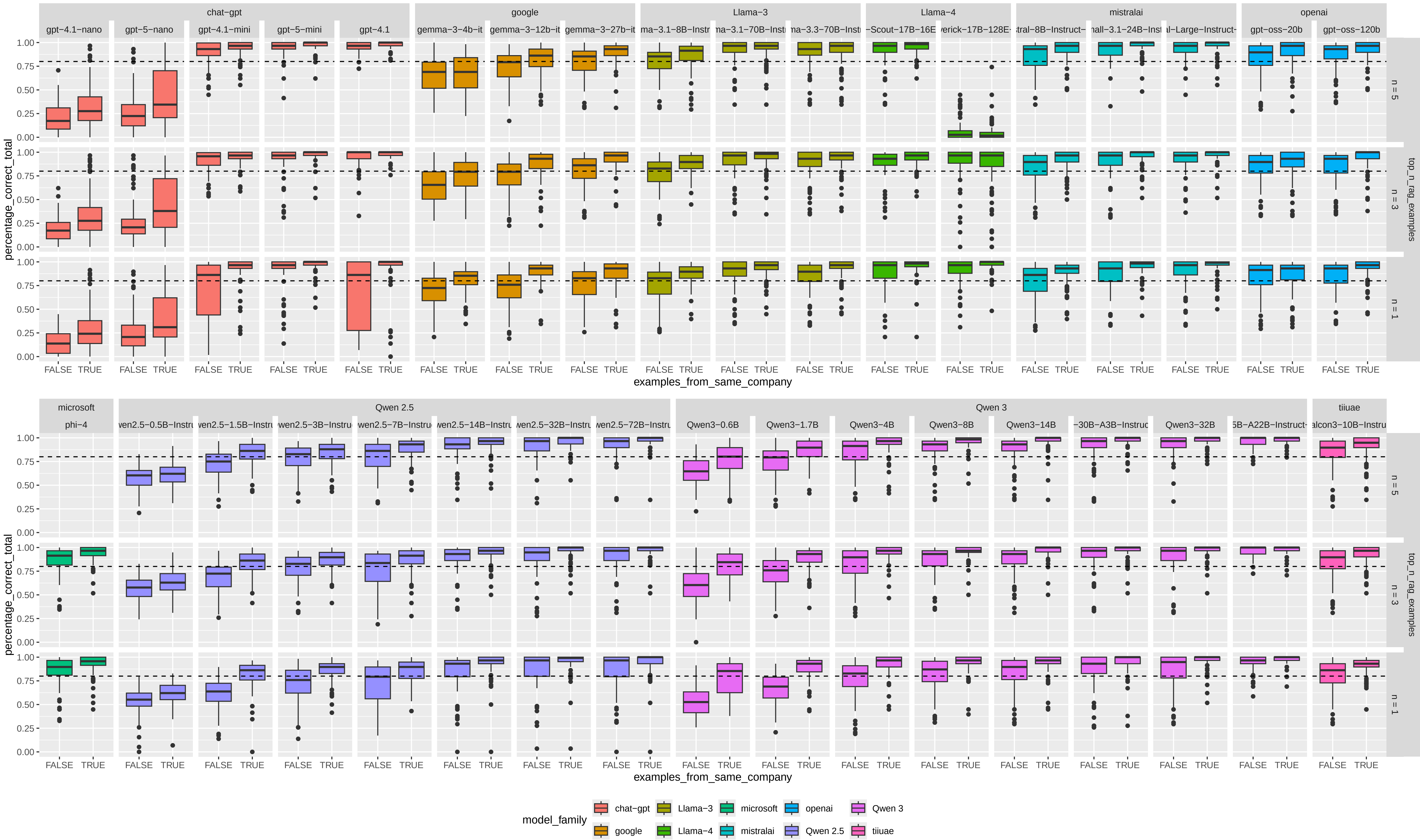


Figure G.18: Comparing the overall extraction performance depending on the condition if examples from the same company can be used.

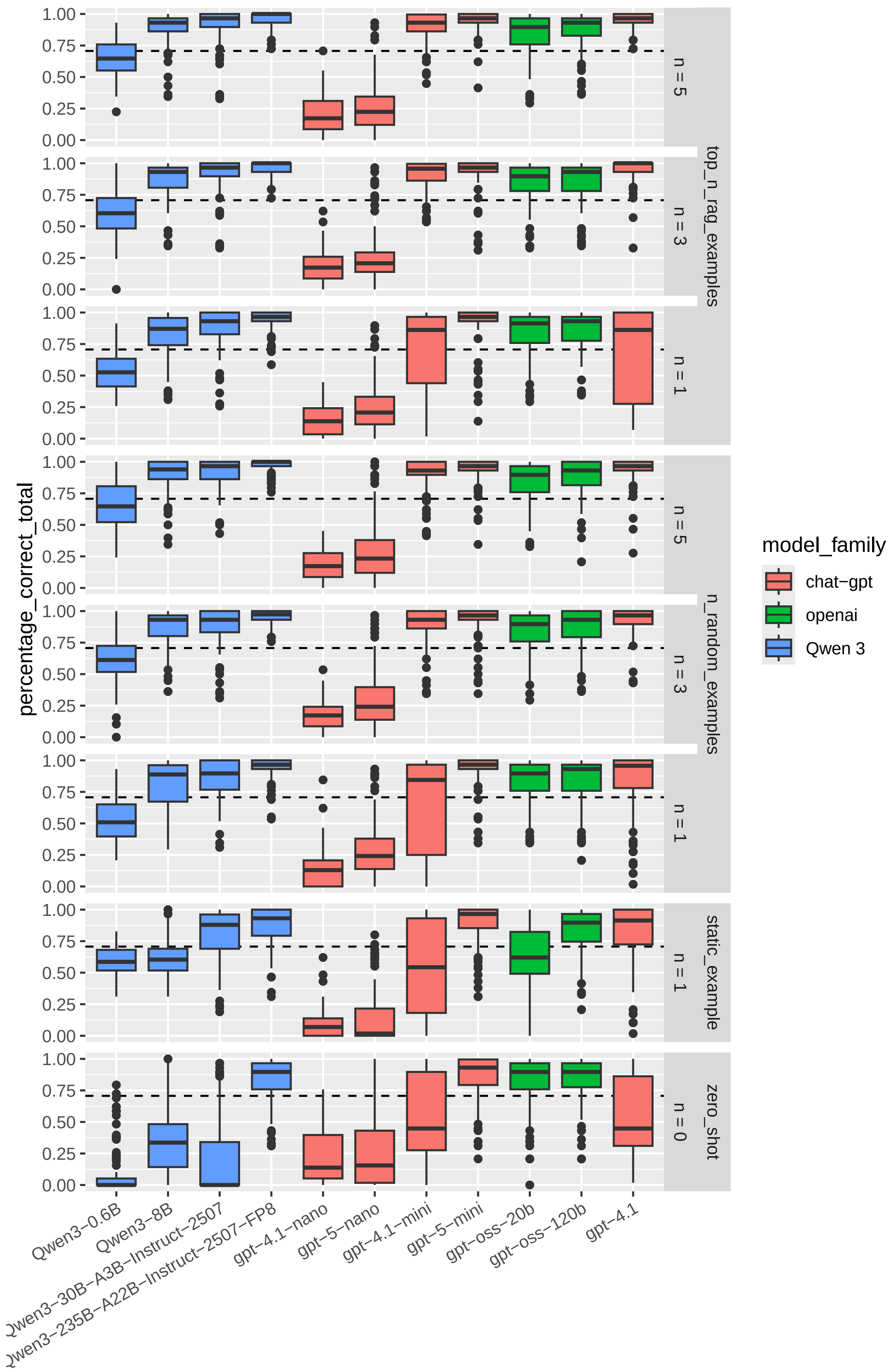


Figure G.19: Comparing the percentage of correct predictions overall for OpenAi's LLMs with some Qwen 3 models

