BHT Berliner Hochschule für Technik

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

Studiere Zukunft

Gutachter

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

Abstract

Content of this thesis is a benchmark on information extraction from PDFs. The focus are annual reports of German companies. Special characteristic of the task is handling hierarchies in tables with financial data to prepare the data for import into a relational database.

The benchmark is composed of two sub tasks and the performance of different open source large language models is tested with different prompting approaches and compared to alternative methods.

This can be seen as a reimplementation study of "Extracting Financial Data from Unstructured Sources: Leveraging Large Language Models" - a paper published by Li et al. (2023). The key differences are the application on German documents using open source large language models.

We show, that also smaller open source LLM (large language model)s can be used to identify the pages that contain the information of interest and to extract it. Based on these findings we sketch a process, how humans can use LLMs to extract information from financial reports.

Zusammenfassung

Gegenstand dieser Arbeit ist ein Benchmark zur Informationsextraktion aus PDF-Dateien. Dabei wird sich auf das Auslesen der Bilanzen und Gewinn- und Verlustrechnungen aus Jahresabschlüssen deutscher Unternehmen beschränkt. Ein besonderer Aspekt der Aufgabe ist die Berücksichtigung der Hierarchie innerhalb der Tabellen, um die Werte einem festen Schema zuzuordnen und so den Import in eine relationale Datenbank vorzubereiten.

Reading advices

The author recommends 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[1]. The more, the merrier.

Goals and Learnings

Achieved:

- thesis with bookdown
- docker image creation
- cluster orchestration
- llm usage
- guided decoding

[1] Most of the time the thesis was inspected at a third of the authors 42" screen with 4k resolution. For inspecting the large overview graphics it is a very handy tool the author recommends every data scientist or software developer.

Missed:

- Administrating a k8s cluster
- Fine tuning a model
- using small language models
- training a lm
- using vllms

Dedication

Micha

Contents

Chapter 1

Introduction

1.1 Motivation

- market: public administration, companies with data of special requirements for treating (secret and personal data (high risk data)) <– DSGVO, AI act
  - next market for hyper scalers might be public administration with local computing clusters
- whom is it helping
- why now: digital sovereignity, AI act; people want NLP AI products, frameworks get easier
- is the problem easier solvable then years ago? why?

missing law to access digital data and no law to choose the format of the data extensible Business Reporting Language as a standard changing from HGB to IFSR

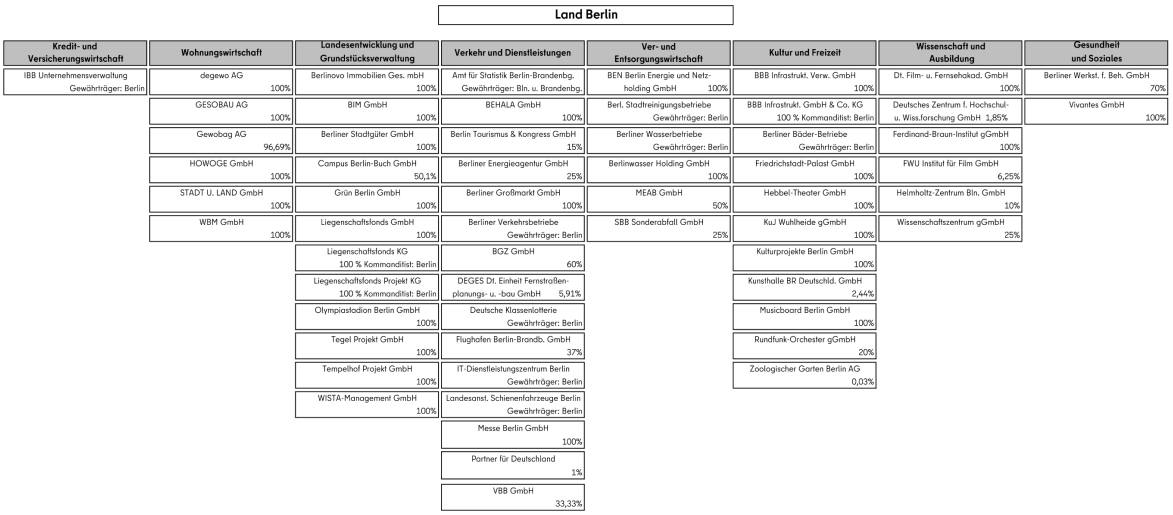


Figure 1.1: Overview of companies Berlin holds share at

1.1.1 Human in the loop

test HITL (human-in-the-loop)

human in the loop (Mosqueira-Rey et al., 2023; Natarajan et al., 2024; Wu et al., 2022)

1.2 Objectives

The sixth division at RHvB is auditing the companies Berlin is a stakeholder of. Basic information they have to process are the balance sheets and profit and loss accounting. Those information is provided via their annual reports in form of PDF files. The provided annual reports often differ from the publicly available ones in matter of information granularity and design and are treated as non public information. Automate the extraction of those information would be a good starting point for AI assisted information retrieval from PDFs for the RHvB overall.

It is important to get numeric values totally accurate; numeric values are difficult to handle for langauge models

- special part of big problem? central question
- two sentences: why this problem? new problem or just a part in the big task? hard to solve of straight forward? research or application? what was not done and why?
- building a system? what task to solve? core functionality? typical use cases?

Research questions and hypothesss

Q1: Can a LLM be used to efficiently extract financial information from German annual reports? Q2. Can LLMs be used to identify the page of interest automatically?

Q3: Can confidence scores be used to head up the human in the loop on which results to double check? (How can sources of the automatic extraction being communicated down stream in order to make double checking easy before making decisions?) Q4: Can contextual information from similar documents reduce errors made during table extraction? Q5: What are characteristics of financial tables that make it hard for LLMs to identify / extract them? (How does the length and complexity of financial documents (e.g., multi-column layouts, nested tables) affect table extraction performance?)

1.3 Methodology (1 p)

- how to solve the problem?
- what foundations to have in mind?
- proceeding?

Experimental / Comparative Research • Reimplementing framework(s) • Comparing / Benchmarking • Frameworks • Models • Methods • Use cases • Ablation test

1.4 Thesis Outline (0.5 p)

1.5 To place in chapters above

This master thesis is motivated by a use case from practical work at the Berlin court of audit (Rechnungshof von Berlin; RHvB). The auditors often are faced with the problem that they need information that is provided as natural language or in tables inside of unstructured documents, i.e. in PDF files. The goal of this thesis is benchmarking methods for automated information extraction from specific tables from PDF files.

Ideally, the data extraction pipeline is able to autonomously * identify the pages with the tables of interest. * identify the tables of interest on these pages. * extract the information as provided into a structured table (e.g. as JSON, a csv file or HTML code). * transform the data into a given schema, stripping all aggregated values.

It should extract the values without errors. It would be nice if the computation time and energy consumption is as low as possible.

A more realistic approach, that is also beneficial to satisfy the AI Act (keine Entscheidung ohne menschliche Beteiligung), is an assistant system, that helps extracting information. Key features to get the human into the loop already at the step of information extraction for such an assistant might be:

- showing the results together with the systems confidence.
- showing the results next to the values of the source.
- allowing in place adjustments to the extracted data.

A sound decision making is only possible if the information the decision is based on is valid.

1.6 RHvB

- what does the RHvB do
- why is this important
- what does it not do yet (because data source is missing)

1.7 Datenverfügbarkeit

- keine Regelung, in welcher Form der Rechungshof die Daten, die er benötigt, bereitgestellt zu bekommen hat

Das Gesetz zur Förderung der elektronischen Verwaltung (EGovG) wurde erlassen, “um die Verwaltung effektiver, bürgerfreundlicher und effizienter zu gestalten.” (BMI, Referat O2, 2013)

§ 12 EGovG

- Vorhaben zur Datenkatalogisierung innerhalb der Verwaltung angestoßen, aber noch nicht richtig gestartet
- Vornehmlich für Bürger*innen Zugang

1.8 Unstrukturierte Daten

- Beispielbilder

1.8.1 Portable Document Format

- print optimized
- Table structure information gets lost
- Bild und Textextract

1

Chapter 2

Literature review (less than 10 p)

(5 to 10 lines)

- overview of subchapters
- relevance for reader (Gutachter)
- link to previous chapter
- relevant basic tasks
- parameter vs active parameter

2.1 NLP history

2.2 Basic terms

2.3 Supervised Learning Approaches

2.3.1 Generalized Linear Models

2.3.2 Random Forest

XGBoost not used finally, because calculation SHAP (SHapley Additive exPlanations) values for XGBoost model took to long for just a first glimpse on what might influence the extraction.

2

2.3.3 Large Language Models

2.3.3.1 Embeddings

2.3.3.2 Neural networks in NLP

2.3.3.3 Attention / Multi-Head

2.3.3.4 Transformers

2.3.3.5 Encoder

2.3.3.6 Decoder

2.3.3.7 BERT

2.3.3.8 Bi-Encoder

2.3.3.9 Mixture of Experts

2.3.3.10 Guided decoding

generation template strict (closed) vs open

2.3.3.11 Classification trained models (not used)

Soft max

2.3.3.12 Few-shot Learning

2.3.3.13 RAG

2.3.3.14 GPT (Generative Pretrained Transformers)

2.3.4 Information extraction

closed-domain vs open-domain

2.4 Data balancing

2.4.1 Under sampling, oversampling

2.5 Evaluation Metrics

2.5.1 For classification

2.5.2 For regression

2.6 Technological topic (related work)

- LLM generation
- structured output

- Fewshot
- context length can be harmful
- most important papers
- connection of papers (timeline)
- what used, what not?
- extending existing paper?

2.7 Term frequency

2.7.1 Extraction of numeric values

99.5 % or 96 % accuracy for extracting financial data from Annual Comprehensive Financial Reports (Li 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.)

Too many hallucinated values when it was NA instead (Grandini et al., 2020)

2.8 optinal more topics like previous

2.9 Summary (0.5 p)

- lessons learned
- link to goal thesis
- link to next chapter

2.10 To place in chapters above

2.11 Table extraction tasks

2.11.1 Difficulties

- Beispielbilder

2.12 Document Extrtaction Process

2.12.1 Document Layout Analysis

An important step in the process of extracting information from documents is to recognize the layout of a document (Zhong et al., 2019).

Getting the order of texts correct align captions to tables and figure identify headings, tables and figures

One of the most popular datasets used for training and benchmarking is PubLayNet (see PubLayNet on paperswithcode.com). It contains over 360_000 document automatically annotated images from scientific articles publicly available on PubMed Central (Zhong et al., 2019, p. 1). This was possible, because the articles have been provided in PDF and XML format. For the annotations most text categories (e.g. text, caption, footnote) have been aggregated into one category. <– is this a problem for later approaches where a visual and textual model work hand in hand to identify e.g. table captions?

Manual annotated datasets often were limited to several hundred pages. Deep learning methods need a much larger training dataset. Previously optical character recognition (OCR) methods were used.

Identify potentially interesting pages with text / regex search. Check if there is a table present on this page.

Object detection

2.12.1.1 Vision Grid Transformer

2.12.2

2.13 Tools

2.13.1 TableFormer

SynthTabNet <– has it: - nested / hierarchical tables, where rows add up to another row? - identifying units and unit cols/rows

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 as LLMs, presented by Li et al. (2023). We apply this framework on German annual reports of multiple companies and focus on using open source 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 in form 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 of interest is ordered by a hierarchy defined in HGB (2025). The information to extract are 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 recognized and finally its numeric value has to be extracted. In this thesis no special techniques for the table extraction sub field are used.

3.2 Research Design & Philosophy

The research design for this thesis is set up, following the guideline found in Wohlin et al. (2024) and Figure 3.1 shows the decisions made. 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. The data collected in our experiments is of quantitative nature and its evaluation uses (semi-)quantitative methods.

Research logic? Methods fit deductive. But I start with a very specific problem. Literature describing basic approaches?

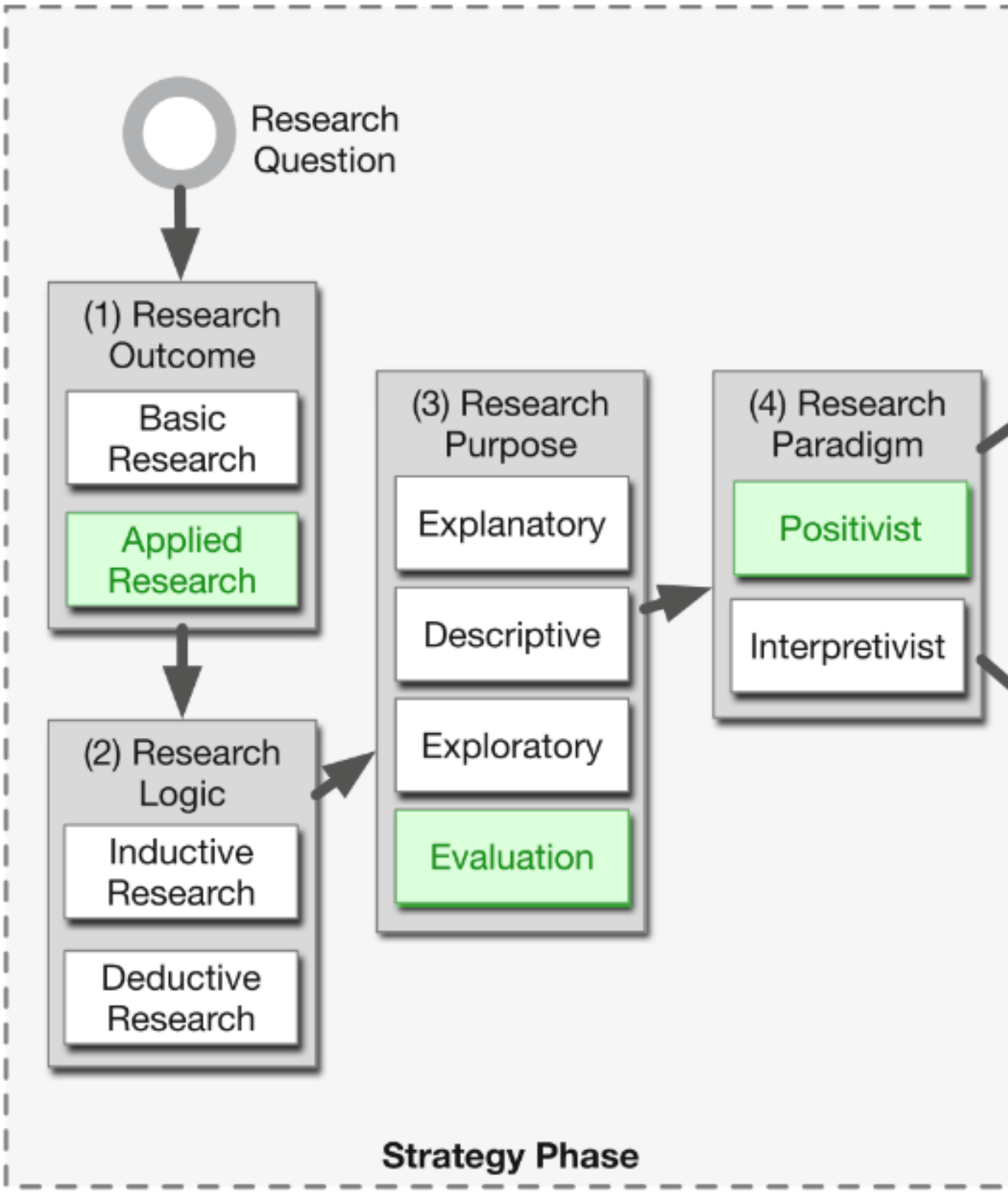


Figure 3.1: Showing the decisions made regarding the research design. (The figure is adapted from Wohlin & Aurum (2015). The copyright for the original figure is held by Springer Science+Business Media New York 2014.)

3.2.1 Research questions

For this thesis we formulate two main research questions:

1. How can we use LLMs effectively to locate specific information in a financial report?
2. How can we use LLMs effectively to extract these 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.

3.2.2 Evaluation framework

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**.

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 values 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.

3.2.3 Evaluation research

We compare different approaches to solve the two tasks, searching for the most effective setup, to solve the problems. A task 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 performance as well.

The results should be used to implement an application that is used by the employees of RHvB (Rechnungshof von Berlin) in future.

3.3 Evaluation Strategy

3.3.1 Metrics

Page identification 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.

Precision-recall curve

Information extraction We use two measures to describe the approaches performances for the information extraction task. First, we check how many of the predicted numeric value 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 very imbalanced. Nevertheless, we report the F1 score to establish a comparability with the results od the page identification task.

3.3.2 Benchmarking

Comparing the 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.

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 there 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 shows 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

Page identification For the page identification task companies (mainly) from the first row of Figure 1.1 have been selected, to build the ground truth from. The idea 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 (optical character recognition) preprocessing, we include the documents of a second company for this category.

Information extraction For the information extraction task two sampling iterations are gone through. 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 allow to test the in-company approach for all companies.

3.4.2 Ground truth creation process

The ground truth for both tasks is created by by manual labor of the authors. The results of early experiments are used to check the ground truth for mistakes or missed items.

Page identification For the page identification task the chosen documents are searched for the target pages either by using the search functionality, TOC (table of contents) 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.

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.

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 A.2.1 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 Hardware normalization

To make the runtime of different LLMs running on different amounts and types of GPU (graphics processing unit)s comparable, we conducted a benchmark running the models Qwen2.5-7B and Qwen2.5-32B with different hardware compositions on the Datexis cluster. Figure 3.2 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.