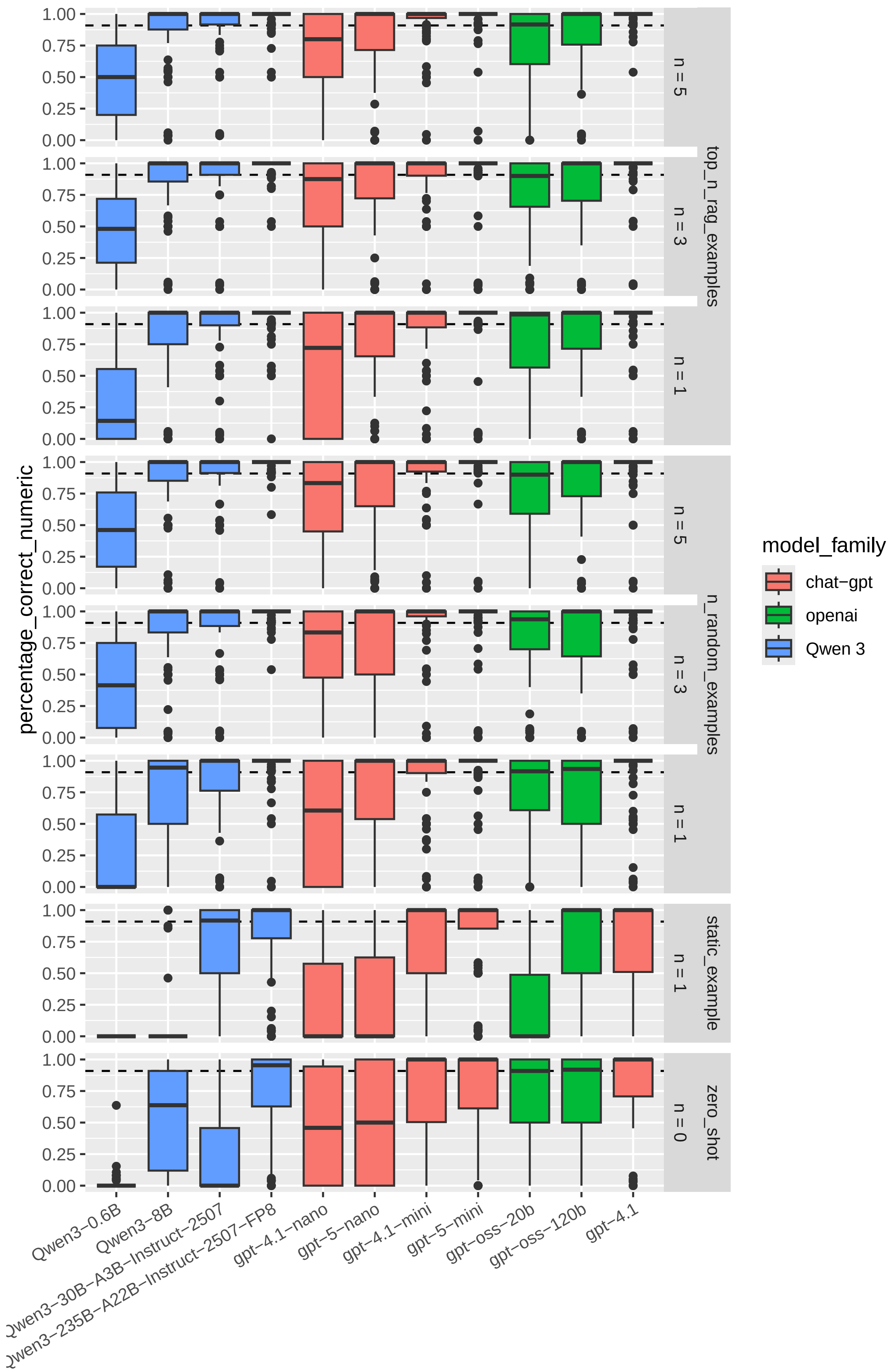


Figure G.20: Comparing the percentage of correct numeric predictions for OpenAi's LLMs with some Qwen 3 models

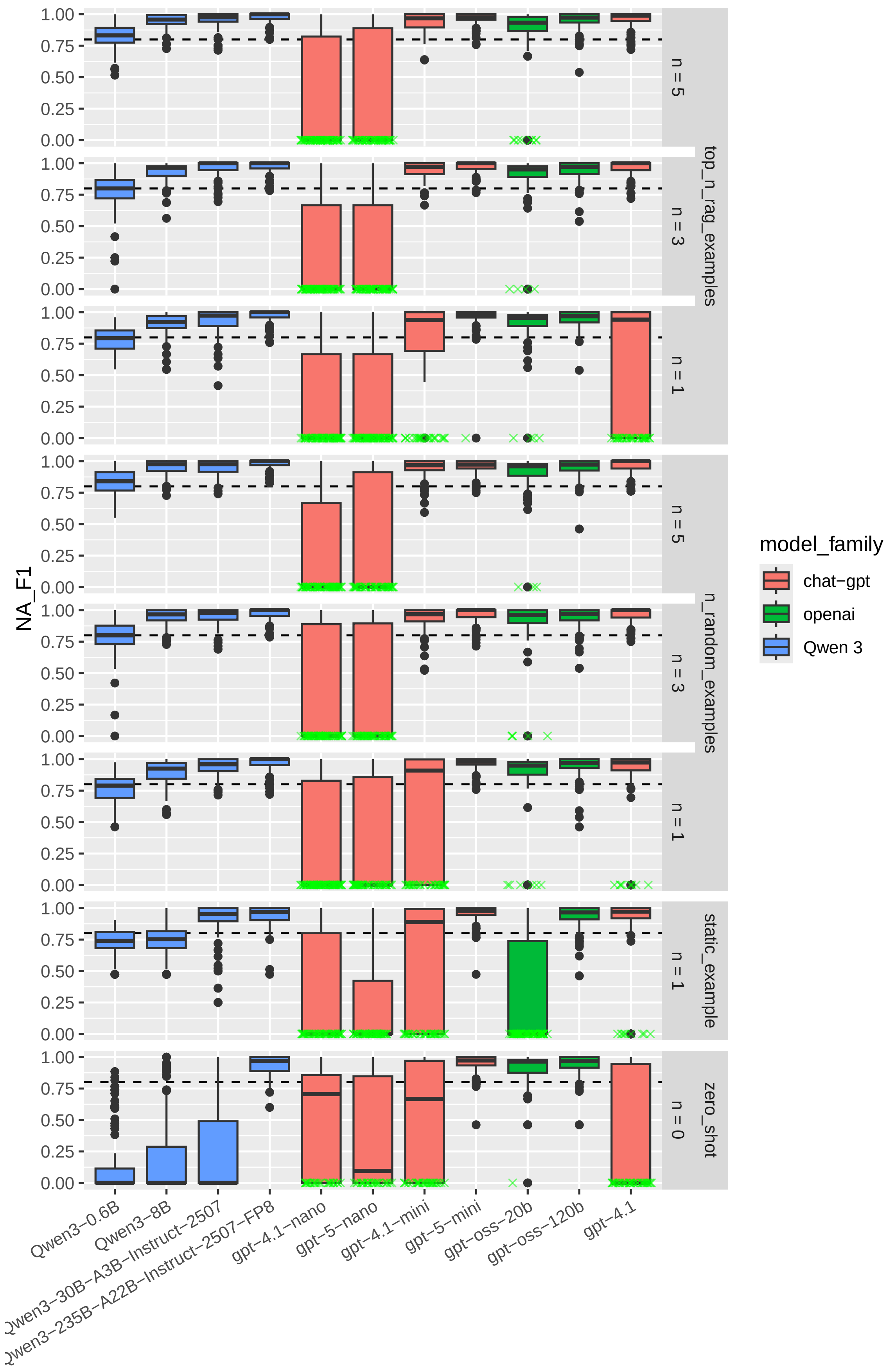


Figure G.21: Comparing the F1 score for predicting the missingness of a value for OpenAi's LLMs with some Qwen 3 models. The green crosses indicate results where a model has predicted only numeric values even though there have been missing values.

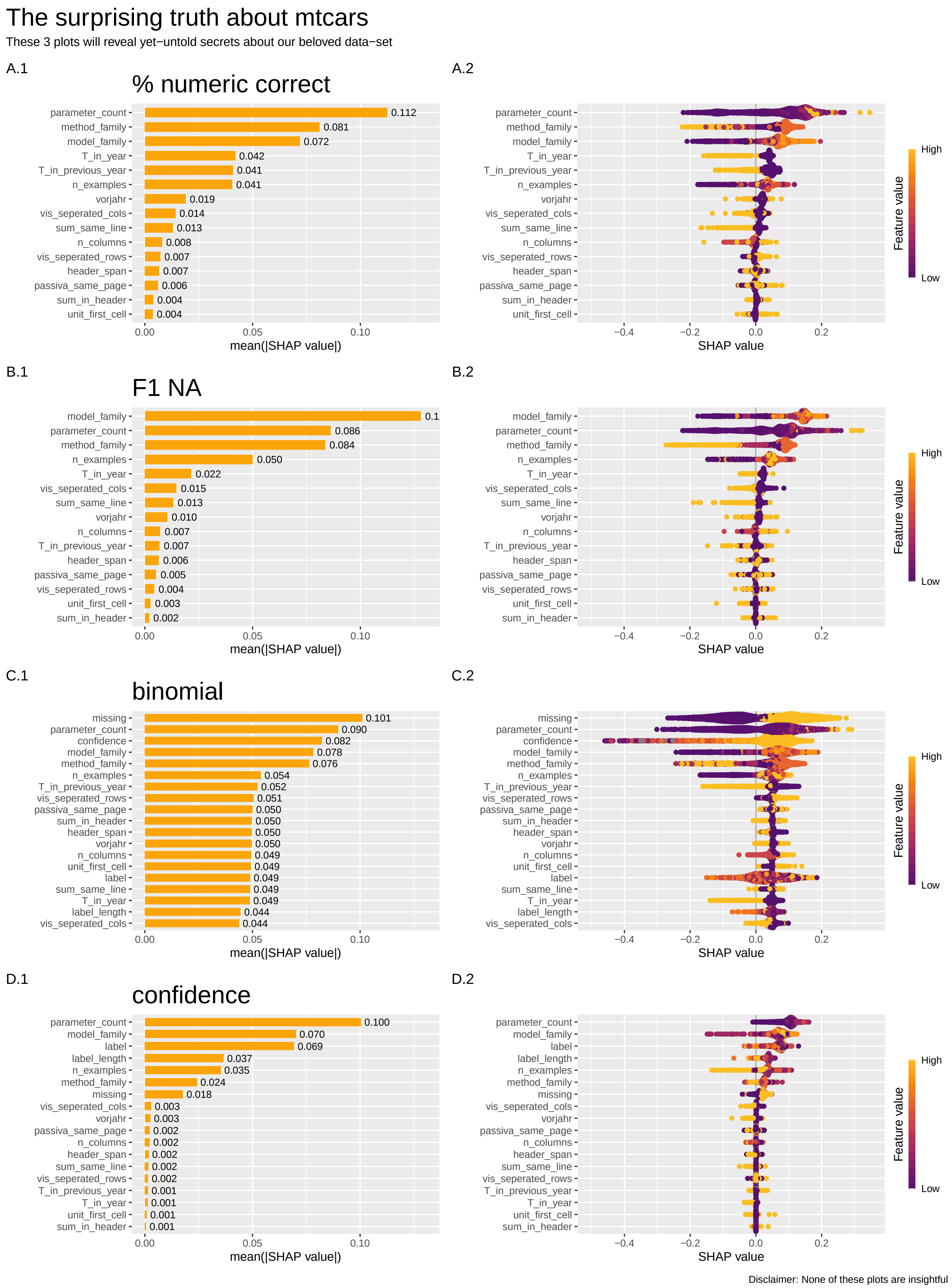


Figure G.22: Mean absolute SHAP values and beeswarm plots for real table extraction with LLMs

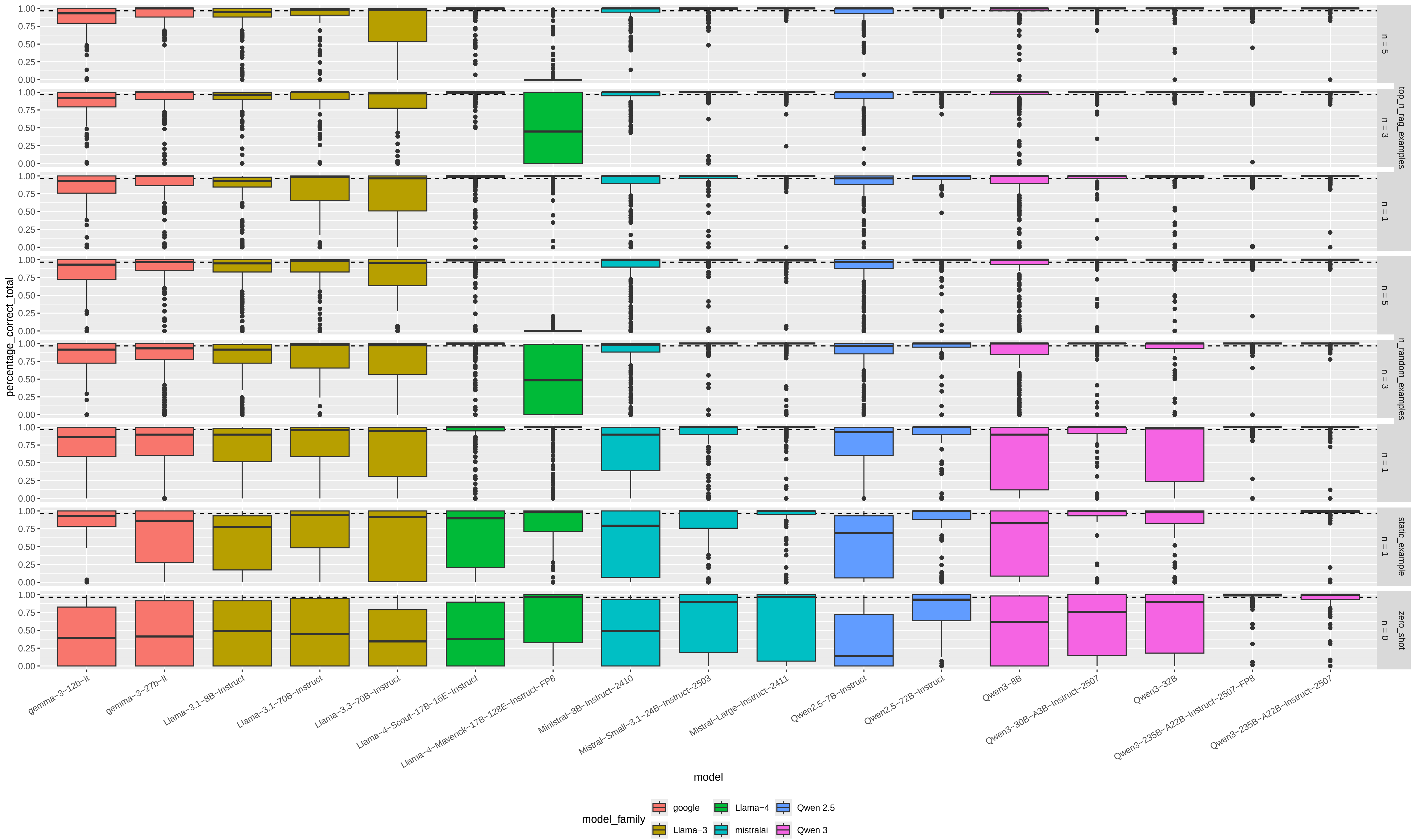


Figure G.23: Percentage of correct extracted or as missing categorized values for table extraction task on synthetic Aktiva tables