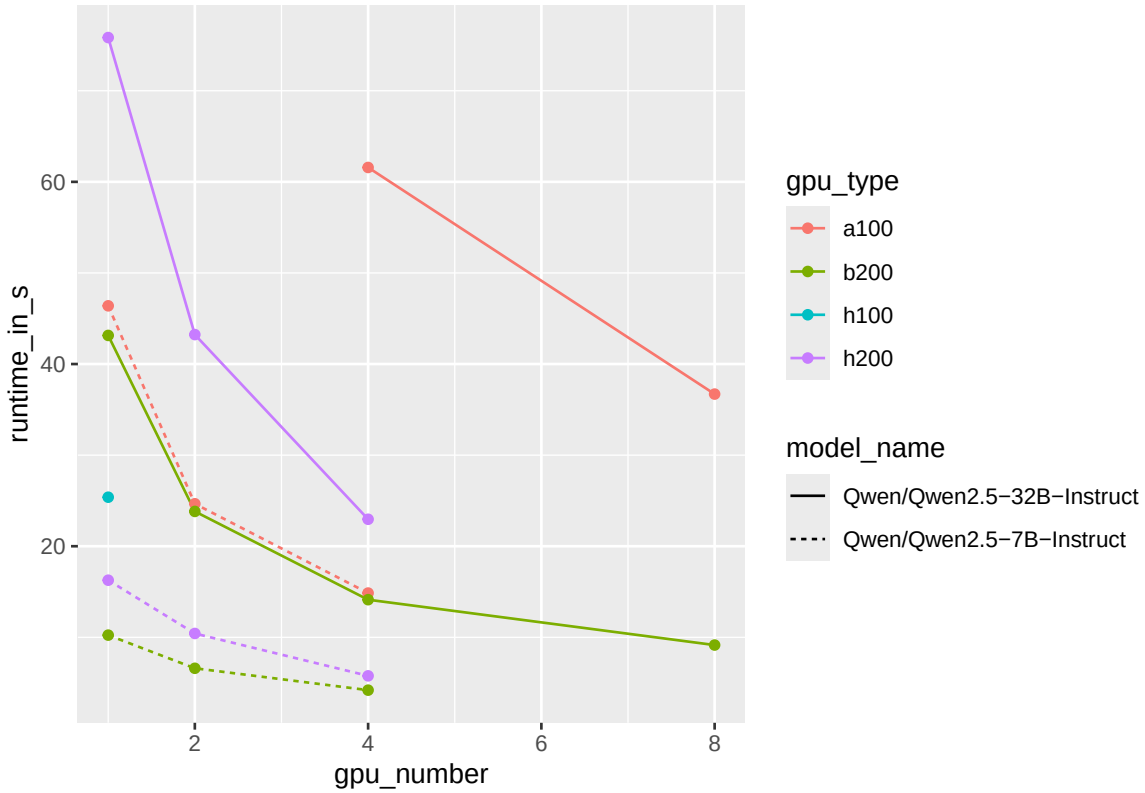


Figure 3.2: Showing the runtime to classify 100 pages with the multi-class approach, providing three random examples for the in-context learning.

We run our final experiments with the vLLM (Virtual Large Language Model) library on Python, using its batch processing capabilities. Our first test used the *transformers* library directly and did not use batch processing. Section A.2.3 shows the runtime reduction that is achieved with the final setup.

3.5.2 Error analysis

The ultimate goal is to fully automate the information extraction task at hand. Thus, it is important to analyse potential errors, to identify obstacles that hinder performance and find ways to further improve the system.

We expect to find issues with wrong extracted or halluzinated numbers, wrong entity recognition and false positive *null* values.

Quantitative / stratified: We will compare the error rates based on the different variables of the experiments. For the approaches using LLMs to solve the problem there are model specific, prompting strategy specific and example specific variables.

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

Tools and criteria

Reporting

Example:

To better understand the limitations of the evaluated models, we will conduct a detailed error analysis. We will first quantify the types of errors (e.g., false positives, false negatives, misclassifications) using confusion matrices and error rate statistics. Additionally, we will manually inspect a sample of erroneous predictions to identify common causes, such as ambiguous table layouts, OCR errors, or model misinterpretations. Errors

will be categorized by document type and extraction task to reveal systematic weaknesses. Representative error cases will be documented to illustrate typical failure modes and to inform potential improvements for future work.

3.5.3 Baseline selection rationale

see section Evaluation research

3.5.4 Evaluation methods

- box plots
- PR-curve
- random forest + SHAP

3.6 Ethical & Practical Considerations

3.6.1 PDF extraction limitations

One library warned that some PDFs have a meta tag that says they should not be extracted

pdfminer informs that some pdfs should not be extracted based on their authors will (meta data field)

3.6.2 Computational constraints

3.6.3 Generalizability scope

Chapter 4

Implementation (max 5p)

4.1 Speedup with vLLM and batching

4.2 Setup (Dockerfile and PV)

4.3 Page identification

Due to the imbalanced distribution of the classes the accuracy is not a good metric to compare the performance of the different methods. The number of pages of interest is much smaller than the number of irrelevant pages. Therefore, precision, recall and F1 score are presented as well.

4.3.1 Baselines

4.3.1.1 Regex based

results dependend on regex pattern

start with pypdf backend and simple regex developed more sophisticated regex based on missed pages

took wrong identified pages as base for a table detection benchmark and n-shot base for llm classification (contrasts)

some tables can't be found without previous ocr; some pages hold image of table and machine readable text

LLM based

4.3.1.2 Term frequency based

VLLM based was not implemented

4.4 Table detection

Can be used to narrow down set of possible pages

Can be used to focus only on the table content (measure if correct area was identified would be necessary)

Vision model as baseline

4.4.1 LLM

- table: yes/no
- akiva: yes/no
- multiclass

4.4.2 Vision Model

Yolo

4.4.3 Docling and Co

VLLM based was not implemented

4.5 Information extraction

4.5.1 Baselines

simple regex?

4.5.2 Simple pipeline

- extract text (if document can't be passed directly)
- query LLM directly

4.5.3 Sophisticated approaches

not implemented

- with pipelines
- Nougat
- maker
- Azure
- docling

Chapter 5

Results

This chapter presents the results for the two research questions of this thesis:

1. How can we use LLMs effectively to locate specific information in a financial report?
2. How can we use LLMs effectively to extract these information from the document?

Section 5.1 presents the results for the first research question. Section 5.2 presents the results for the second question.

Each section will start with an overview about the specific sub tasks as well about the models, methods and data used to investigate the research question. The subsections present the results of the sub tasks. At the end of each section all results get compared and summarized.

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 identifying 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 (Gewinn- und Verlustrechnung) will be used.

Li et al. (2023) describes two ways to identify the relevant pages (see Figure A.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 is presenting four approaches to identify the page[2] of interest.

- Subsection 5.1.1 presents the performance of a page range refinement using a list of key words with a regular expression.
- Subsection 5.1.2 presents the performance of a TOC understanding approach
- Subsection 5.1.3 presents the performance of a text classification using LLMs.
- Subsection 5.1.4 presents the performance of a term-frequency approach.

In subsection 5.1.5 the results get compared and summarized. Subsection @ref() proposes an efficient combination of approaches to solve the task of this thesis and discusses its limitations.

[1] 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.

Table 5.1: 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

Dataset describtion Figure 5.1 shows how the document base for the tasks in this section is composed[3]. 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 5.1 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.

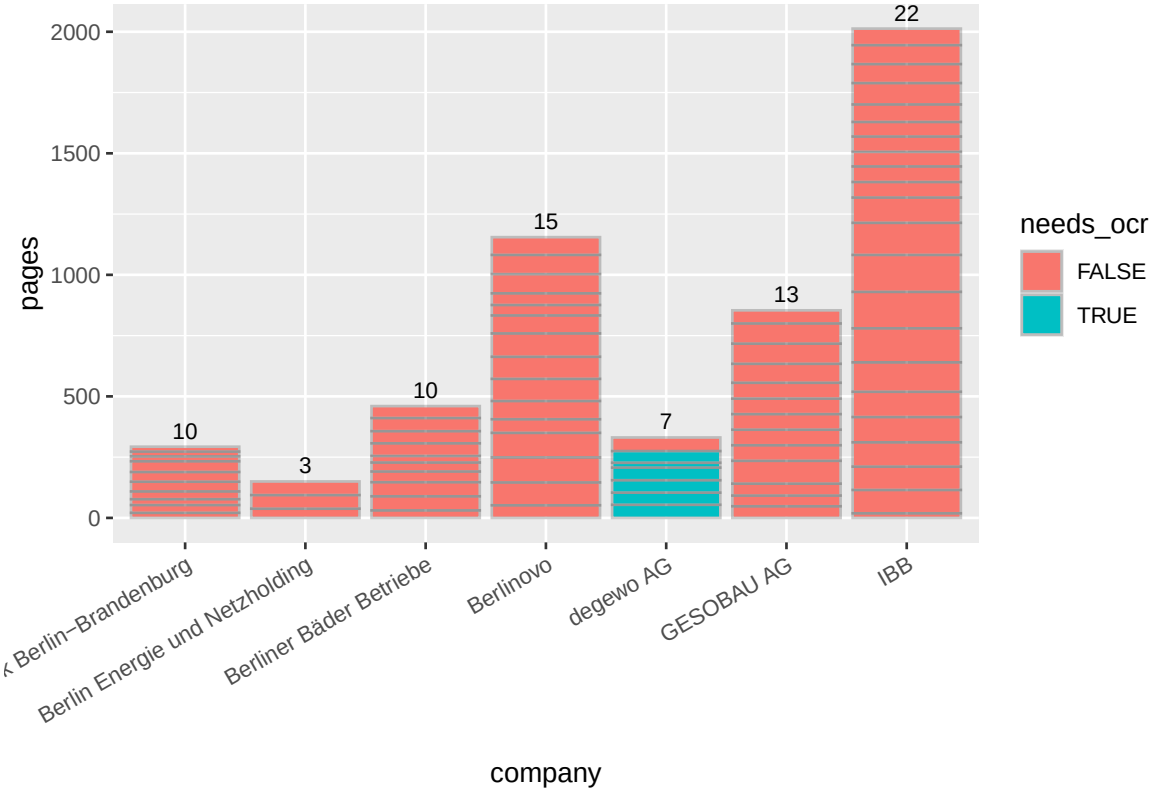


Figure 5.1: Showing the number of pages (bar height) and number of documents (number above the bar) per company for the data used for the page identification task. Some documents would require ocr before being processed and were not used.

Table @ref(tab:display_multiple_tables_per_type_and_document) 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.

This task is broken down to a classification task all of the approaches presented in this section but the TOC understanding approach.

[3]I downloaded all publicly available annual reports for some of the companies shown in the first row of Figure 1.1. I assumed that this will give a representative sample of document structures for the other companies of the same type. Realizing that the degewo AG reports would require ocr preprocessing I additionally downloaded reports for GESOBAU AG. This approach could have been more systematic. For the second task I downloaded reports for all companies available and tried to use a balanced amount of reports per company.

Table 5.2: 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

Thus, we prompt the LLM to classify if the text extract of a given page

for implementation: As described in A.2.1 open source libraries have been used to extract the text from the annual reports.

5.1.1 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[4]. 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 A.4).

Table 5.2 shows the mean performance for precision, recall and F1 for the three regular expressions for the three types of pages to identify[5]. 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 5.2 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*[6].

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[7]. But table A.1 in section A.2.1 shows that there are differences in the extraction speed.

Code can be found at “benchmark_jobs/page_identification/page_identification_benchmark_regex.ipynb”

Todo: * look into details where they differ and if it is because of a line break or whitespace?

5.1.2 Table of Contents understanding

The second approach presented in this section leverages the TOC understanding capabilities of LLMs. Li et al. (2023) use this approach with long documents as a first step to determine a page range of 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 (Portable Document Format) document can be embedded

[4]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.

[5]See Figure C.2 for a graphical representation.

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

[7]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().

5.1

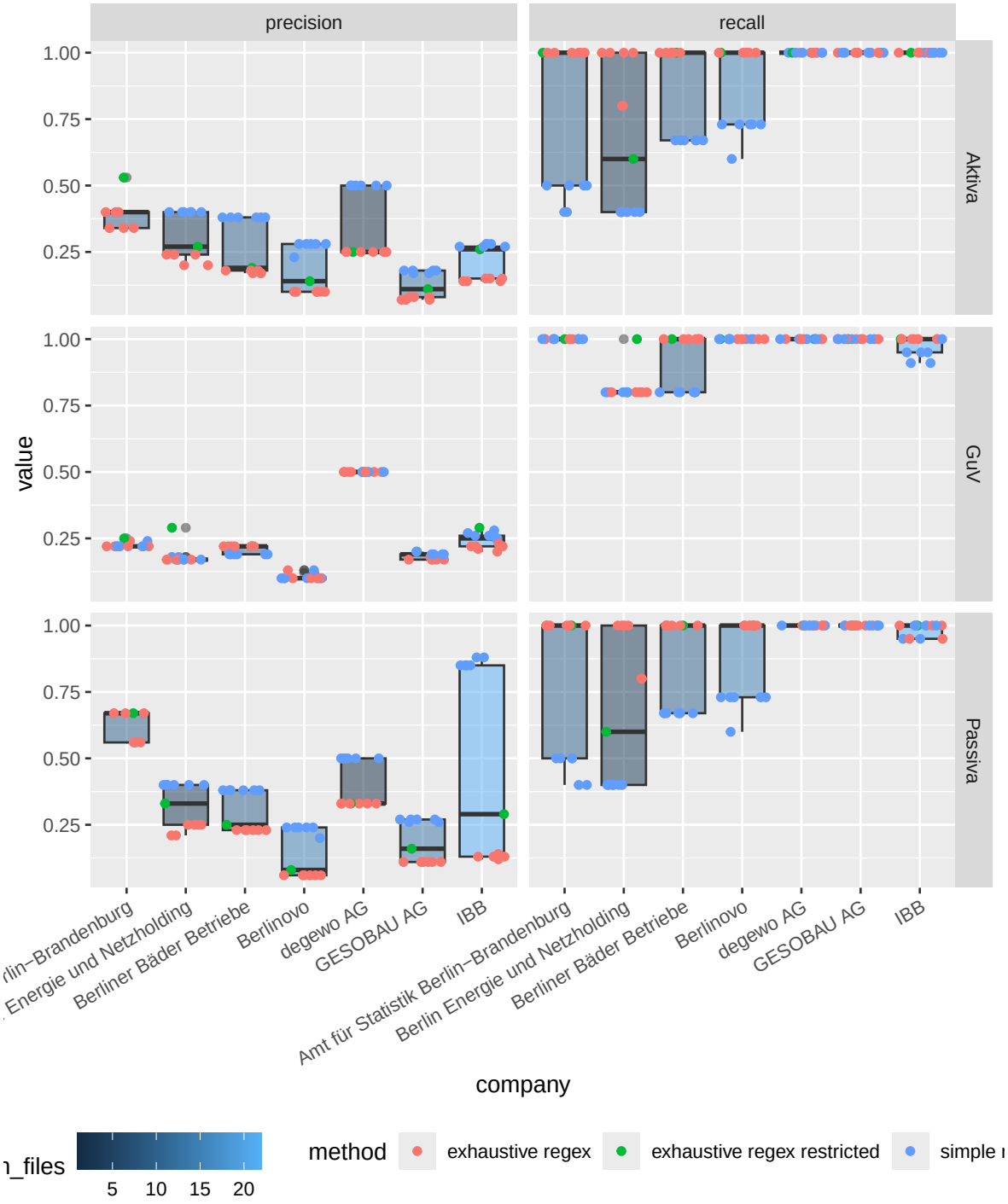


Figure 5.2: Comparing the performance among different companies.

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 5.1.2.1 shows the results for the text based approach. Subsection 5.1.2.1 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 A.3.1. 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"

Discussion:

- Li et al. (2023) did not report any issues with this approach. They use few-shot learning and Chain-of-Thought techniques to help the LLM to understand the task. They ask just for one information at a time.
- ChatGPT 4 vs Mistral 2410 8B (huge parameter difference)
- For a lot of short annual reports one can find the tables of interest within the first eight pages as well.

5.1.2.1 Details for the approaches

Text based Li 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 ACFR (Annual Comprehensive Financial Report)s is found within the initial 165 lines of the converted document (Li 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 5.3). 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 A.3.1]. For the same documents Mistral 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.

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

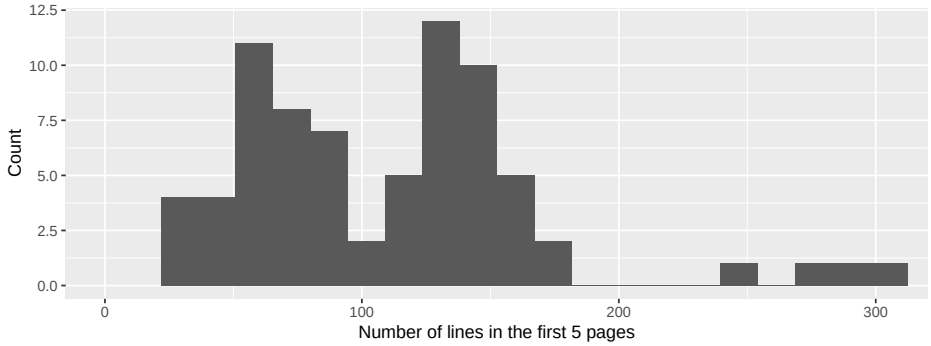


Figure 5.3: Histogram of the number of lines in the first 5 pages of the annual reports

5.1

Table 5.3: 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

```
## |              1 | Bilanz                 |          25 |           6 |
## |              1 | Anhang                 |          31 |           7 |
## |              1 | Anlagenspiegel         |          39 |           8 |
## |              1 | Bestätigungsvermerk    |          42 |           9 |
```

5.1.2.2 Results

Comparison of the different approaches: base prompt Table 5.3 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 %.

Figure 5.4 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 5.4 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.