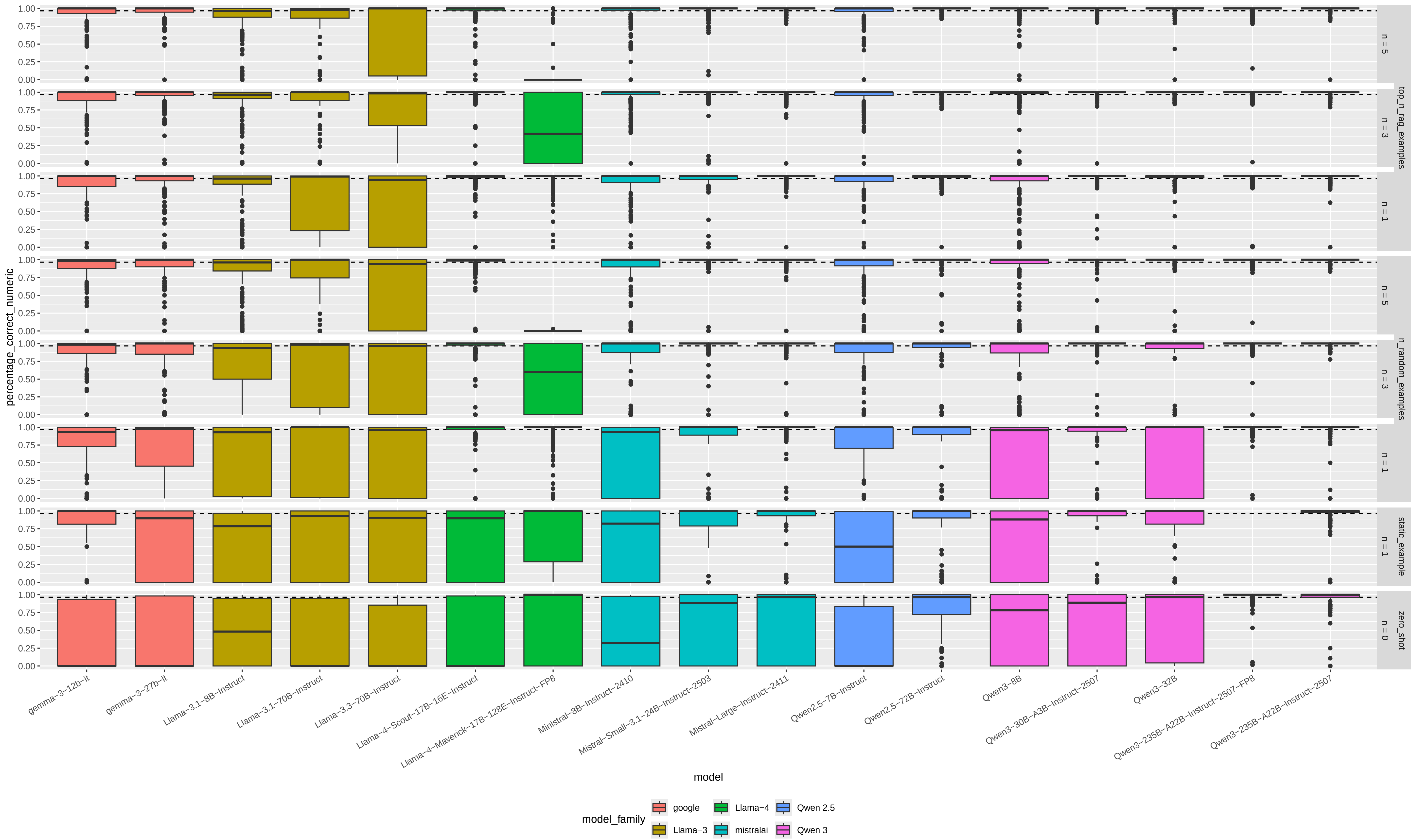


Figure G.24: Percentage of correct extracted numeric values for table extraction task on synthetic Aktiva tables