Table 5.4: 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

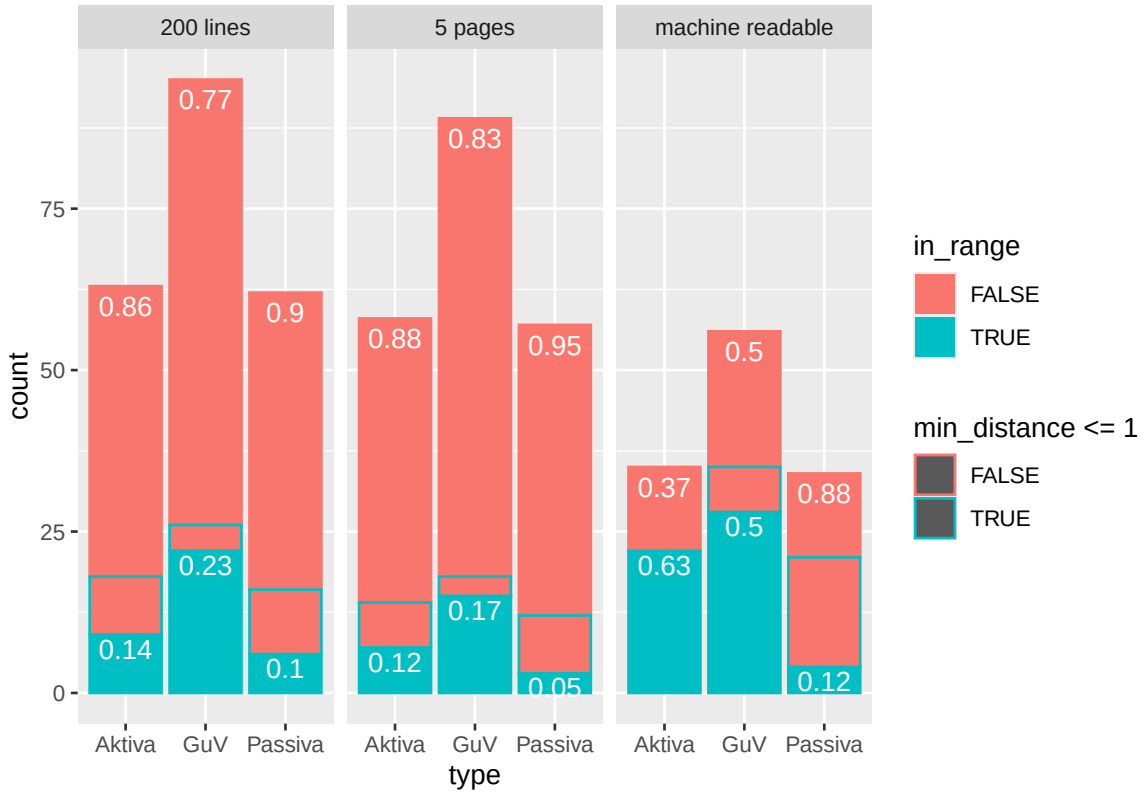


Figure 5.4: Comparing number of found TOC and amount of correct and incorrect predicted page ranges

Comparison of the different approaches: 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 C.3). Figure 5.5 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 5.5 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 5.6 shows, that the machine readable TOC approach is the fastest as well.

Table 5.7 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 5.6 shows, that especially the ranges for **GuV** are not normally distributed. Some far off lying range sizes are shifting the mean off from the median.

Table 5.5: 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}

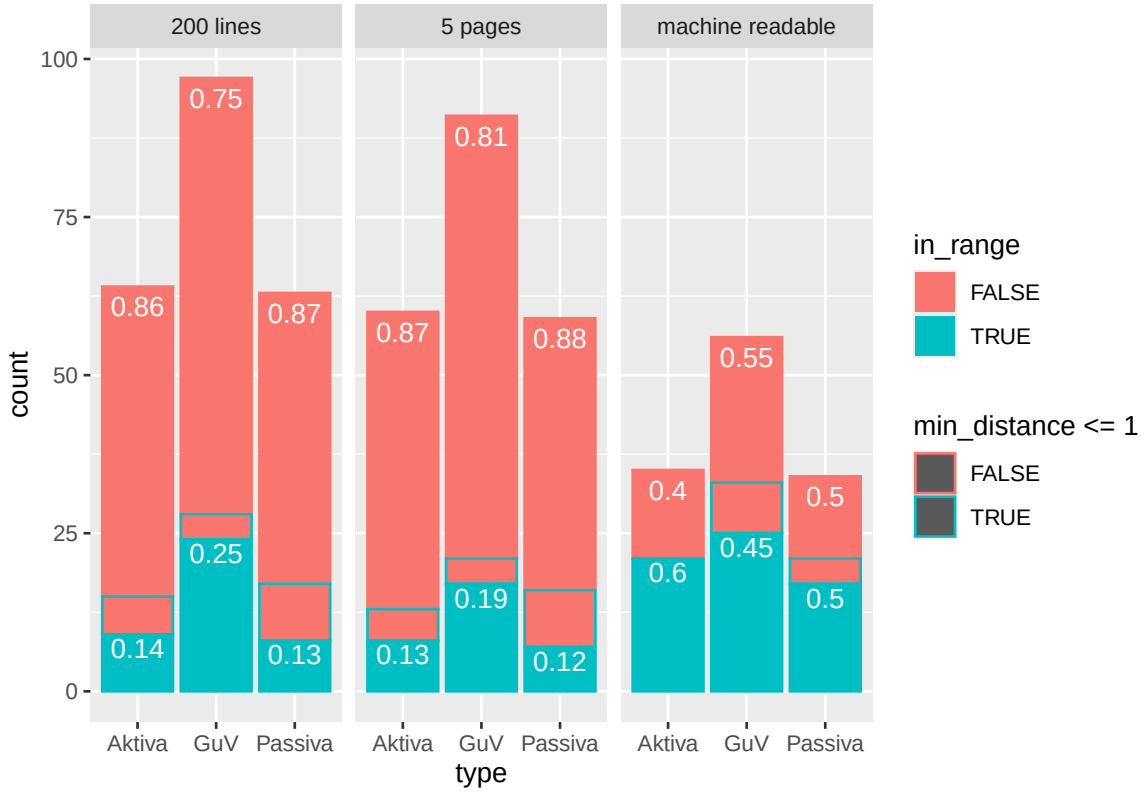


Figure 5.5: Comparing number of fount TOC and amount of correct and incorrect predicted page ranges

Table 5.6: 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

Table 5.7: 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

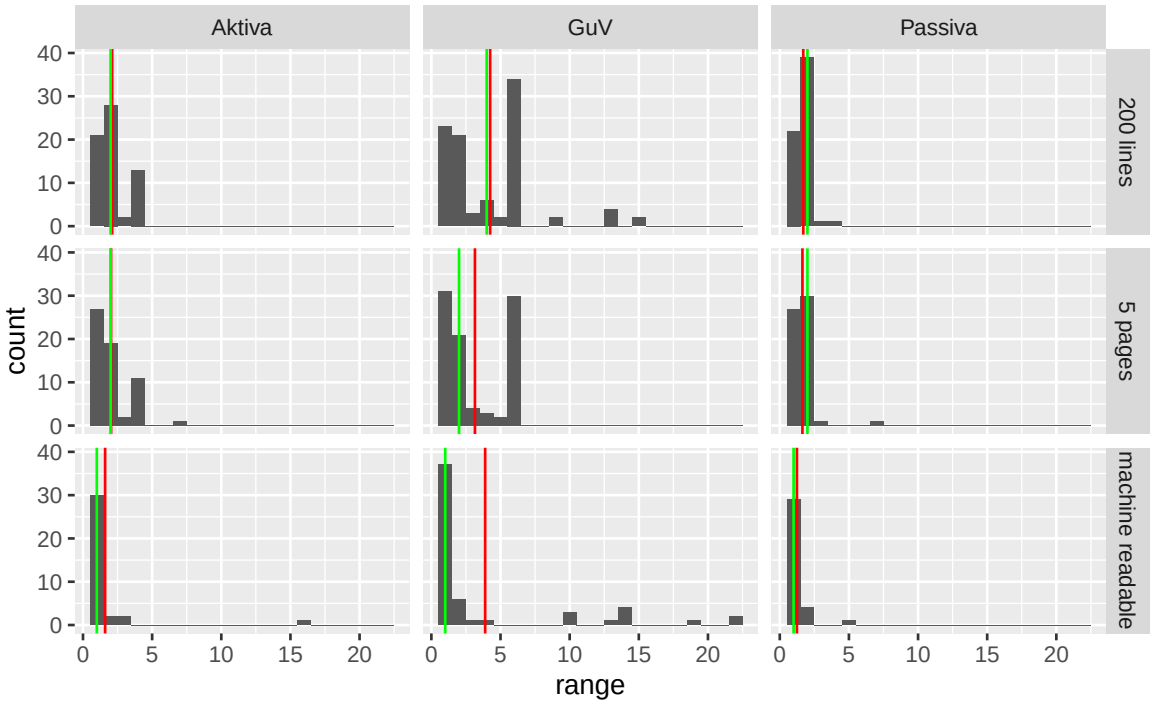


Figure 5.6: Comparint the predicted page range sizes. The red vertical line shows the mean and the green one shows the median of these sizes.

Figure 5.7 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.

Machine readable TOC approach specific results Figure 5.8 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.

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[8]. This was tested because the relation between heading and value for the column *page_number* might have been clearer[9] in json for a one-dimensional working LLM.

Delete or place somewhere else?:

- Thus it is safer to go with the 200 lines approach. But it also takes longer. 5.6
- Values can be higher than 80, the total number of PDF files, since there can be multiple tables of interested for the same type in a single document or a table of interest can span two pages.

5.1.3 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*.

[8]This result is based on a single test run.

[9]With json the key *page_number* gets repeated every line, while it is just mentiones once in the beginning of the markdown formatted tables.

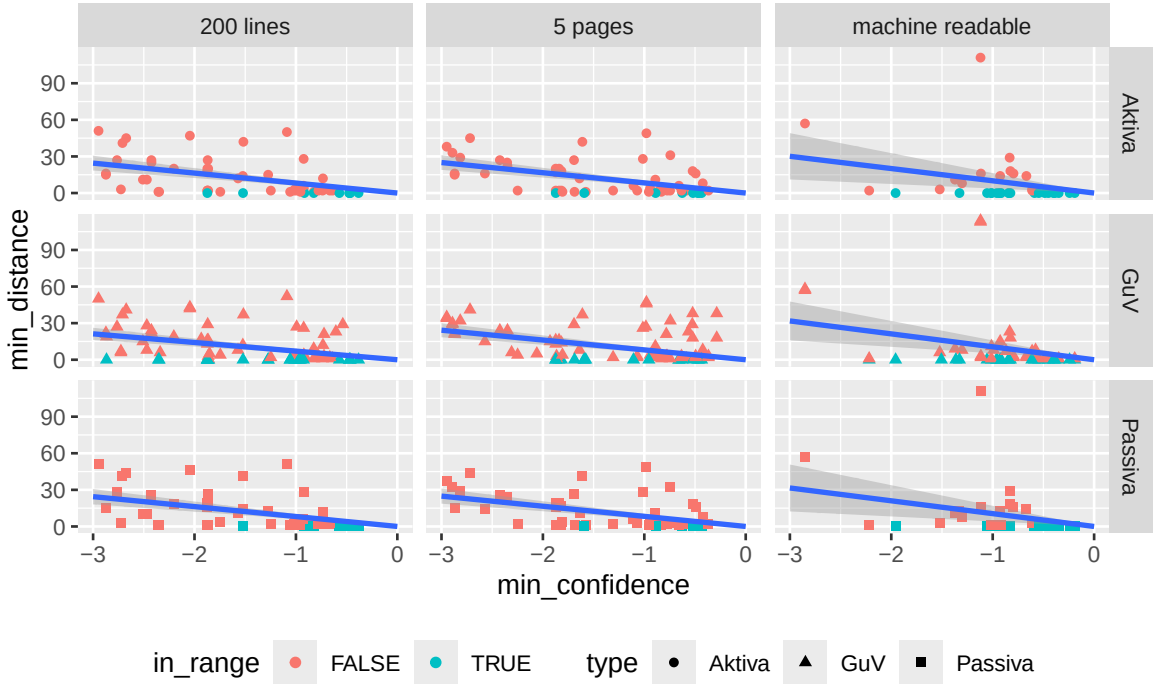


Figure 5.7: Showing the minimal distance of the predicted page range to the actual page number overthe logprobs of the models response confidence.

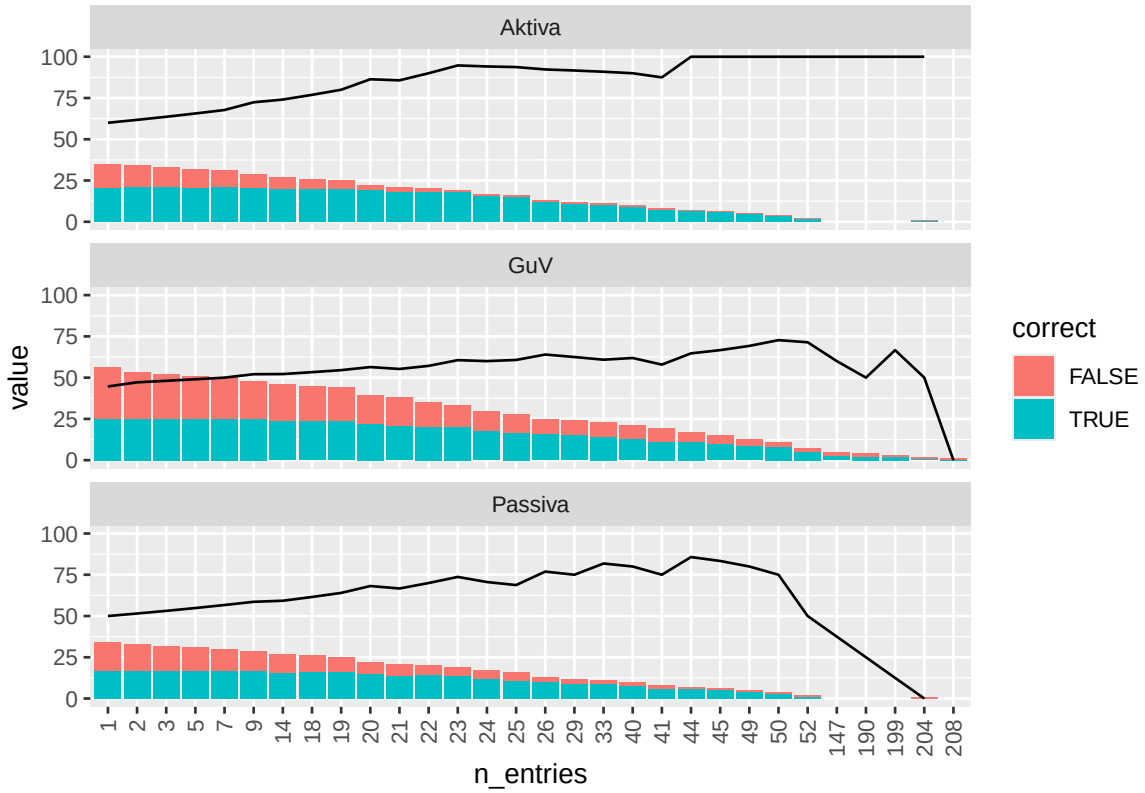


Figure 5.8: Showing the amount of correct and incorrect predicted page ranges (bars) and the percentage of correct predictions (black line).

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 A.3.2.

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 5.9 shows how many examples the LLM gets provided, depending on the classification type and chosen parameter *n_example*[10]. 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 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.

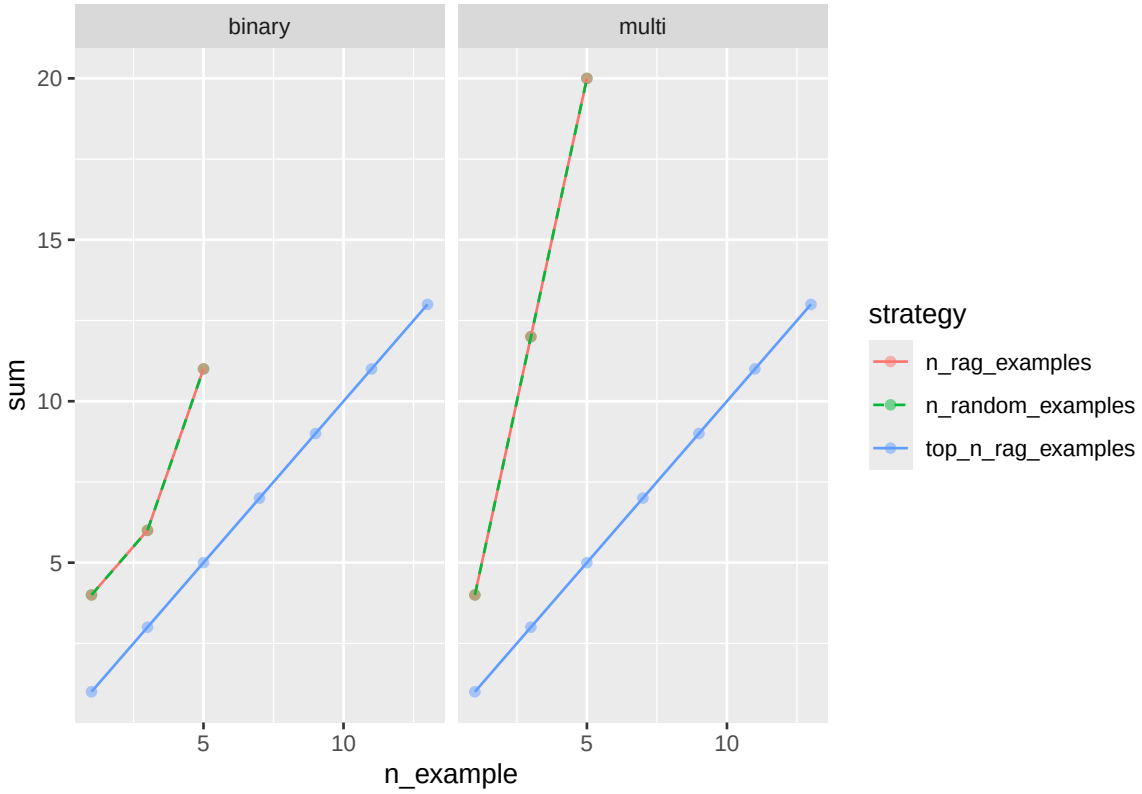


Figure 5.9: Comparing the actual number of provided examples depending on the classification type, example selection strategy and chosen parameter n-examples. The slope for the top-n-rag-examples strategy is the same for both approaches. The line for the strategies n-random-examples and n-rag-examples is equal within each approach.

Table 5.8 shows, which models have been used in the classification benchmarks. Overall 25 models from 6 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

[10] See also Table B.1.

Table 5.8: Overview of benchmarked LLMs for the classification tasks.

model_family	model	parameter_count
Falcon310BInstruct	Falcon310BInstruct	10
gemma312bit091	gemma312bit091	12
gemma327bit091	gemma327bit091	27
gemma34bit091	gemma34bit091	4
gemma3nE4Bit091	gemma3nE4Bit091	4
Llama3	Llama3.18BInstruct	8
Llama3	Llama3.170BInstruct	70
Llama3	Llama3.370BInstruct	70
Llama4	Llama4Maverick17B128EInstructFP8	17
Llama4	Llama4Scout17B16EInstruct	17
Ministral8BInstruct2410	Ministral8BInstruct2410	8
MistralLargeInstruct2411	MistralLargeInstruct2411	124
MistralSmall3.124BInstruct2503	MistralSmall3.124BInstruct2503	24
phi4	phi4	15
Qwen 2.5	Qwen2.50.5BInstruct	0.5
Qwen 2.5	Qwen2.51.5BInstruct	1.5
Qwen 2.5	Qwen2.53BInstruct	3
Qwen 2.5	Qwen2.57BInstruct	7
Qwen 2.5	Qwen2.514BInstruct	14
Qwen 2.5	Qwen2.532BInstruct	32
Qwen 2.5	Qwen2.572BInstruct	72
Qwen 3	Qwen38B	8
Qwen 3	Qwen330BA3BInstruct2507	30
Qwen 3	Qwen332B	32
Qwen 3	Qwen3235BA22BInstruct2507	235

5.1

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

5.1.3.1 Binary classification

Table 5.9 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 0.93 0.1 higher than the median F1 score for **Aktiva** (0.84) and 0.2 higher than the median F1 score for **Passiva** (0.74).

Mistral 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 Mistral 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[11], because it has a total of 109B parameters. It was surprising that Googles gemma models perform so bad[12].

Figure 5.10 shows, the classification performance for Mistral 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

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

[12] 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.

Table 5.9: 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	run
mistralai	Ministral8BInstruct2410	GuV	n_rag_examples	3	0.99	
meta-llama	Llama4Scout17B16EInstruct	GuV	n_rag_examples	3	0.98	
Qwen	Qwen2.532BInstruct	GuV	n_rag_examples	1	0.93	
mistralai	Ministral8BInstruct2410	Passiva	n_rag_examples	3	0.92	
mistralai	Ministral8BInstruct2410	Aktiva	n_rag_examples	3	0.92	
meta-llama	Llama4Scout17B16EInstruct	Passiva	n_rag_examples	3	0.86	
Qwen	Qwen2.532BInstruct	Aktiva	n_rag_examples	1	0.85	
Qwen	Qwen3235BA22BInstruct2507	Aktiva	n_rag_examples	3	0.85	
meta-llama	Llama4Scout17B16EInstruct	Aktiva	n_rag_examples	1	0.84	
meta-llama	Llama4Scout17B16EInstruct	Aktiva	n_rag_examples	3	0.84	
Qwen	Qwen2.532BInstruct	Passiva	n_rag_examples	1	0.81	
microsoft	phi4	Aktiva	law_context	1	0.7	
microsoft	phi4	Passiva	law_context	1	0.66	
google	gemma327bit091	Passiva	n_rag_examples	1	0.58	
google	gemma327bit091	Aktiva	n_rag_examples	1	0.54	
google	gemma327bit091	GuV	n_rag_examples	1	0.52	
tiiuae	Falcon310BInstruct	Passiva	n_random_examples	1	0.5	
tiiuae	Falcon310BInstruct	Aktiva	n_rag_examples	1	0.45	
tiiuae	Falcon310BInstruct	GuV	top_n_rag_examples	1	0.34	

(*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[13], 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 B.1.

The best performing model, Mistral 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 5.10 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[14] 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 5.10 also shows, that the results are stable[15]. 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 5.11 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 com-

[13] 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.

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

[15] 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.

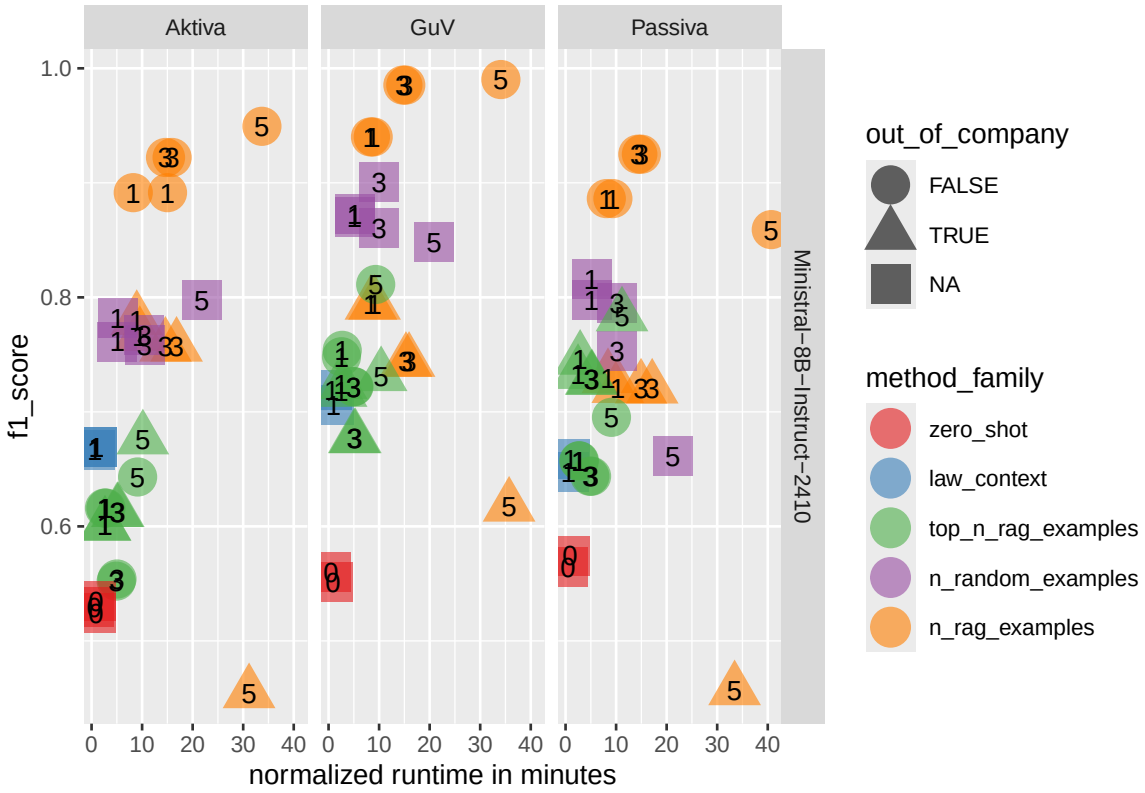


Figure 5.10: Showing F1 score performance over normalized runtime for binary classification for Mistral-8B-Instruct-2410.

parability 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 C.4 and Figure C.5 shows the F1 performance over normalized rundtime for all benchmarked models. Comparing Mistral-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.

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.

Confidence I 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. Figure 5.12 shows the distribution of reported confidence score for the binary classification with target type **Aktiva** for all tables types grouped by their correctness for Mistral-8B-Instruct-2410. The LLM just returns one prediction and its confidence[16]. The abscissa shows $confidence = exp(logprob)$, if the answer is *yes* and $confidence = 1 - exp(logprob)$ if the answer is *no*.

[16]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.

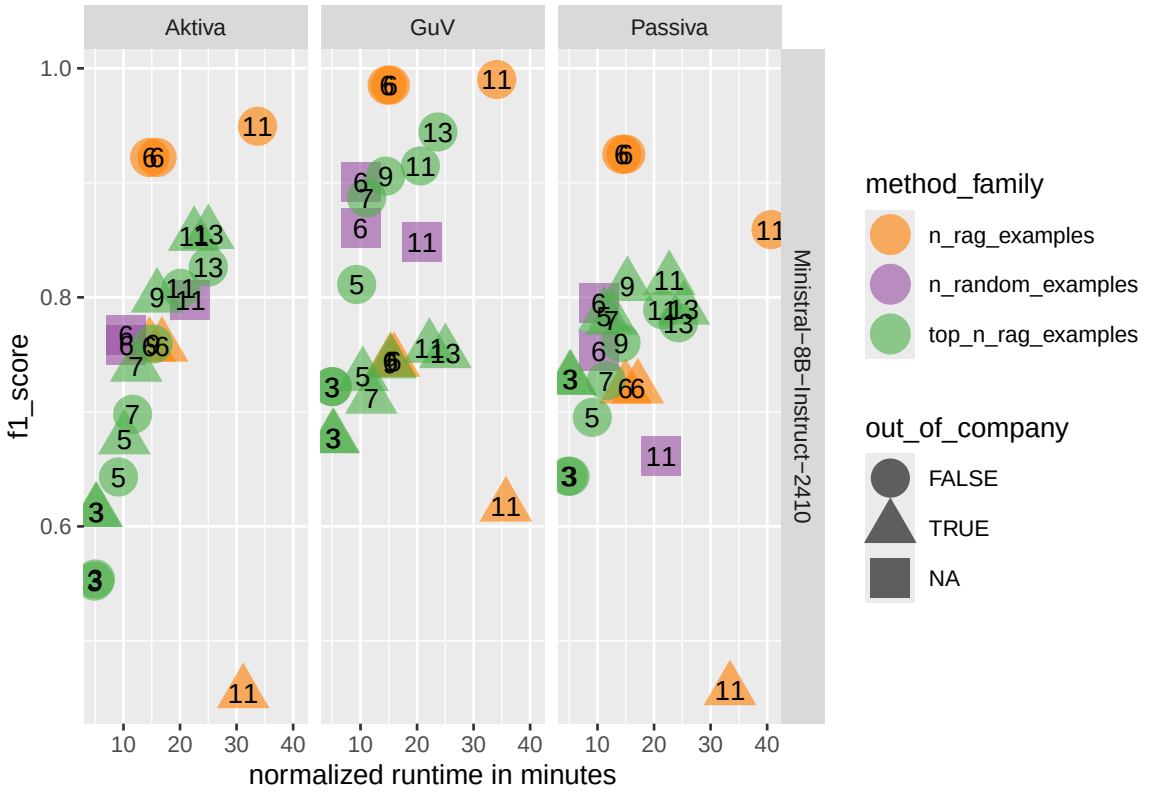


Figure 5.11: Showing F1 score performance over normalized runtime for binary classification for Mistral-8B-Instruct-2410. Comparing the performance based on the real number of provided examples.

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**[17] 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 5.13 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 5.14 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 (area under the curve) 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.

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[18].

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 5.15 summarizes the relation between reported confidence and correctness of the classification for

[17] 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.

[18] Thus, a human has in average to check four pages and select the correct **Passiva** page among them.

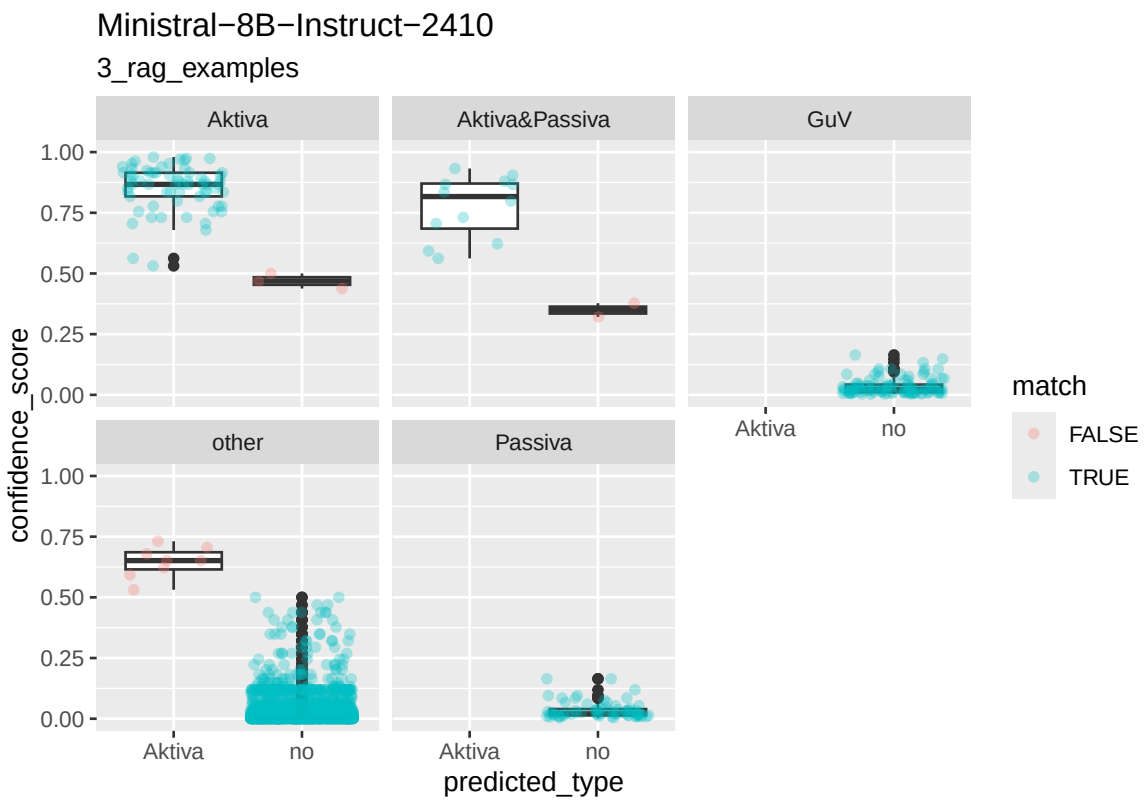


Figure 5.12: Showing the confidence score for the Aktiva classification task grouped by table type and correctness for Mistral-8B-Instruct-2410.

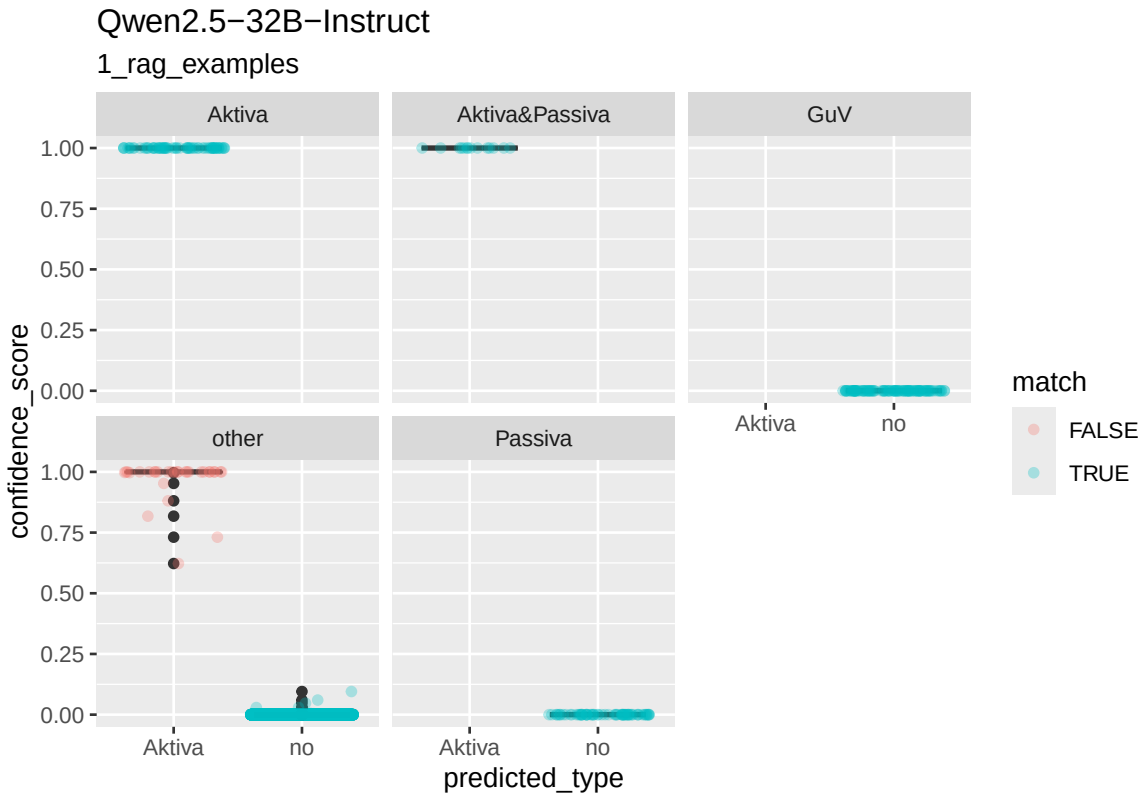


Figure 5.13: Showing the confidence score for the Aktiva classification task grouped by table type and correctness for Qwen-2.5-32B-Instruct.