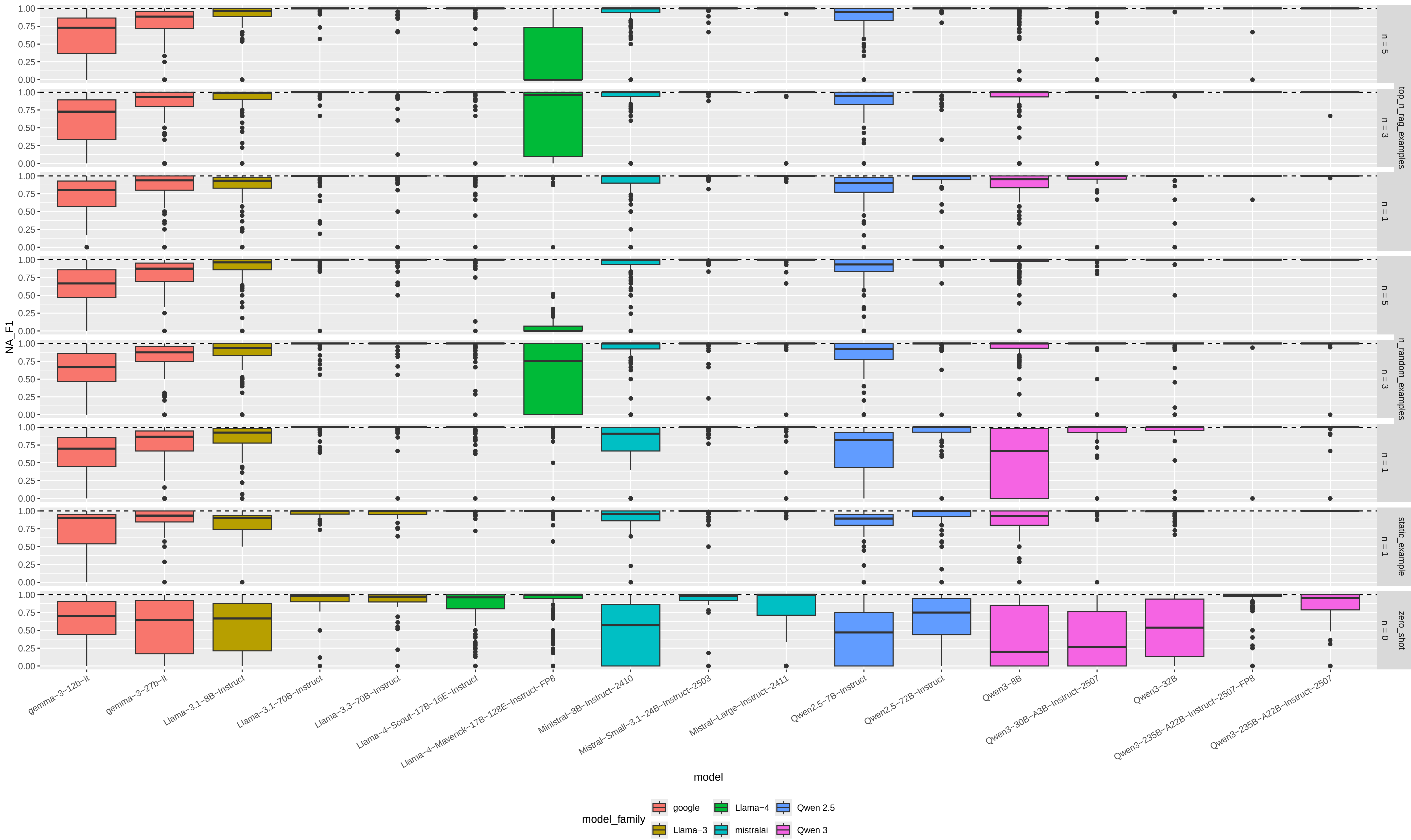


Figure G.25: F1 score for the missing classification if a value is missing for table extraction task on synthetic Aktiva tables

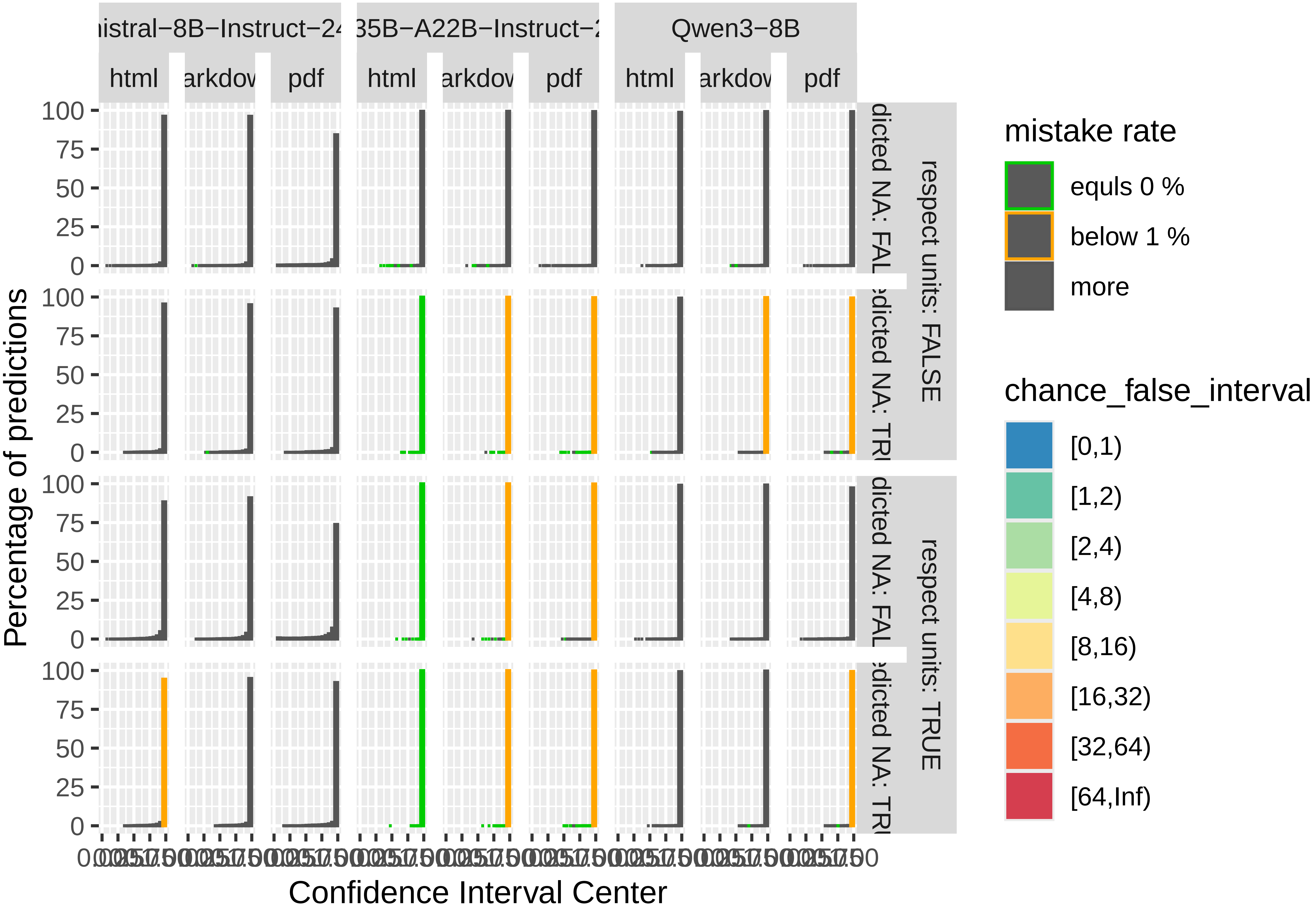
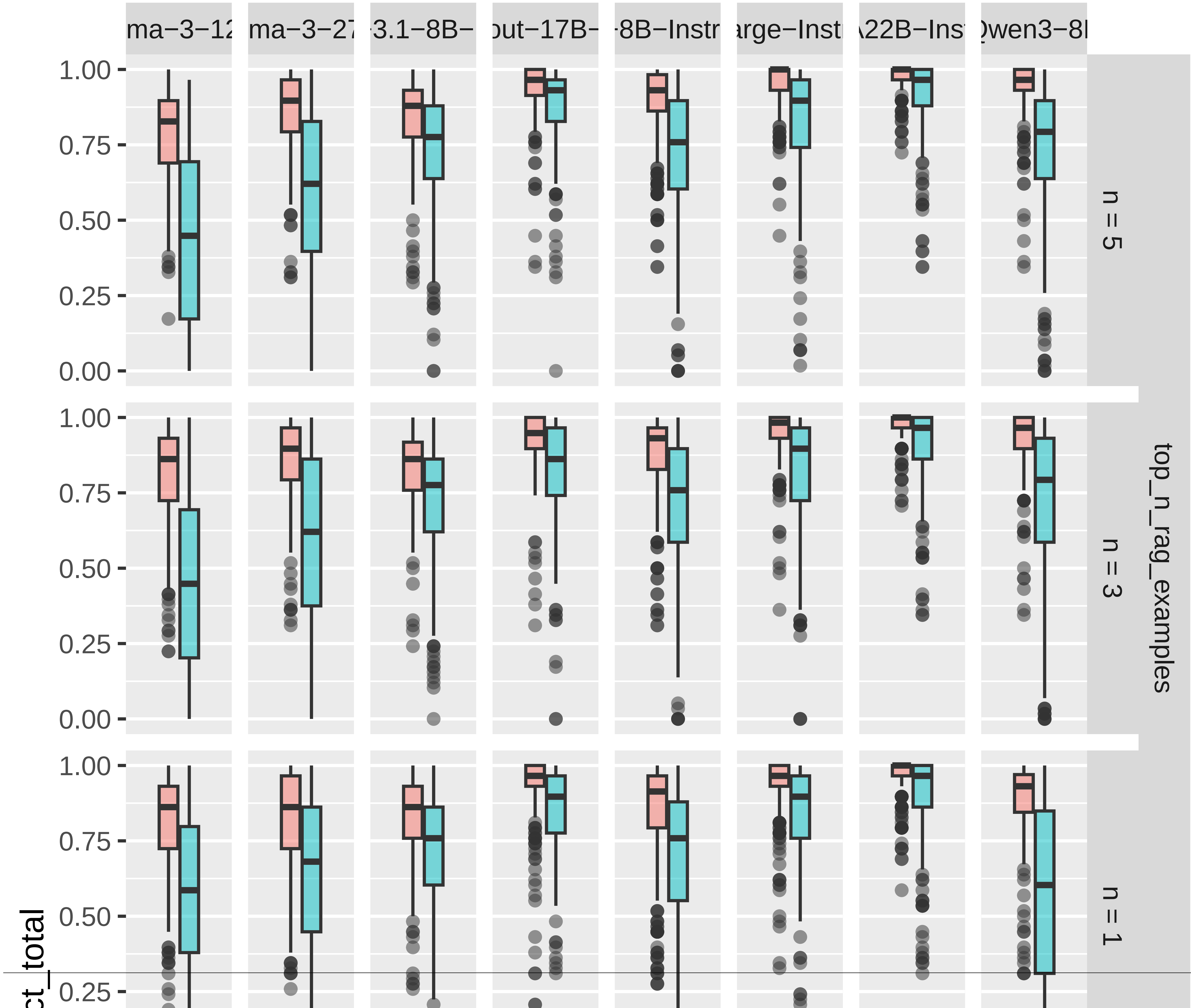
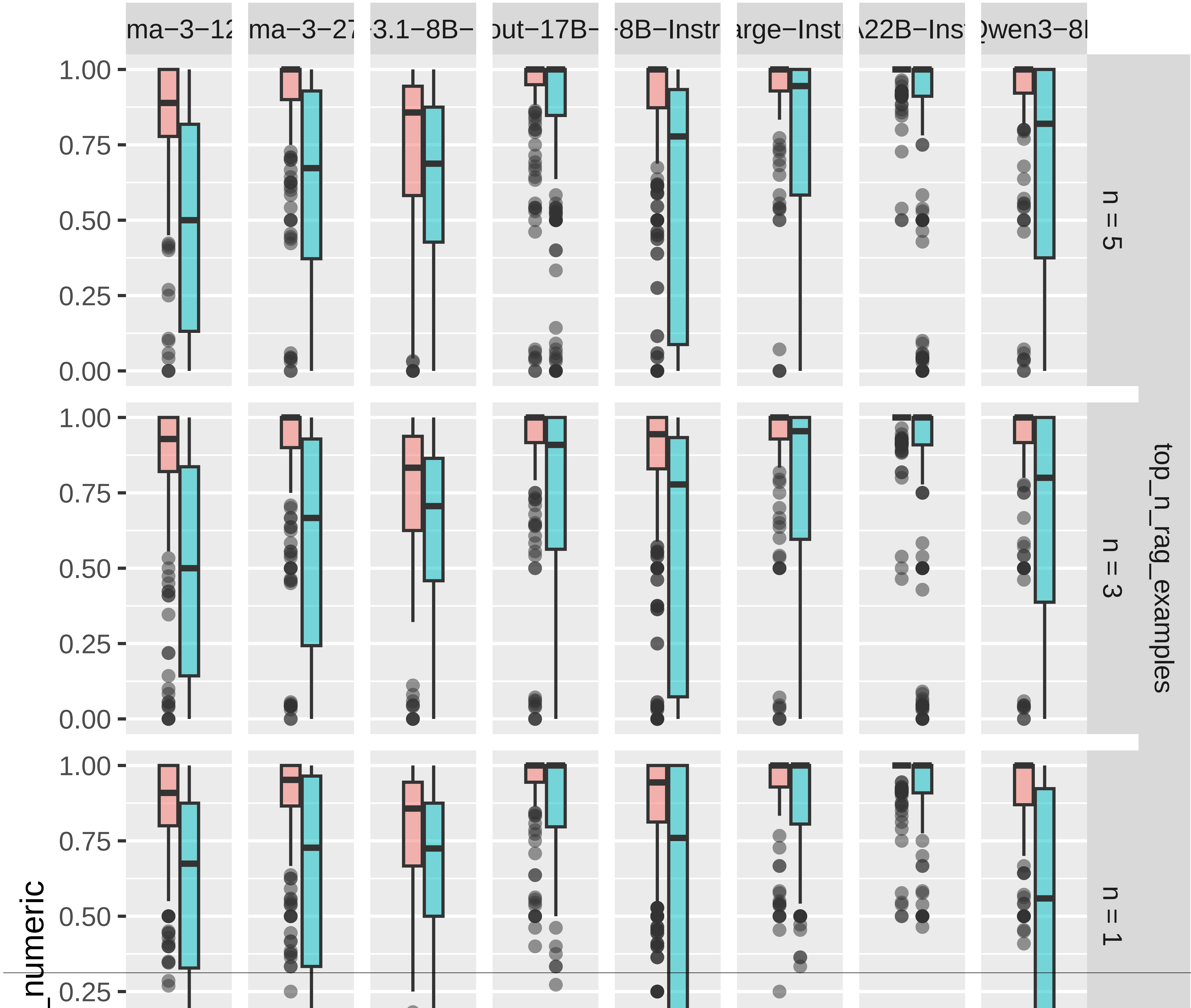
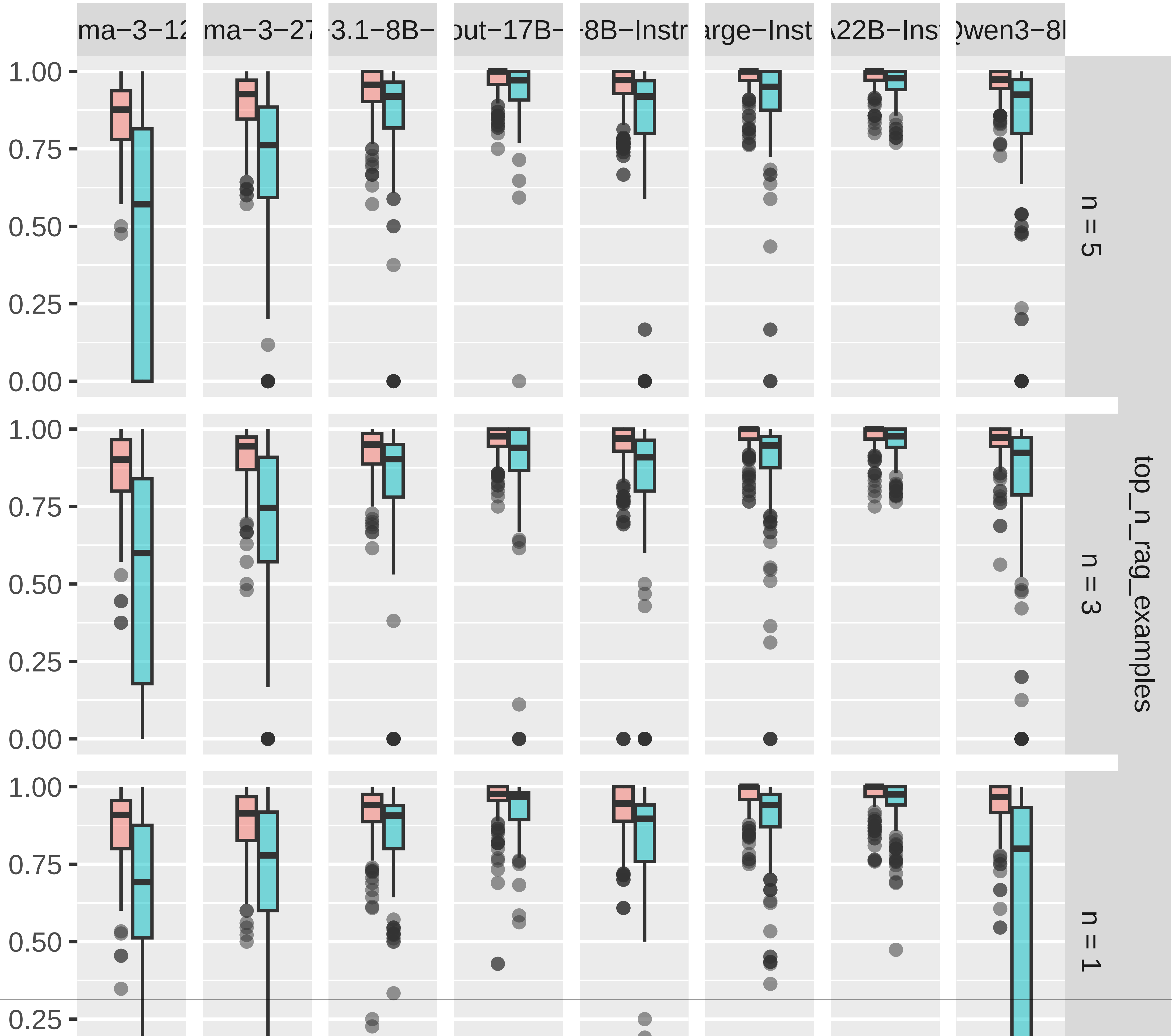
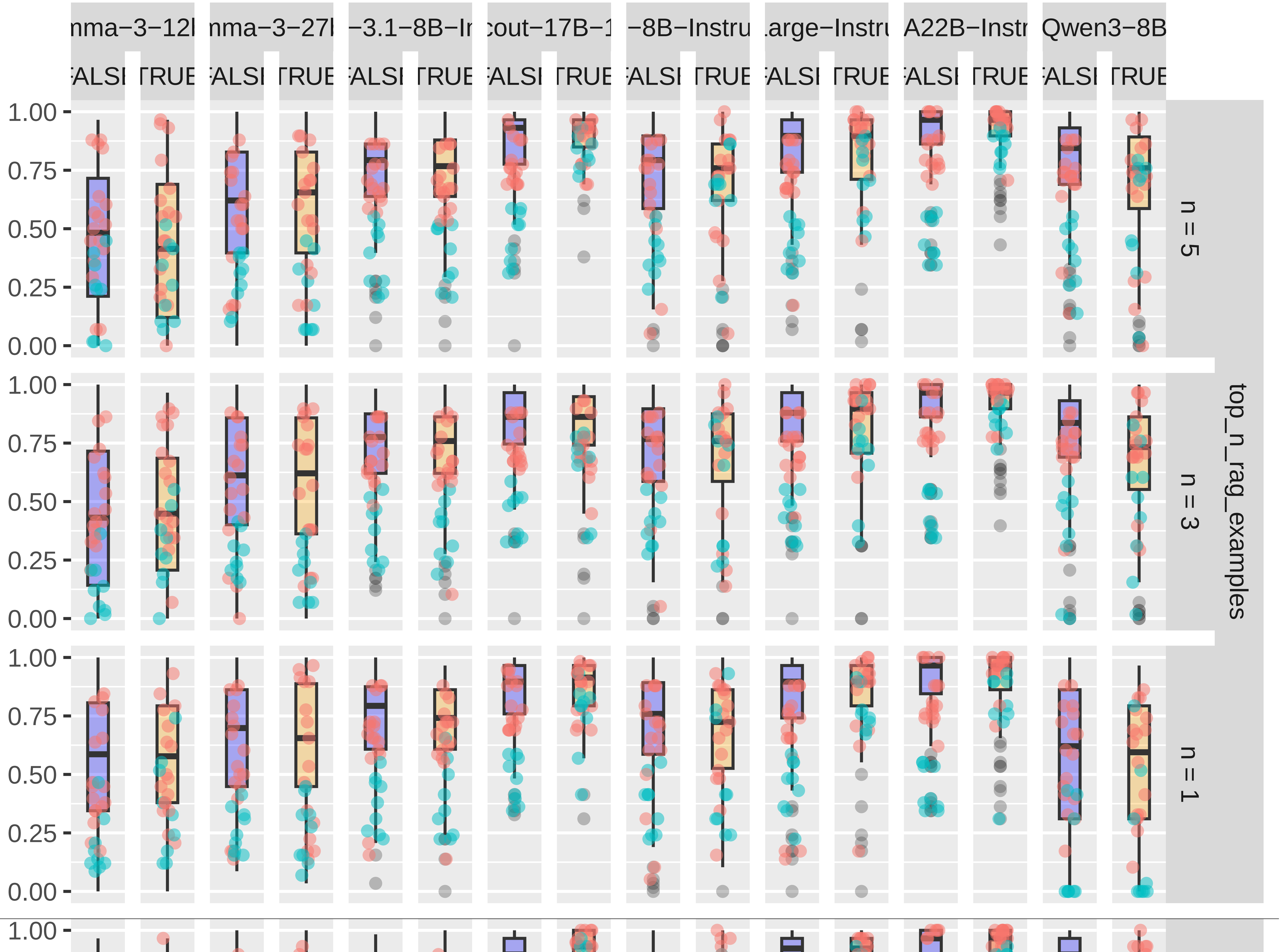