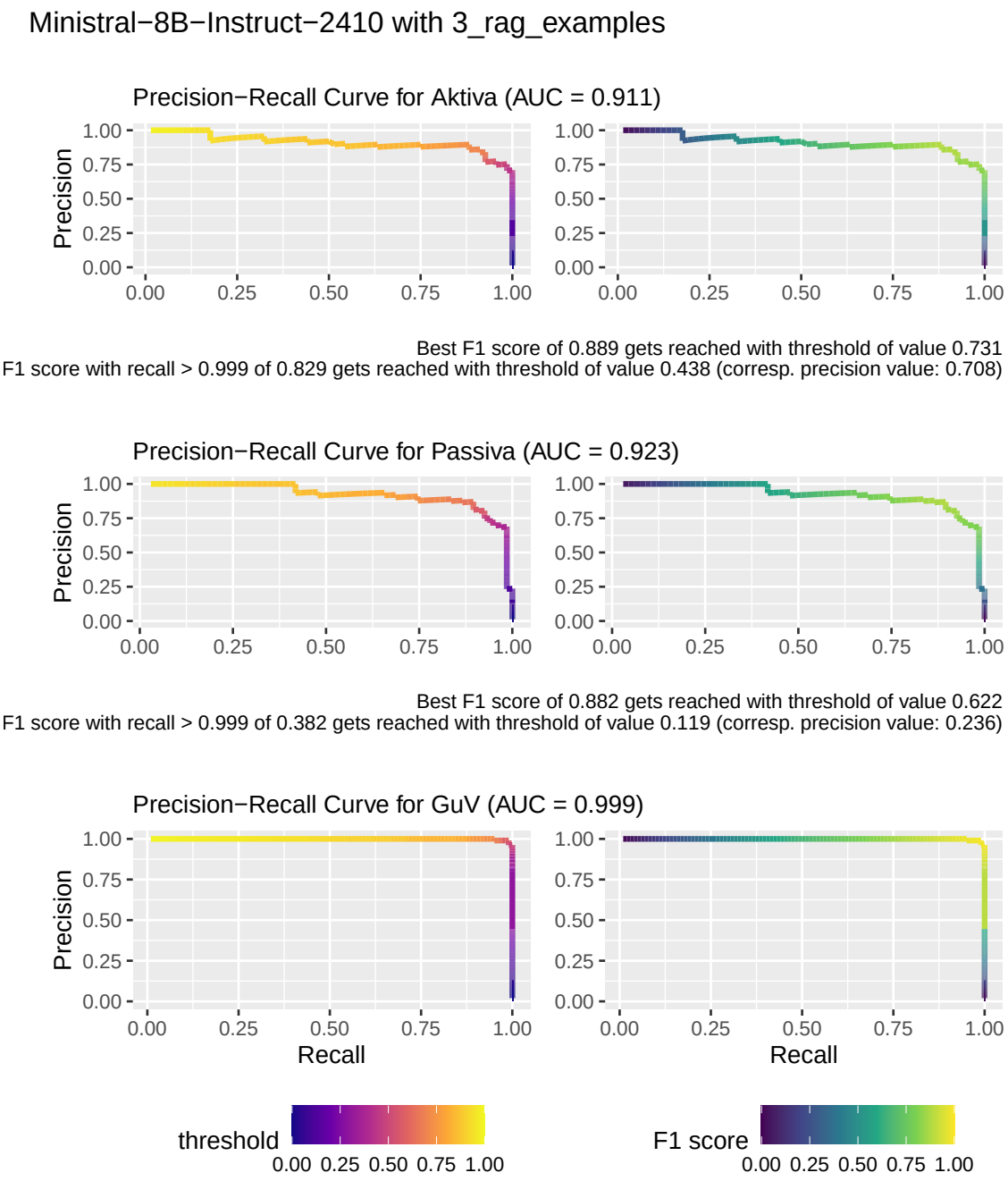


Figure 5.14: Showing the precsion-recall-curve for the best performing model.

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 5.16 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.

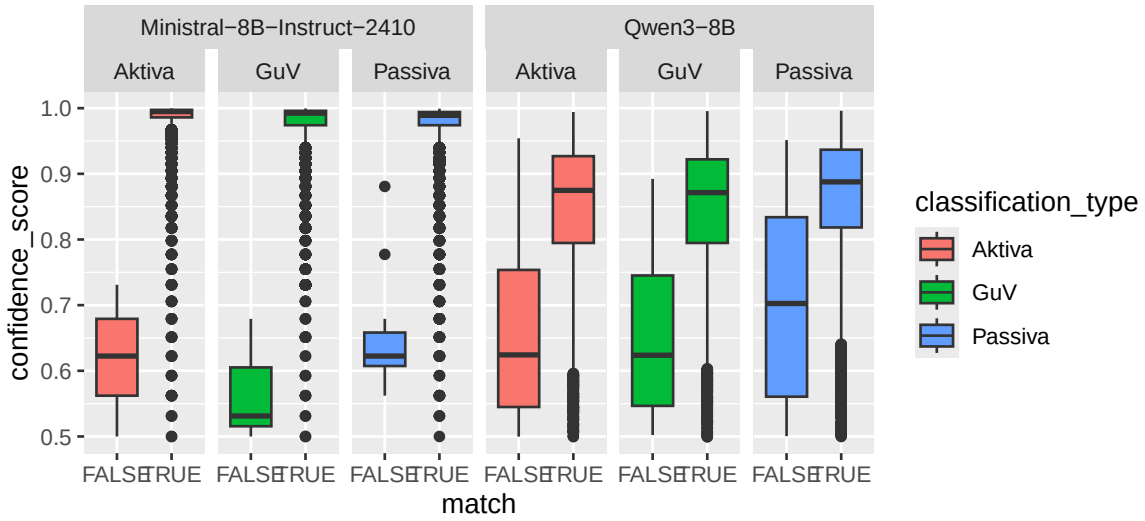


Figure 5.15: Comparing the reported confidence scores for the page identification task for the Mistral and Qwen 3 with 8B parameters.

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

5.1.3.2 Multi-class classification

Table 5.10 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 5.11 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.

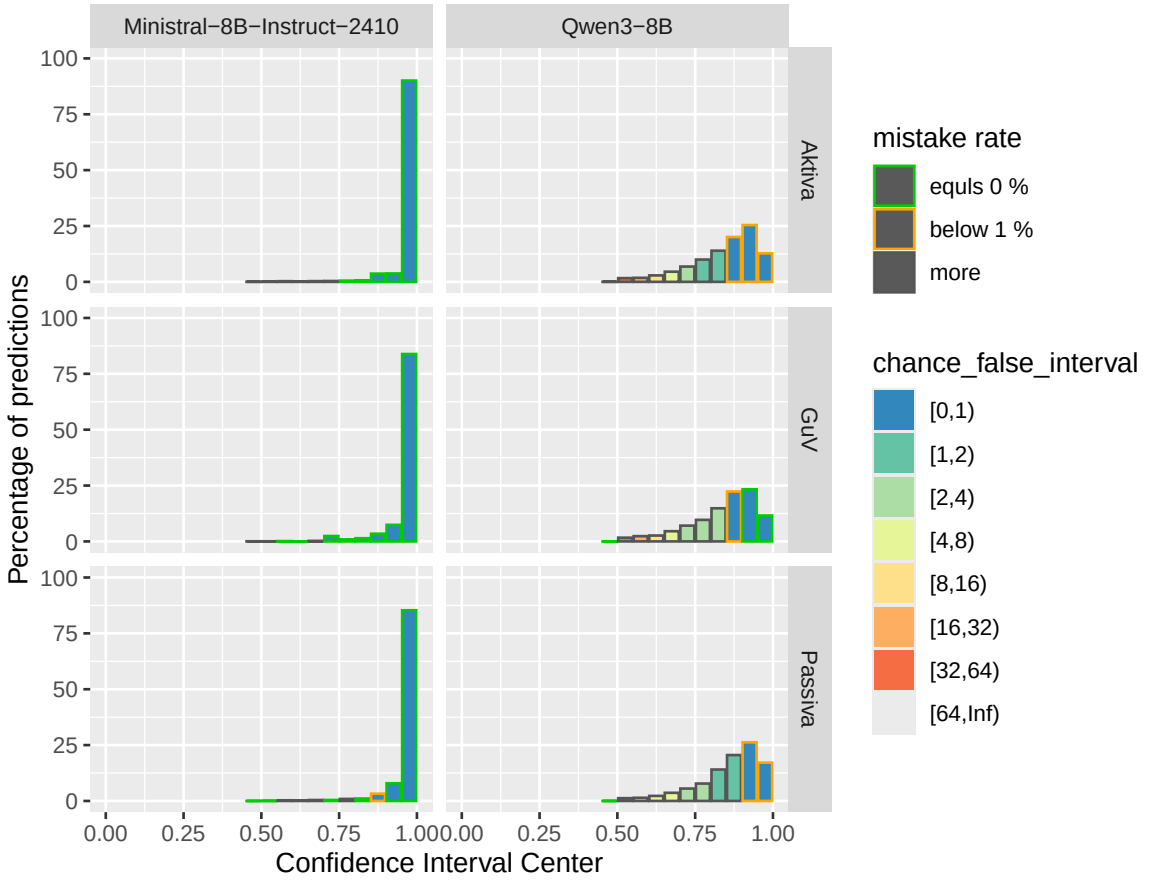


Figure 5.16: Estimating the relative frequency to find a wrong classification over different confidence intervals

Table 5.10: 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	runtime i
meta-llama	Llama4Scout17B16EInstruct	Aktiva	n_rag_examples	3	1	44
meta-llama	Llama4Scout17B16EInstruct	GuV	n_rag_examples	1	1	25
mistralai	MistralLargeInstruct2411	Passiva	n_rag_examples	1	0.99	70
meta-llama	Llama4Scout17B16EInstruct	Passiva	n_rag_examples	3	0.99	44
mistralai	MistralLargeInstruct2411	Aktiva	n_rag_examples	3	0.98	187
Qwen	Qwen2.532BInstruct	Aktiva	n_rag_examples	3	0.98	56
Qwen	Qwen3235BA22BInstruct2507	GuV	n_rag_examples	3	0.97	111
Qwen	Qwen2.572BInstruct	Passiva	n_rag_examples	1	0.97	39
mistralai	MistralLargeInstruct2411	GuV	n_rag_examples	1	0.96	70
google	gemma327bit091	Aktiva	n_rag_examples	3	0.89	42
google	gemma327bit091	Passiva	n_rag_examples	1	0.82	26
google	gemma327bit091	GuV	n_rag_examples	1	0.79	26
tiiuae	Falcon310BInstruct	GuV	n_rag_examples	1	0.71	8
tiiuae	Falcon310BInstruct	Aktiva	n_rag_examples	3	0.71	23
microsoft	phi4	Passiva	n_rag_examples	2	0.67	16
microsoft	phi4	Aktiva	n_random_examples	1	0.6	4
tiiuae	Falcon310BInstruct	Passiva	top_n_rag_examples	3	0.59	4
microsoft	phi4	GuV	n_rag_examples	1	0.46	17

Table 5.11: 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 s
mistralai	Ministral8BInstruct2410	Aktiva	n_rag_examples	1	0.98	686
mistralai	Ministral8BInstruct2410	Passiva	top_n_rag_examples	3	0.96	279
mistralai	Ministral8BInstruct2410	GuV	top_n_rag_examples	3	0.95	279
meta-llama	Llama3.18BInstruct	Passiva	n_rag_examples	1	0.95	593
Qwen	Qwen2.53BInstruct	Aktiva	n_rag_examples	1	0.86	492
meta-llama	Llama3.18BInstruct	Aktiva	top_n_rag_examples	3	0.86	269
google	gemma312bit091	Aktiva	n_rag_examples	3	0.85	2733
Qwen	Qwen2.53BInstruct	Passiva	top_n_rag_examples	1	0.83	187
Qwen	Qwen2.53BInstruct	GuV	n_rag_examples	1	0.76	492
tiiuae	Falcon310BInstruct	GuV	n_rag_examples	1	0.71	868
tiiuae	Falcon310BInstruct	Aktiva	n_rag_examples	3	0.71	2393
google	gemma312bit091	Passiva	n_rag_examples	3	0.69	2733
microsoft	phi4	Passiva	n_rag_examples	2	0.67	1660
meta-llama	Llama3.18BInstruct	GuV	top_n_rag_examples	1	0.65	205
microsoft	phi4	Aktiva	n_random_examples	1	0.6	493
tiiuae	Falcon310BInstruct	Passiva	top_n_rag_examples	3	0.59	494
google	gemma312bit091	GuV	top_n_rag_examples	1	0.47	232
microsoft	phi4	GuV	n_rag_examples	1	0.46	1725

Figure 5.17 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 B.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*

Confidence Figure 5.18 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[19].

Figure 5.19 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[20], Ministral-8B-Instruct-2410 with *3_rag_examples*. One can see there are some wrong classifications for the minority classes as well.

[19]It might be possible to request the n most probable answers to get confidence scores for all different predictions. But this was not investigated.

[20]The best performance results with *top_11_rag_examples* but the plot was less interesting and its F1 score was not listed in Table 5.11.

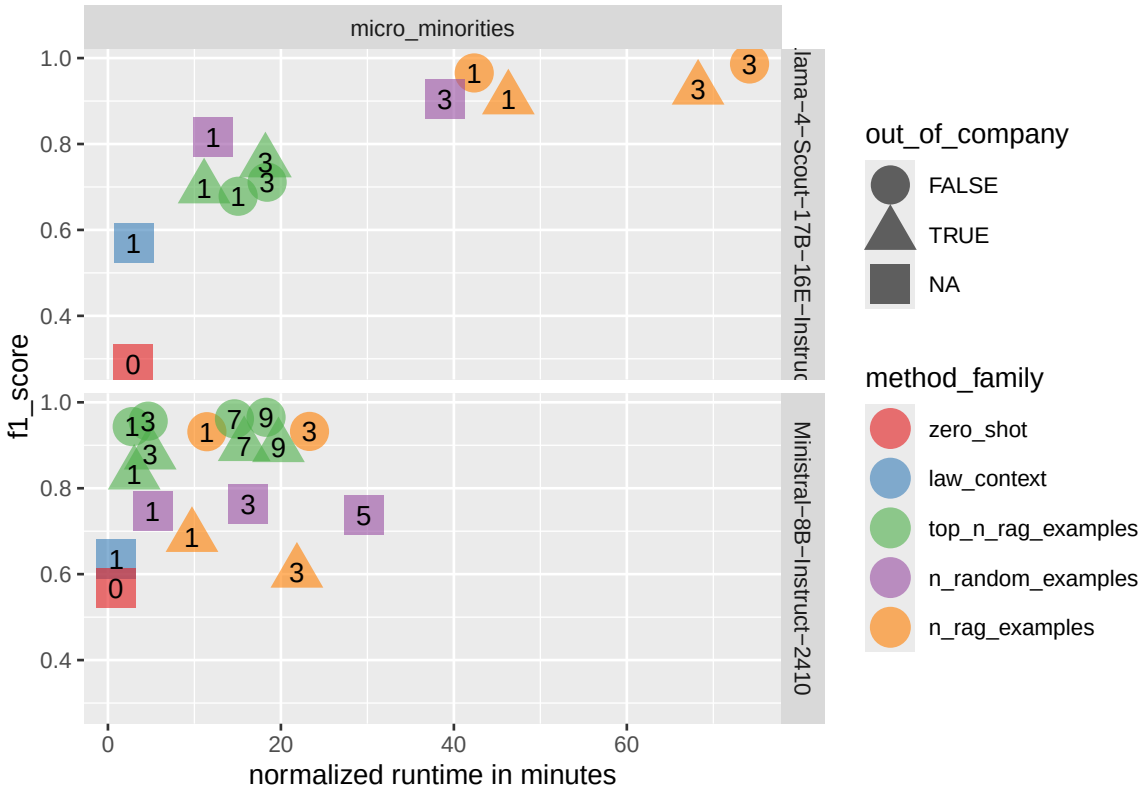


Figure 5.17: Comparing F1 score micro averaged for the minority classes for two models over their normalized runtime.

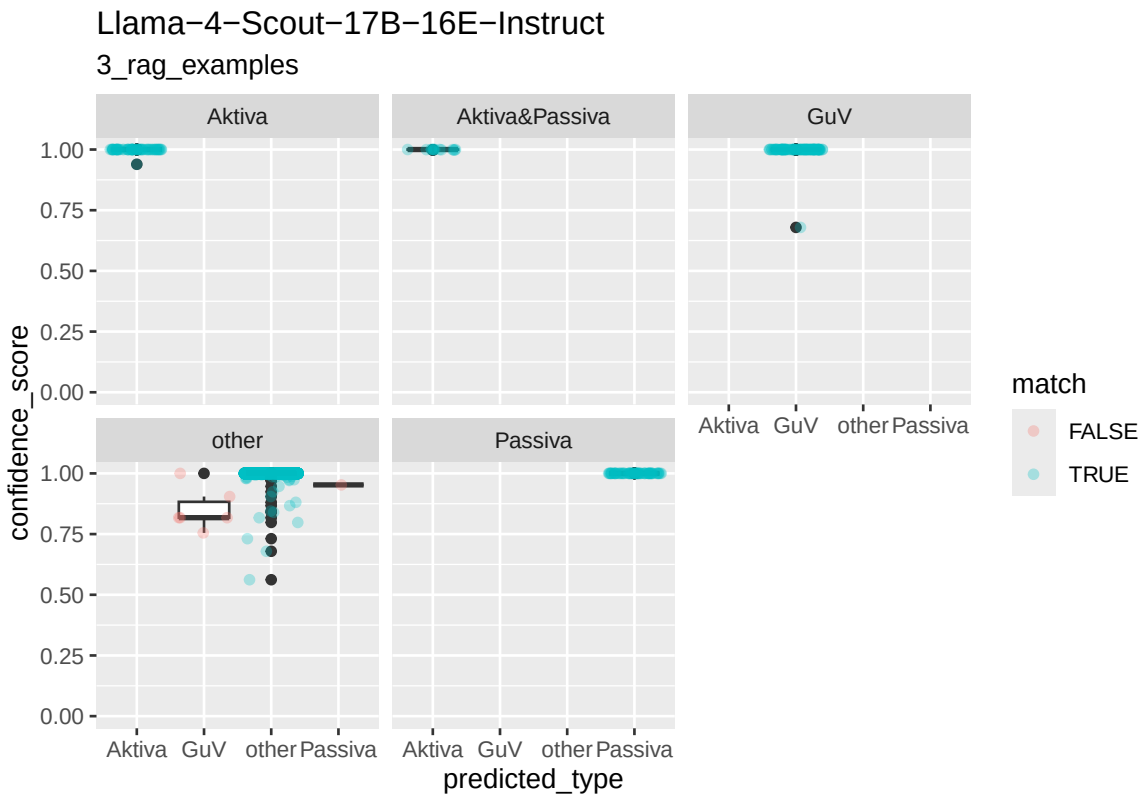


Figure 5.18: Showing the reported confidence scores for all predictions of Llama 4 Scout grouped by the true target type. Errors have only been made within the majority class.