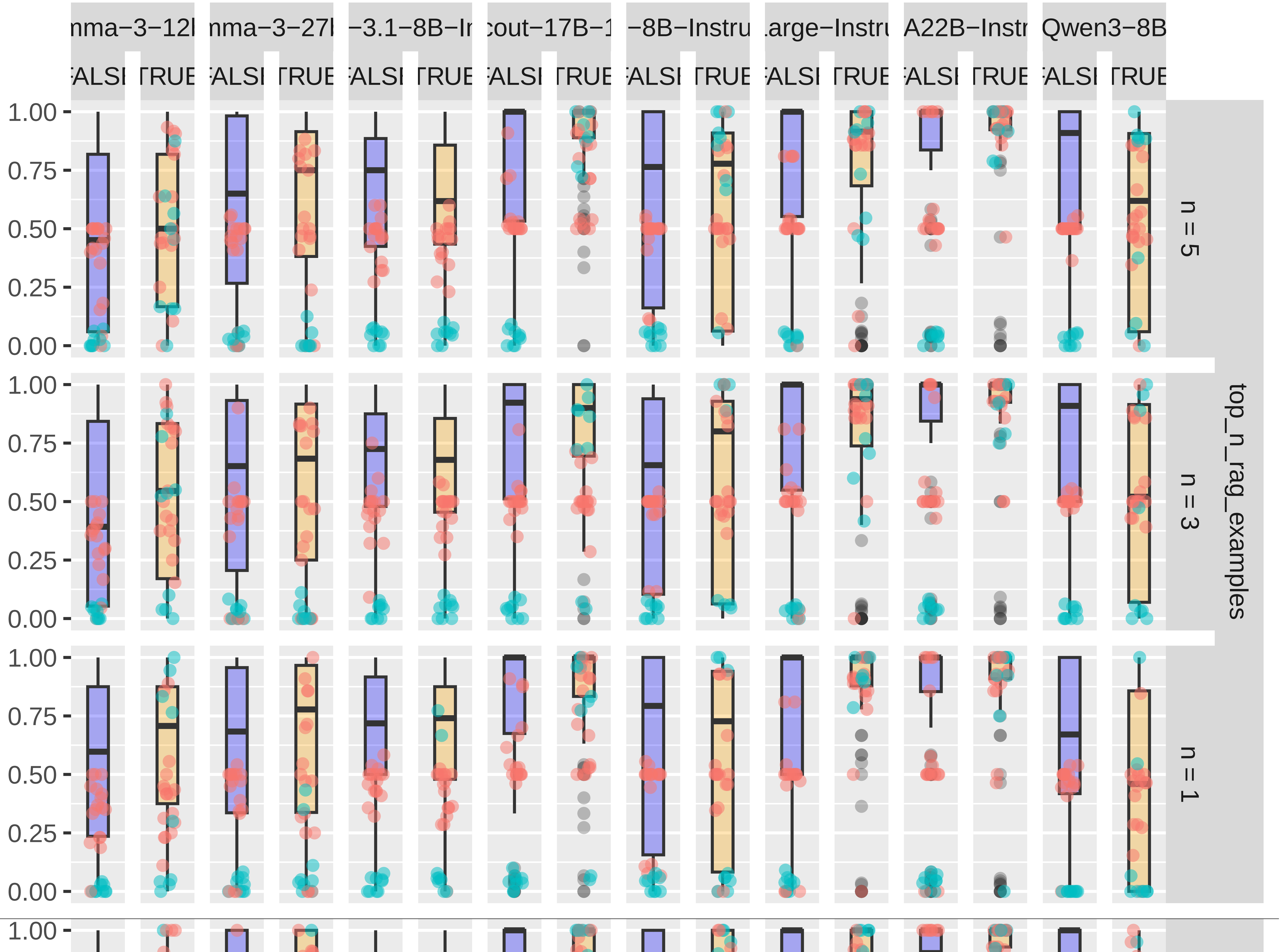
Figure G.26: Estimating the relative frequency to find a wrong extraction result over different confidence intervals for predictions for the synthetic table extraction task. Additionally grouped by input format.

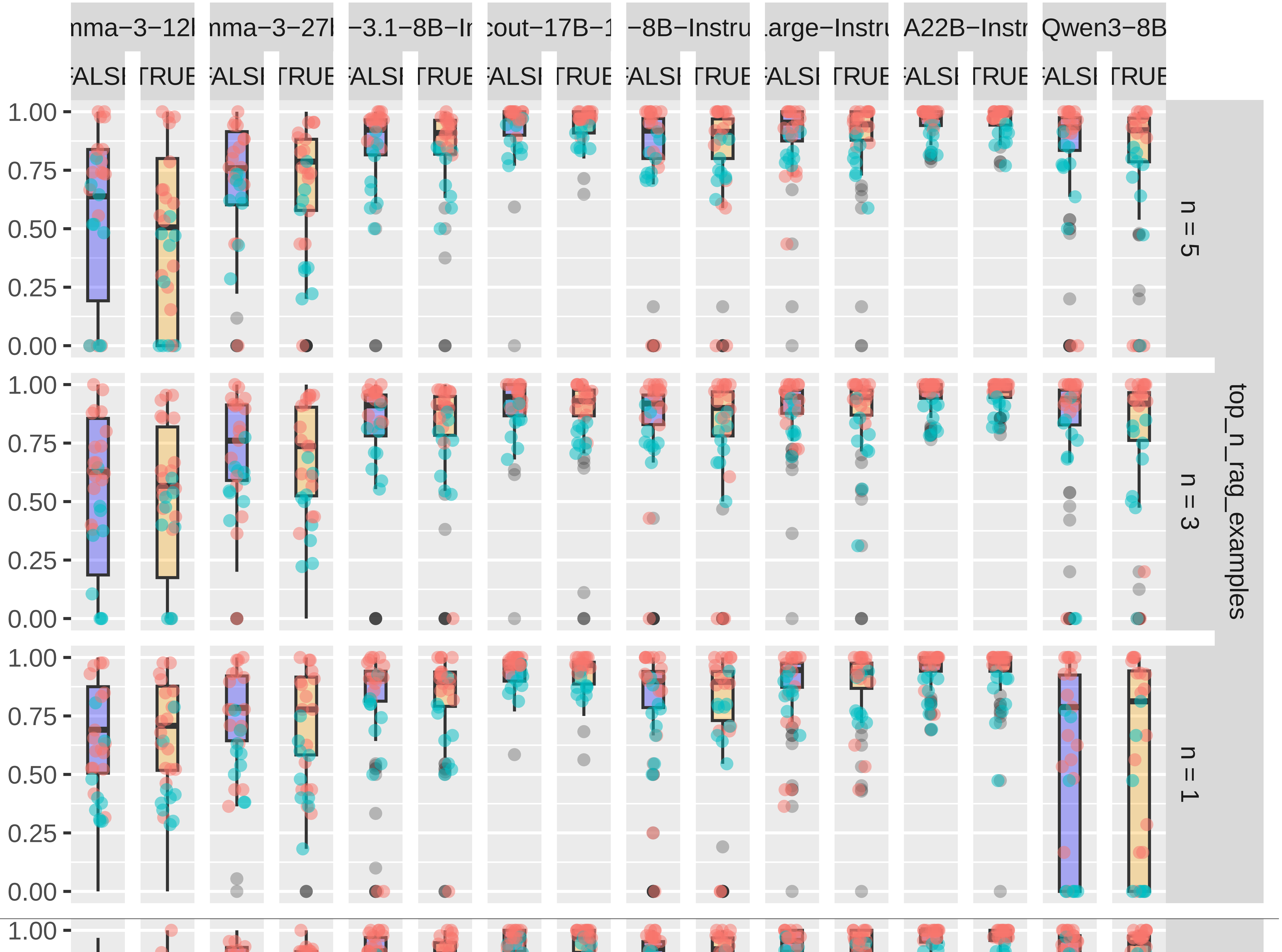

ma−3−12
ma−3−27
3.1−8B−
out−17B−
8B−Instr
arge−Inst
22B−Ins
Qwen3−8
1.00
0.75
0.50
0.25
0.00
n = 5
n = 3
n = 1
top_n_rag_examples
ct_total


ma−3−12
ma−3−27
3.1−8B−
out−17B−
8B−Instr
arge−Inst
22B−Ins
wen3−8
n = 5
n = 3
n = 1
top_n_rag_examples
numeric
1.00
0.75
0.50
0.25
0.00


ma−3−12
ma−3−27
3.1−8B−
out−17B−
8B−Instr
arge−Inst
22B−Ins
wen3−8
1.00
0.75
0.50
0.25
0.00
n = 5
n = 3
n = 1
top_n_rag_examples


nma−3−12l
nma−3−27l
−3.1−8B−Ir
cout−17B−1
−8B−Instru
.arge−Instru
A22B−Instr
Qwen3−8B
FALSE
TRUE
1.00
0.75
0.50
0.25
0.00
n = 5
n = 3
n = 1
top_n_rag_examples


nma−3−12l
nma−3−27l
−3.1−8B−Ir
cout−17B−1
−8B−Instru
.arge−Instru
A22B−Instr
Qwen3−8B
FALSE
TRUE
1.00
0.75
0.50
0.25
0.00
n = 5
n = 3
n = 1
top_n_rag_examples


nma−3−12l
nma−3−27l
−3.1−8B−Ir
cout−17B−1
−8B−Instru
.arge−Instru
A22B−Instr
Qwen3−8B
FALSI
TRUE
1.00
0.75
0.50
0.25
0.00
n = 5
n = 3
n = 1
top_n_rag_examples

G.2.2.4 Hypotheses

G.2.3 Synthetic tables

G.2.3.1 Confidence

G.2.4 Hybrid approach

real_table_extraction_llm_synth_context_shap_plot

Chapter H

Layout testing

12sdgsdgj

Landscape Landscape

Hello