Especially, the **Passiva** target type is often classified as *other*. This is problematic for a smooth workflow (see discussion chapter?)

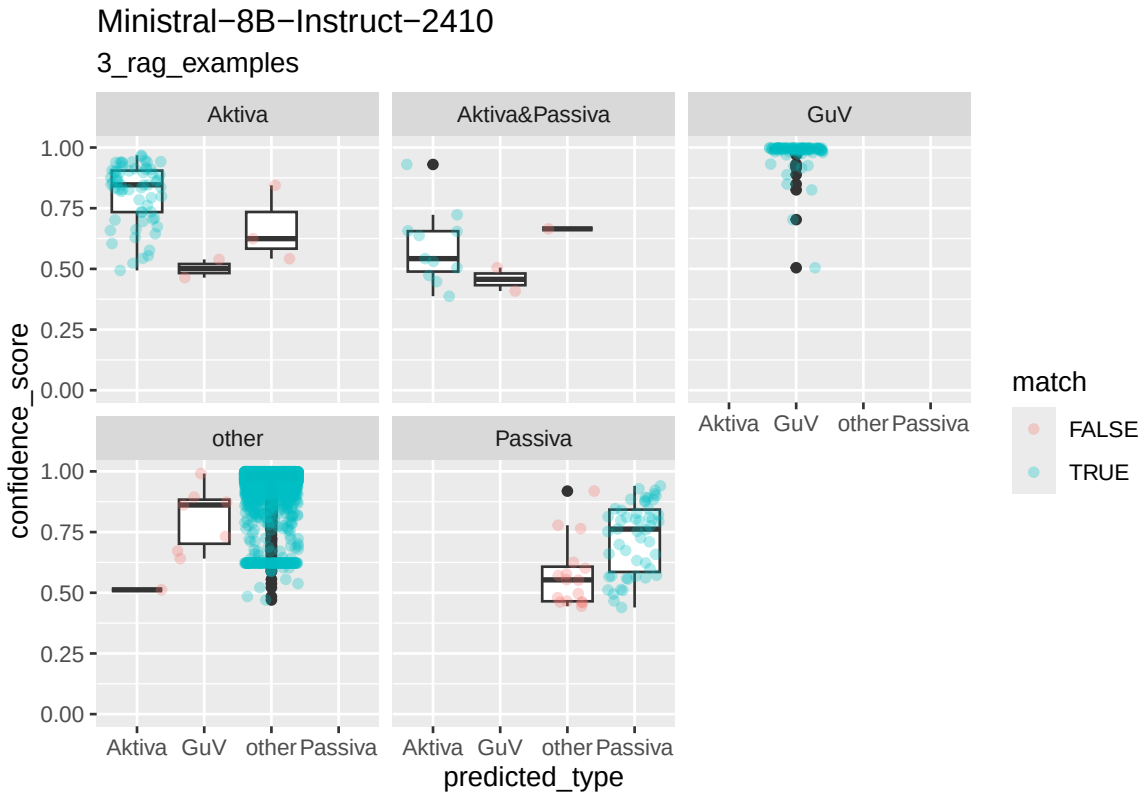


Figure 5.19: Showing the reported confidence scores for all predictions of Ministral 8B grouped by the true target type. Errors have only been made within the majority class.

Figure 5.20 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.

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 C.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 5.21 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 5.22 shows, that there is almost no area, where the empirical rate of wrong classifications is zero[21]. 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 %.

[21]The size of intervals has been narrowed down to 0.1 % and still there was no range without wrong classification for Llama 4 Scout.

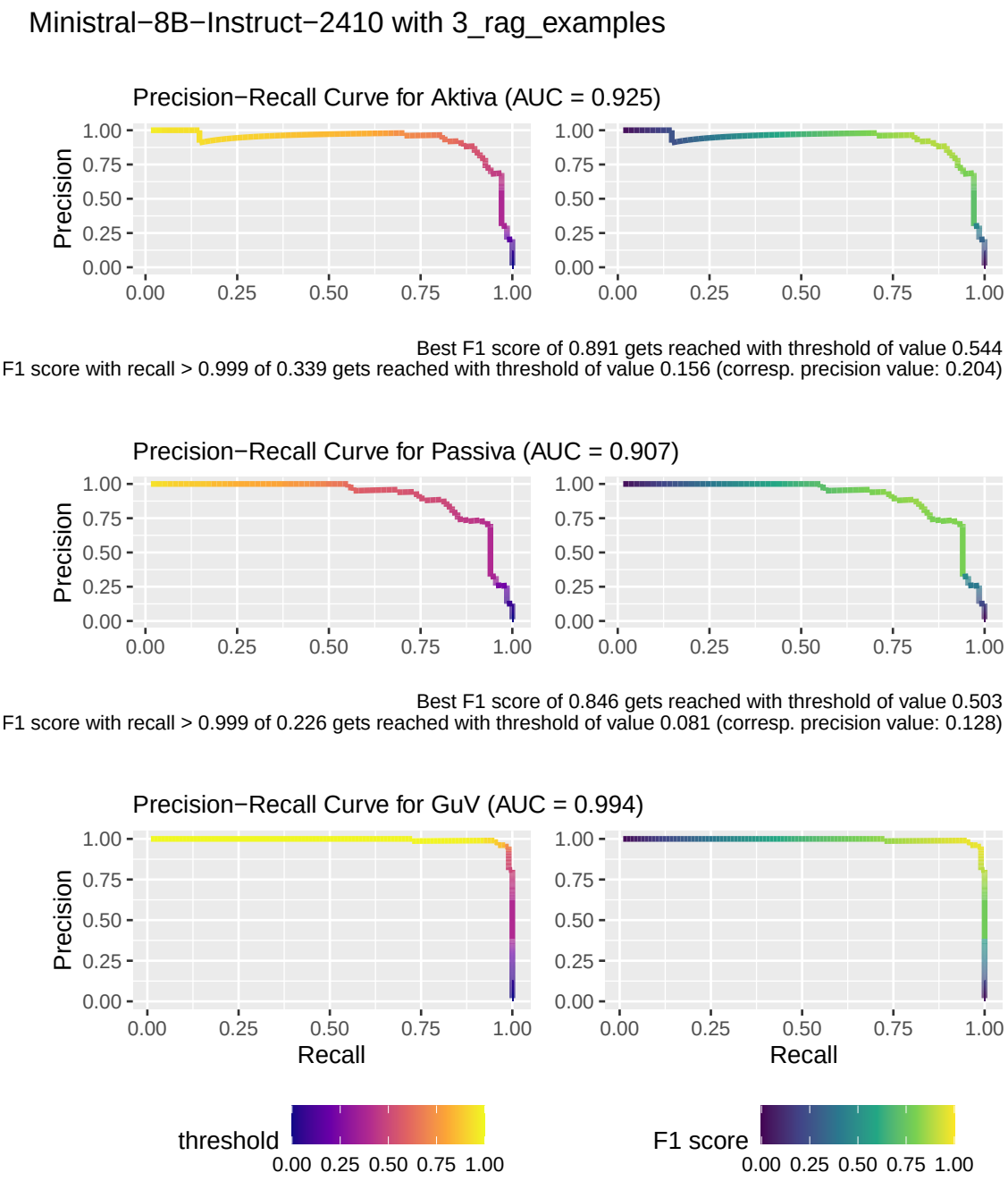


Figure 5.20: Showing the precsion-recall-curve for Ministral-8B-Instruct-2410.

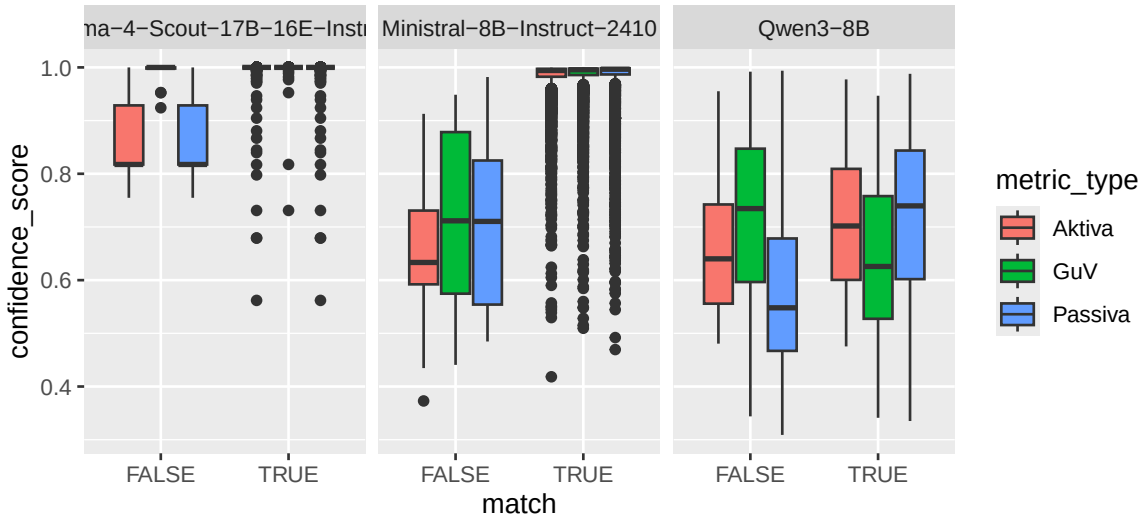


Figure 5.21: Comparing the reported confidence scores for the multi-class page identification task for the Mistral and Qwen 3 with 8B parameters.

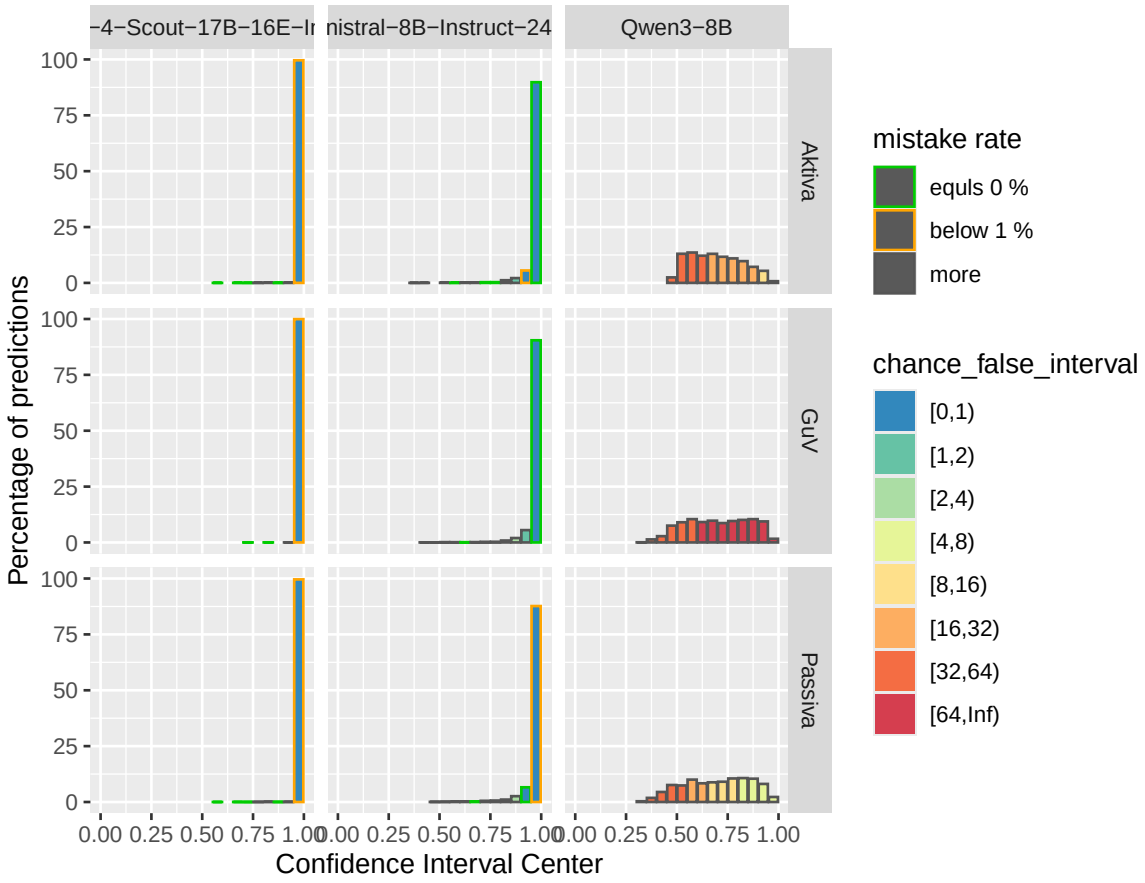


Figure 5.22: Estimating the relative frequency to find a wrong classification over different confidence intervals for the multi-class classification task.

5.1.4 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 (Frequency-Inverse Document Frequency) - 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[22]. A single random forest is trained because the density of floats and specific words is assumed to be similar. The actual type of the 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 5.23 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 5.24 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 5.25 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.

Fianlly, figure 5.26 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.

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

5.1

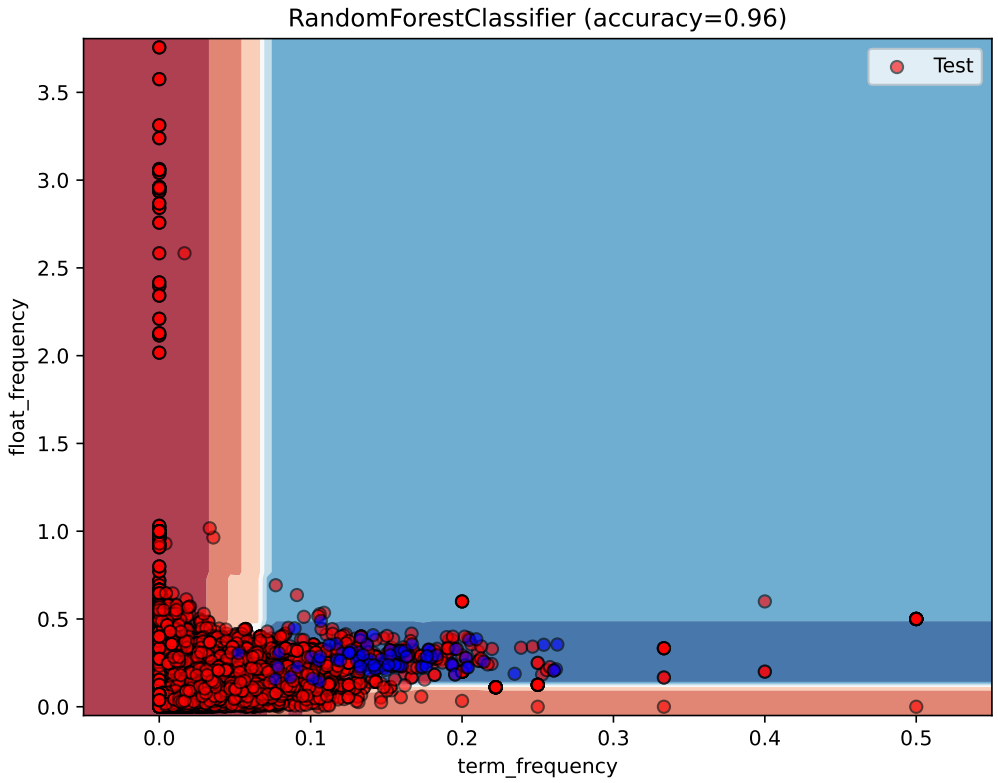


Figure 5.23: Classification map showing which score a data point gets based on its term and float frequency and which type the data points in the test dataset actually have.

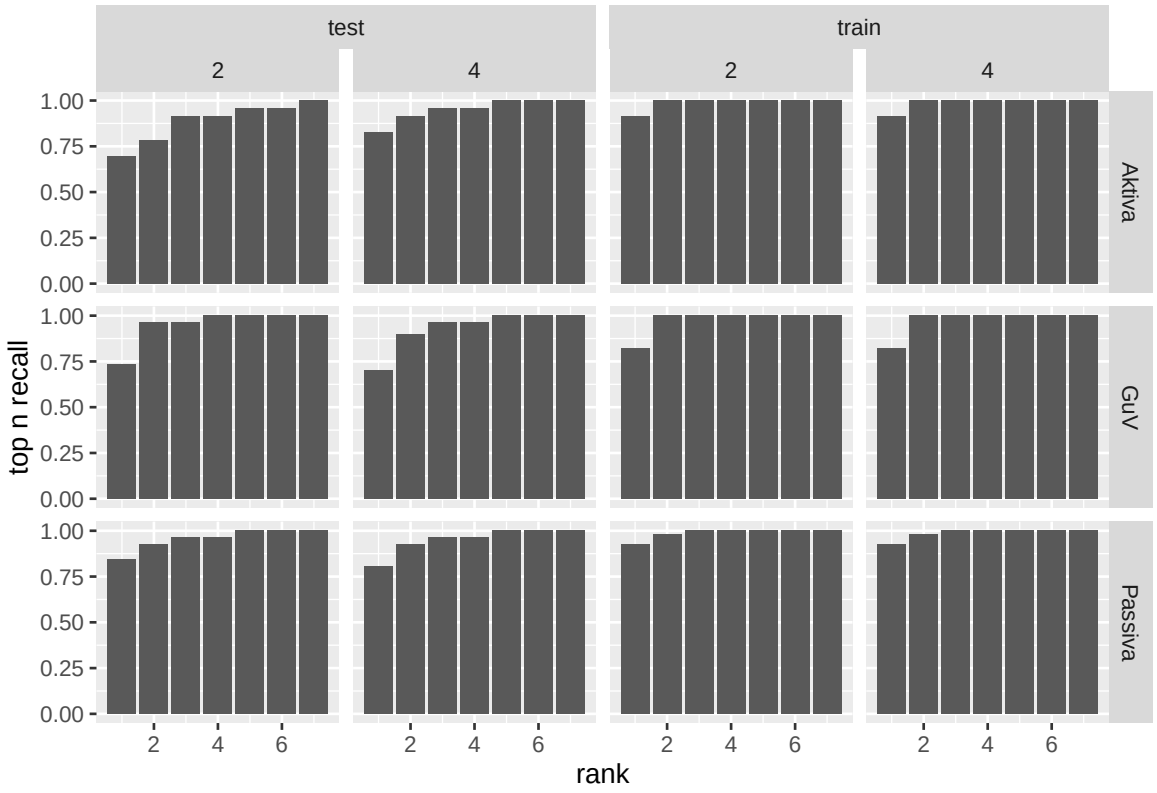


Figure 5.24: Comparing the top n recall on training and test dataset among the random forest with two and four predictors.

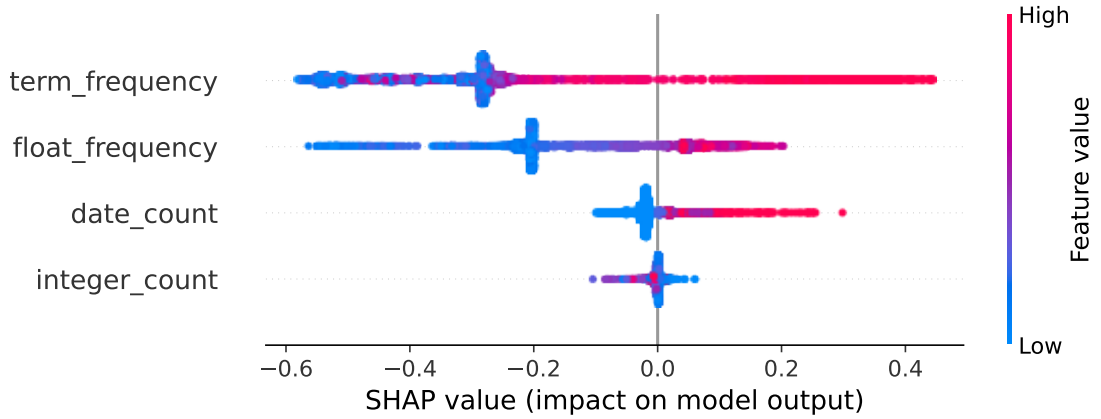


Figure 5.25: Beeswarm plot of SHAP importance values for the four predictors of the second random forest classifier.

5.1

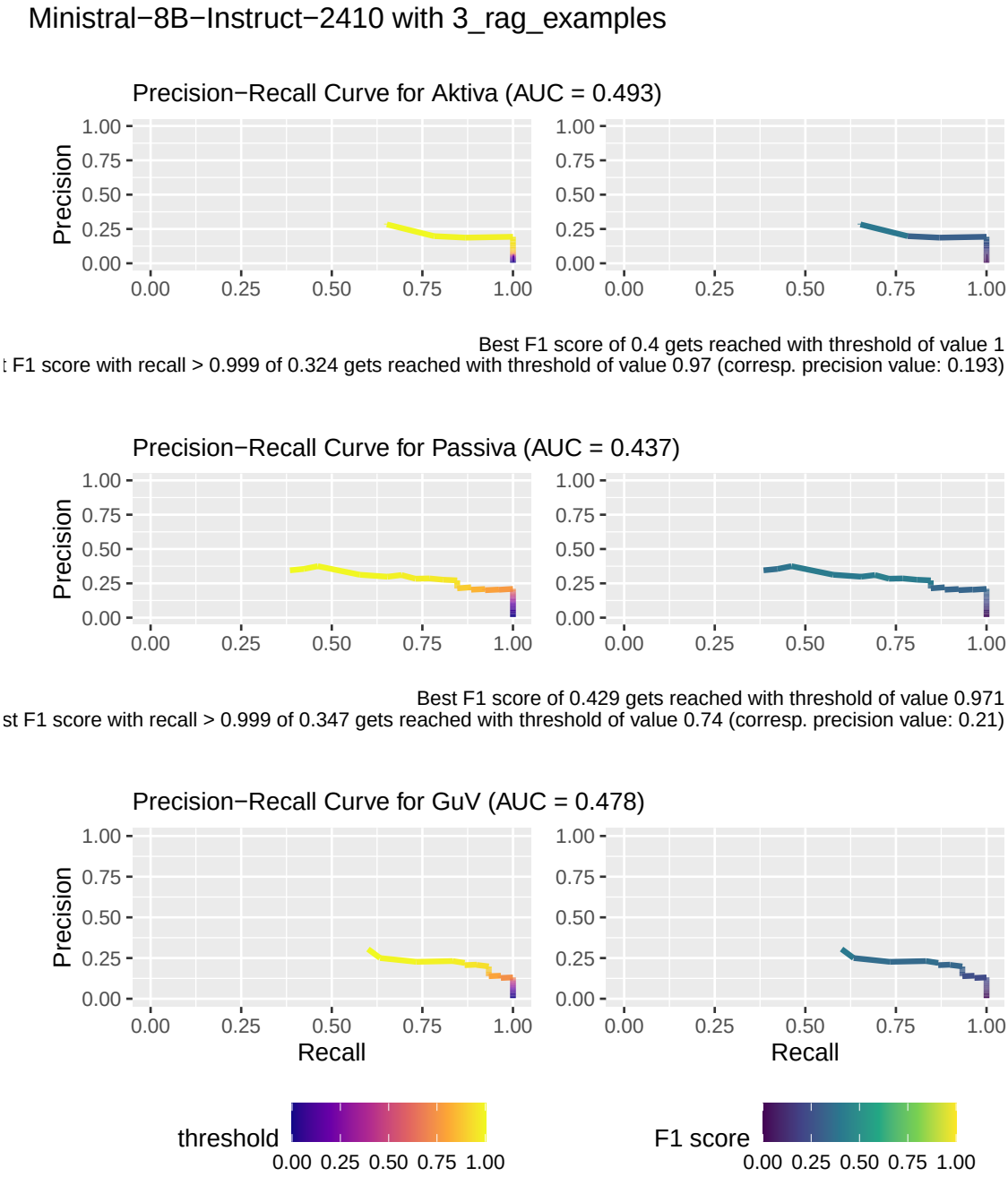


Figure 5.26: Showing the precsion-recall-curve for the random forest with four predictors.

Table 5.12: Comparing page identification perfromance among all four approaches.

approach	strategy	type	precision	recall	F1
Regex	exhaustive	Aktiva	0.132	0.997	0.233
Regex	exhaustive restricted	GuV	0.21	{1}	0.35
Regex	exhaustive	Passiva	0.13	0.993	0.23
TOC	machine readable	Aktiva	0.6	0.256	0.359
TOC	machine readable	GuV	0.446	0.245	0.316
TOC	machine readable	Passiva	0.5	0.21	0.296
TOC	combi	Aktiva	0.338	0.268	0.299
TOC	combi	GuV	0.378	0.363	0.37
TOC	combi	Passiva	0.281	0.222	0.248
LLM binary	Ministral8BInstruct2410, 3_rag_examples	Aktiva	0.906	0.939	0.922
LLM binary	Ministral8BInstruct2410, 3_rag_examples	GuV	0.981	0.99	0.985
LLM binary	Ministral8BInstruct2410, 3_rag_examples	Passiva	0.937	0.914	0.925
LLM multiclass	Ministral8BInstruct2410, 3_rag_examples	Aktiva	0.987	0.937	0.961
LLM multiclass	Ministral8BInstruct2410, 3_rag_examples	GuV	0.903	{1}	0.949
LLM multiclass	Ministral8BInstruct2410, 3_rag_examples	Passiva	{1}	0.761	0.864
LLM multiclass	Llama4Scout17B16EInstruct, 3_rag_examples	Aktiva	{1}	{1}	{1}
LLM multiclass	Llama4Scout17B16EInstruct, 3_rag_examples	GuV	0.944	{1}	0.971
LLM multiclass	Llama4Scout17B16EInstruct, 3_rag_examples	Passiva	0.985	{1}	0.993
TF	high recall	Aktiva	0.193	{1}	0.324
TF	high recall	GuV	0.131	{1}	0.232
TF	high recall	Passiva	0.21	{1}	0.347
human	manual	Aktiva	NA	NA	0.981

5.1.5 Comparison

In this subsection we summarize the performnce and efficiency for all four presented approaches and compare it with the the results a human may achieve with manual labour.

Prediction performance Table 5.12 shows the best performance of achieved by the four presented approaches. The best F1 score is reached by Llama 4 Scout 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 5.2 107 **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 reached 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 other approaches' performance is way worse. Only the term-frequency approach's results could be used downstream, because we find a recall of 1.0. Table 5.13 shows the results of the top k recall for the term-frequency and LLM approaches. 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 5.14 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 of 700 W is used to calculate the energy consumption. For the other approaches, running on my laptop (see section A.1) 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 5.13: Comparing the top k recall for the termfrequency and LLM approaches.

approach	strategy	type	top 1 recall	k for full recall
LLM binary	Ministral8BInstruct2410, 3_rag_examples	Aktiva	0.959	2
LLM binary	Ministral8BInstruct2410, 3_rag_examples	GuV	{1}	{1}
LLM binary	Ministral8BInstruct2410, 3_rag_examples	Passiva	0.932	2
LLM multiclass	Ministral8BInstruct2410, 3_rag_examples	Aktiva	0.932	3
LLM multiclass	Ministral8BInstruct2410, 3_rag_examples	GuV	{1}	{1}
LLM multiclass	Ministral8BInstruct2410, 3_rag_examples	Passiva	0.824	3
LLM multiclass	Llama4Scout17B16EInstruct, 3_rag_examples	Aktiva	{1}	{1}
LLM multiclass	Llama4Scout17B16EInstruct, 3_rag_examples	GuV	{1}	{1}
LLM multiclass	Llama4Scout17B16EInstruct, 3_rag_examples	Passiva	0.973	2
TF	high recall	Aktiva	0.826	5
TF	high recall	GuV	0.7	5
TF	high recall	Passiva	0.808	5

5.1

Table 5.14 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. The fastest and least energy consuming strategy 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 the results of the multi-class strategy are good enough as well. In both strategies the k required for perfect recall is three, using the Ministral-8B-Instruct model[23].

For the TOC approach LLMs are used as well, but they process far less of the documents pages. This can be achieved for the good performing classification strategies by combining the term-frequency approach with the LLM approach.

The manual approach is the slowest. For the benchmark ten random documents were processed by us. We used the TOC and the search function to find key words like **Aktiva** or **Bilanz**. Anyhow, its almost as fast as the multi-classification using Llama 4 Scout. But it requires eight times less energy. Comparing it to Ministral-8B-Instruct it take three times longer but consumes less then half of the energy.

Not taken into account fo this comparison are factors like

- costs to buy and maintain hardware (i.e. a GPU cluster).
- higher costs per runtime if the LLM compute is purchased from cloud providers. CLOUD: price if LLM is in the cloud <– print tokens used
  - four classes (3 random examples): 11 k tokens
  - binary (3 random examples): 6.5 k 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).

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

Table 5.14: Comparing page identification efficiency among all four approaches.

approach	strategy	runtime per document in s	energy in J	costs in CENT
Regex	exhaustive	{0.005}	{0.151}	
TOC	machine readable	0.202	141.58	
TOC	text based	1.939	1357.534	
LLM binary	Ministral8BInstruct2410, 3_rag_examples	35.851	25095.946	
LLM multiclass	Ministral8BInstruct2410, 3_rag_examples	18.905	13233.784	
LLM multiclass	Llama4Scout17B16EInstruct, 3_rag_examples	60.149	42104.054	
TF	high recall	0.138	3.859	
human	manual	61	5368	

- the energy consumed with the food humans eat.

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

5.2 Table 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 are using 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 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[24].

Using a strict schema has advantages 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 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 5.27 shows how many **Aktiva** tables are used for all tasks in this subsection, that use real data instead of synthetic data.

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 (see subsection 5.2.2.2).

[24]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.

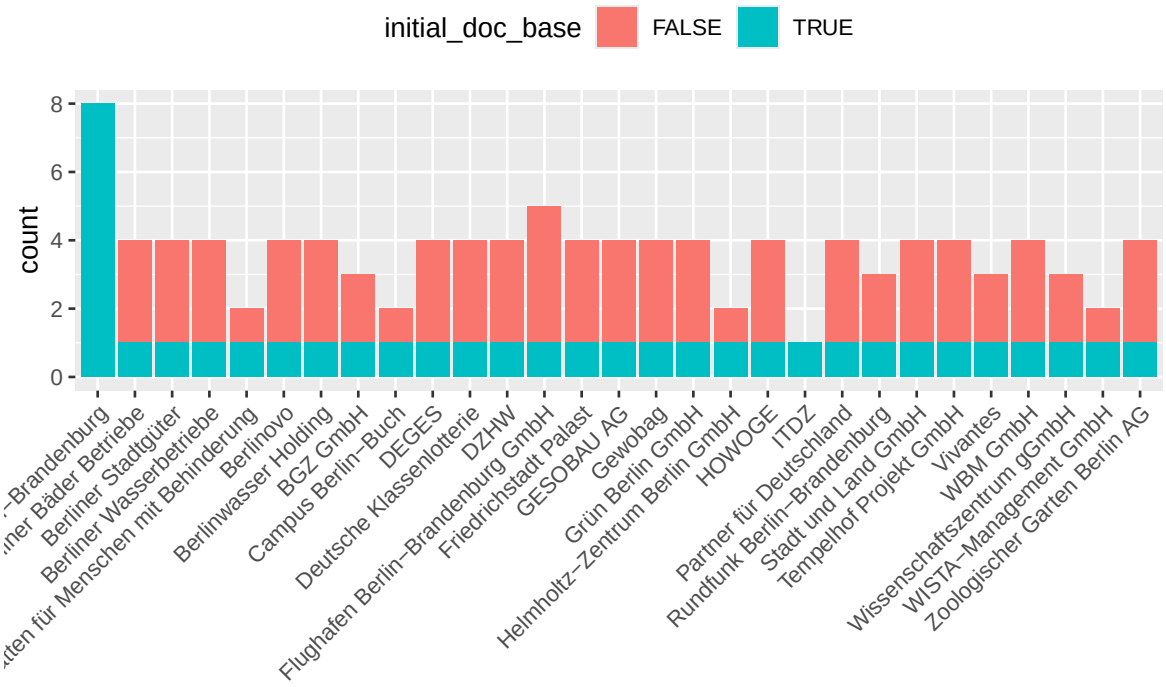


Figure 5.27: Showing the number of documents used for the table extraction task. The number of Aktiva tables is equal to the number documents.

5.2.1 Baseline: Regex

The baseline for the table extraction task is set by an approach using regular expressions on the text extract. Figure 5.28 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 C.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 5.28 shows, that the regex approach performs better[25] 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.

[25] A comparison of the numeric values over all methods can be found in section 5.2.3.

Table 5.15: 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 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:

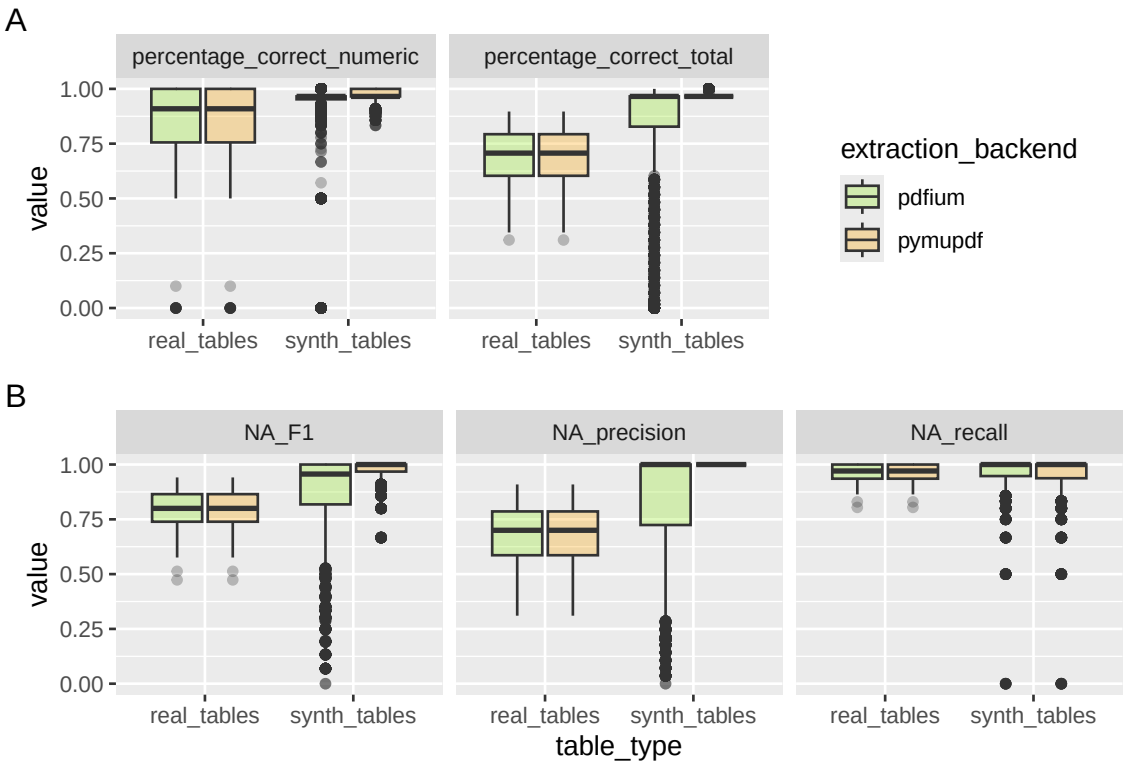


Figure 5.28: Performance overall and on numeric value extraction with regular expressions.

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 5.16 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 5.16: 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

Table 5.16 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 C.10

Synthetic dataset Table 5.17 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 5.29 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 C.12.

Furthermore, does the number of columns is have a positive effect overall. Figure 5.29 B shows, that this effect has inverse direction for the two libraries.

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*.

5.2.2 Extraction with LLMs

where to put?:

- GESOBAU AG and WBM GmbH bad, text selection with mouse odd

Table 5.17: 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

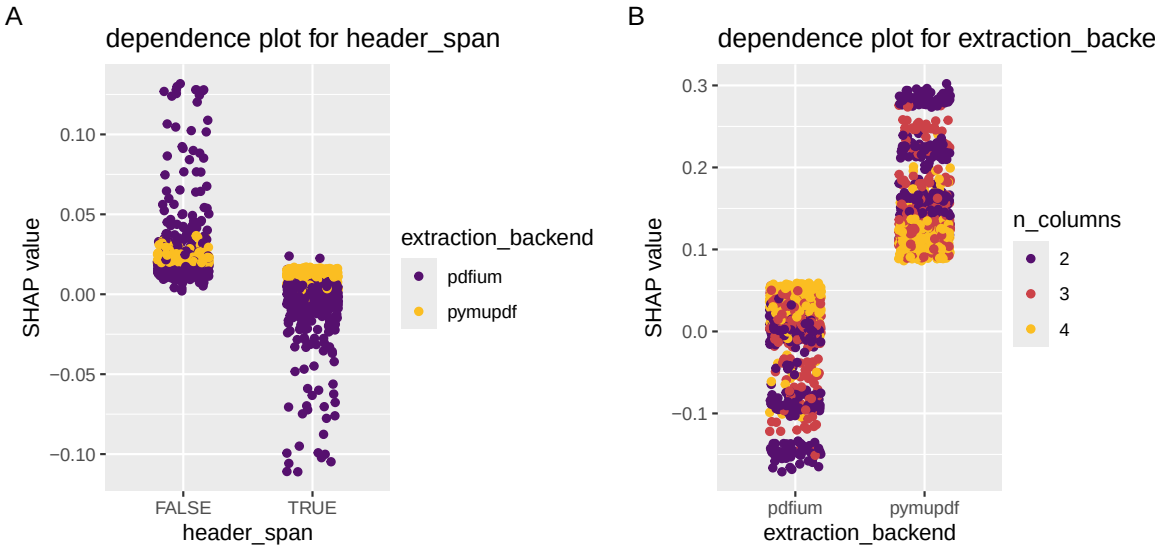
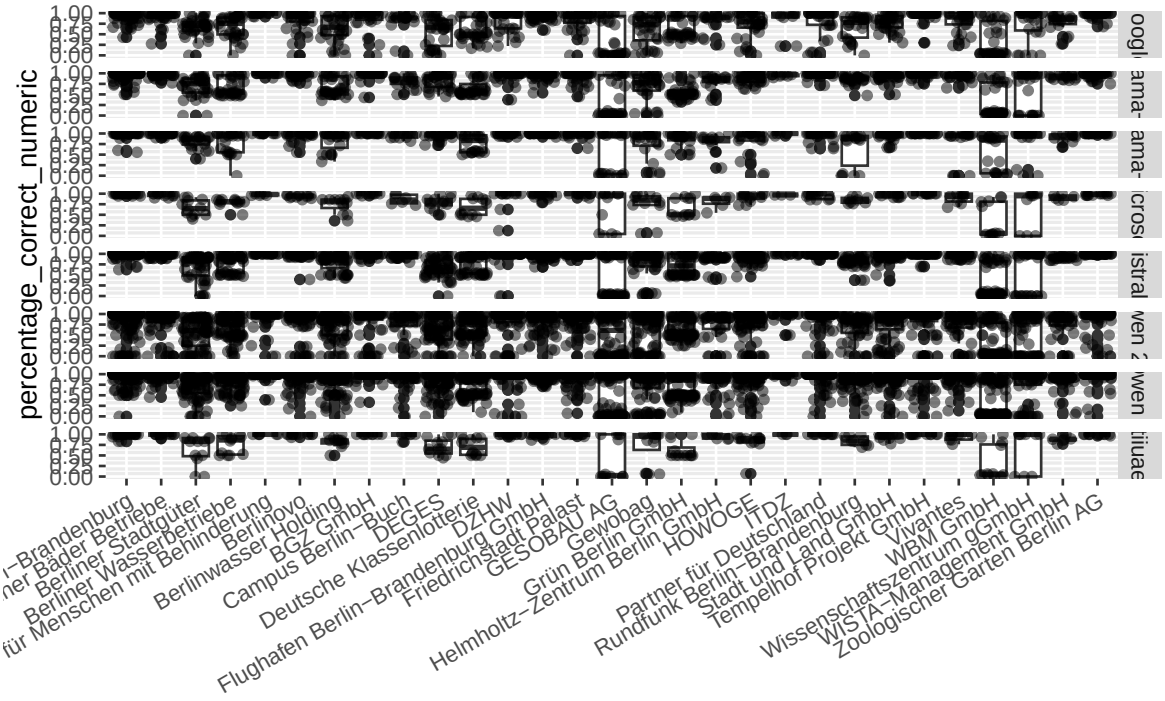


Figure 5.29: Showing the influence of the extraxtion library on the numeric text extraction task with synthetic data for the percentage of correct numeric predictions.

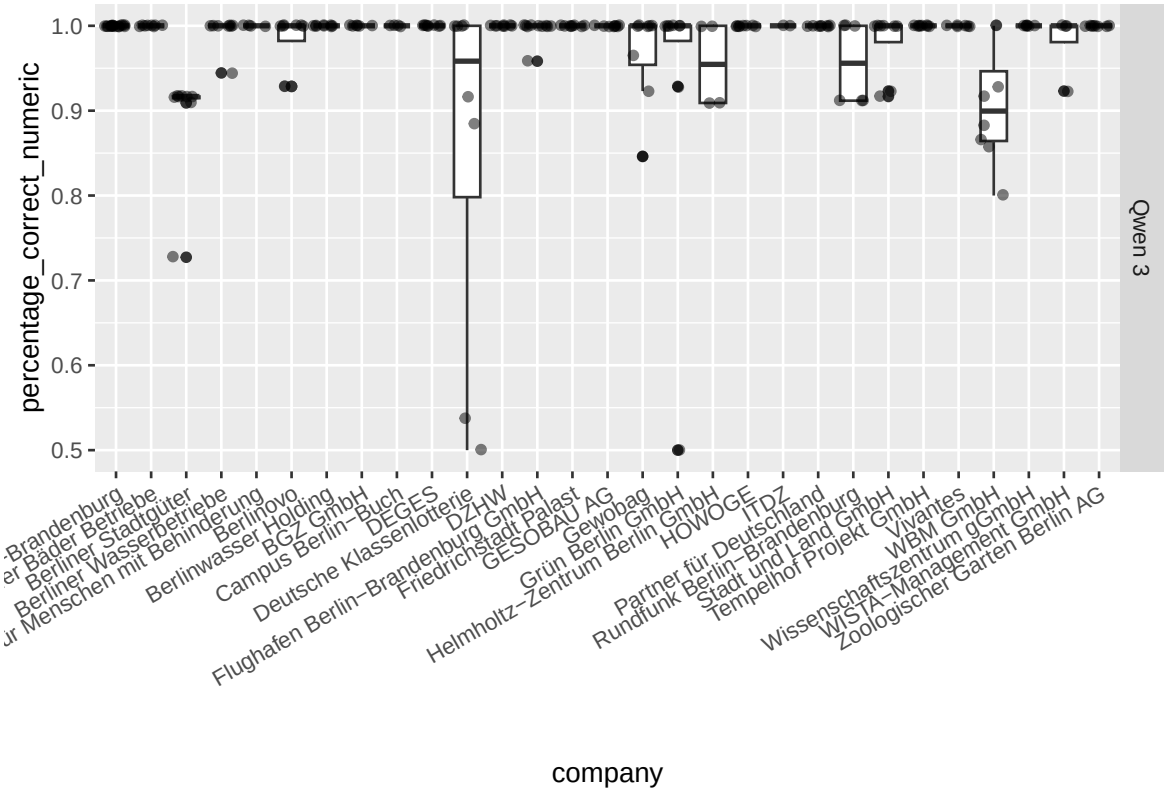
- shift median upwards for GESOBAU: Qwen3, mitral, microsoft, Llama
- shift median upwards for WBM: Llama 4

```
df_real_table_extraction %>% filter(!str_detect(model, "oss")) %>%
  filter(n_examples == 3) %>%
  mutate(.before = 1, company = map_chr(filepath, ~str_split(str_split(., "/")[[1]][5],
  ↪  "__")[[1]][1])) %>%
  group_by(company) %>%
  ggplot() +
  geom_boxplot(aes(x = company, y = percentage_correct_numeric)) +
  geom_jitter(aes(x = company, y = percentage_correct_numeric), alpha= .5) +
  scale_x_discrete(guide = guide_axis(angle = 30)) +
  facet_grid(model_family ~ .)
```

5.2




```
df_real_table_extraction %>% filter(str_detect(model, "235B")) %>%
  filter(n_examples == 5, method_family == "top_n_rag_examples") %>%
  mutate(.before = 1, company = map_chr(filepath, ~str_split(str_split(., "/")[[1]][5],
  ↪  "__")[[1]][1])) %>%
  group_by(company) %>%
  ggplot() +
  geom_boxplot(aes(x = company, y = percentage_correct_numeric)) +
  geom_jitter(aes(x = company, y = percentage_correct_numeric), alpha= .5) +
  scale_x_discrete(guide = guide_axis(angle = 30)) +
  facet_grid(model_family ~ .)
```



This section presents the results for the table extraction task performed with LLMs on **Aktiva** tables. Subsection 5.2.2.1 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 5.2.2.2 presents the results on synthetic **Aktiva** tables. Subsection 5.2.2.3 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 5.2.3.

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

Explain *static_example* method.

5.2.2.1 Real tables

Performance For the table extraction task 32 open source models from 9 model families have been benchmarked[26]. Table 5.18 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 5.19 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.

[26]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 5.18: Comparing best table extraction performance with real 'Aktiva' dataset for each model family

model_family	model	method_family	n_examples	parameter_count	mean_total
Qwen 3	Qwen3235BA22BInstruct2507FP8	top_n_rag_examples	5	235	0.97
mistralai	MistralLargeInstruct2411	top_n_rag_examples	5	124	0.949
Qwen 2.5	Qwen2.572BInstruct	top_n_rag_examples	5	72	0.941
Llama-4	Llama4Maverick17B128EInstructFP8	top_n_rag_examples	1	17	0.939
Llama-3	Llama3.170BInstruct	top_n_rag_examples	5	70	0.926
microsoft	phi4	top_n_rag_examples	2	15	0.909
openai	gptoss120b	top_n_rag_examples	5	120	0.904
tiiuae	Falcon310BInstruct	top_n_rag_examples	3	10	0.884
google	gemma327bit	top_n_rag_examples	3	27	0.86

Table 5.19: 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	median_to
Qwen 3	Qwen38B	top_n_rag_examples	5	8	0.927	0.9
Qwen 2.5	Qwen2.514BInstruct	top_n_rag_examples	5	14	0.925	0.9
mistralai	Ministral8BInstruct2410	top_n_rag_examples	5	8	0.895	0.9
Llama-3	Llama3.18BInstruct	top_n_rag_examples	5	8	0.832	0.8
google	gemma312bit	top_n_rag_examples	3	12	0.811	0.8

5.2

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 5.20 shows, that 3 models perform better without any guidance[27]. 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 5.21 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 C.15, C.16 and C.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

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

Table 5.20: 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
Llama4Maverick17B128EInstructFP8	{0.897}	0.922
Qwen3235BA22BInstruct2507FP8	{0.897}	{0.931}
gptoss120b	{0.897}	0.897
Qwen2.532BInstruct	NA	{0.931}
Qwen2.572BInstruct	NA	0.897
Qwen330BA3BInstruct2507	NA	0.879

Table 5.21: 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
Llama3.18BInstruct	3_random_examples	8	0.81
Llama4Maverick17B128EInstructFP8	5_random_examples	17	0.017
Qwen2.50.5BInstruct	3_random_examples	0.5	0.586
Qwen2.51.5BInstruct	3_random_examples	1.5	0.724
Qwen2.53BInstruct	3_random_examples	3	0.759
Qwen2.57BInstruct	3_random_examples	7	0.862
Qwen30.6B	3_random_examples	0.6	0.612
Qwen31.7B	3_random_examples	1.7	0.776
gemma312bit	3_random_examples	12	0.793
gemma34bit	3_random_examples	4	0.664
gptoss20b	3_random_examples	{20}	{0.897}

Table 5.22: 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
Qwen3235BA22BInstruct2507FP8	5_random_examples	0.97	1
gpt4.1	top_5_rag_examples	0.95	0.97
gpt5mini	top_5_rag_examples	0.95	0.97
Qwen330BA3BInstruct2507	top_5_rag_examples	0.92	0.97
gpt4.1mini	top_3_rag_examples	0.91	0.96
Qwen38B	top_5_rag_examples	0.89	0.93
gptoss120b	5_random_examples	0.88	0.93
gptoss20b	3_random_examples	0.84	0.9
Qwen30.6B	5_random_examples	0.65	0.65
gpt5nano	3_random_examples	0.3	0.24
gpt4.1nano	zero_shot	0.21	0.14

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 @ref(tab.table-extraction-llm-performance-total-gpt-ranking) 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 5.23 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 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[28].

Figure 5.30 shows the distribution of F1 score for up to three examples. It shows green crosses at the bottom

[28]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?

Table 5.23: 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

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 C.19, C.20 and C.21).

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

5.2

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 5.32 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[29]. Only with gpt-5-mini we had no json parsing error.

Out of company performance Table 5.24 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 C.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

Confidence Figure 5.33 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 5.34 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 3.2631144 % for numeric predictions and 1.32515 % for predicting a missing value.

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

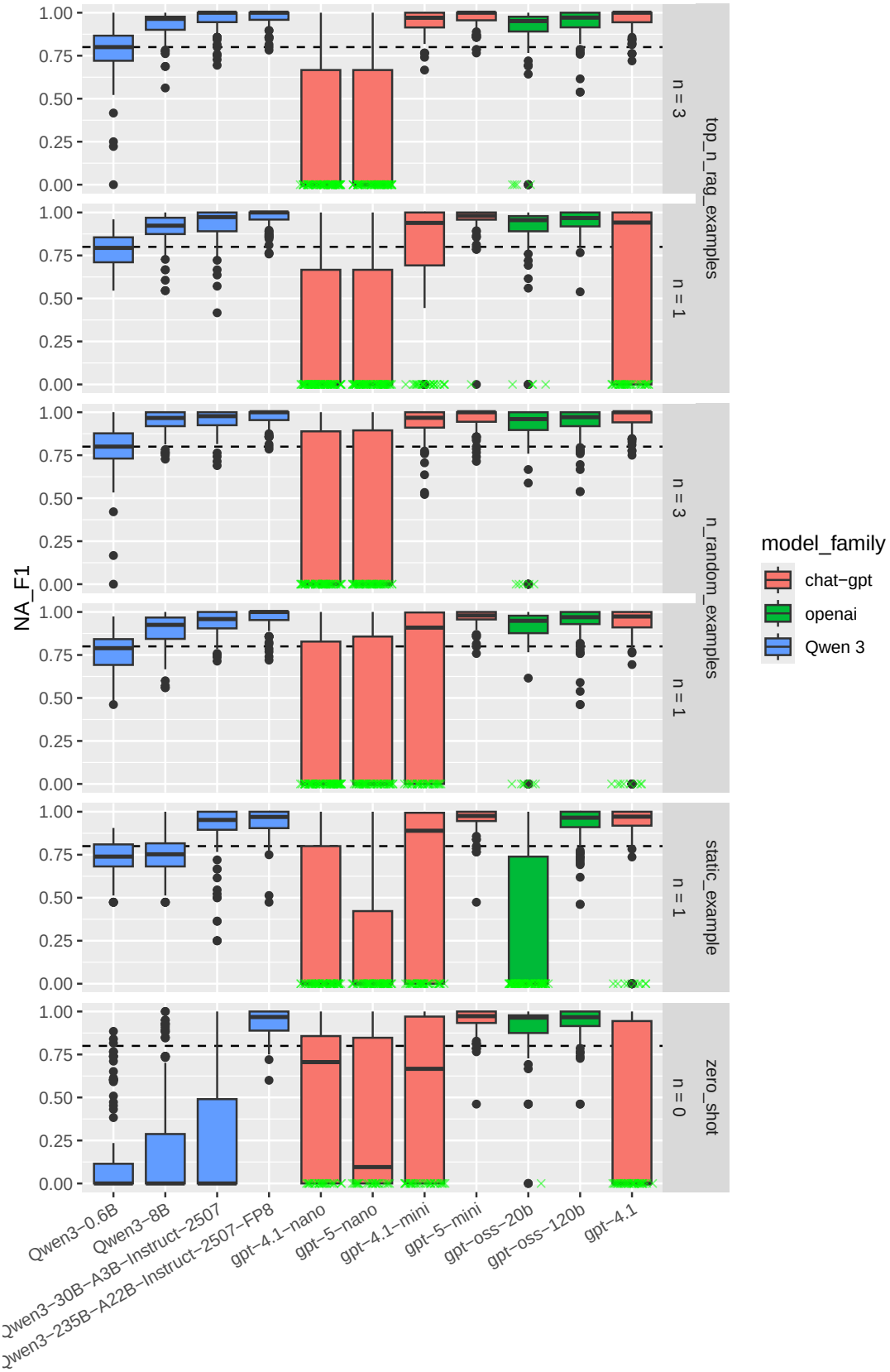


Figure 5.30: Comparing the F1 score for predicting the missingness of a value for OpenAi's LLMs with some Qwen 3 models. The green crosses indicate results where a model has predicted only numeric values even though there have been missing values.

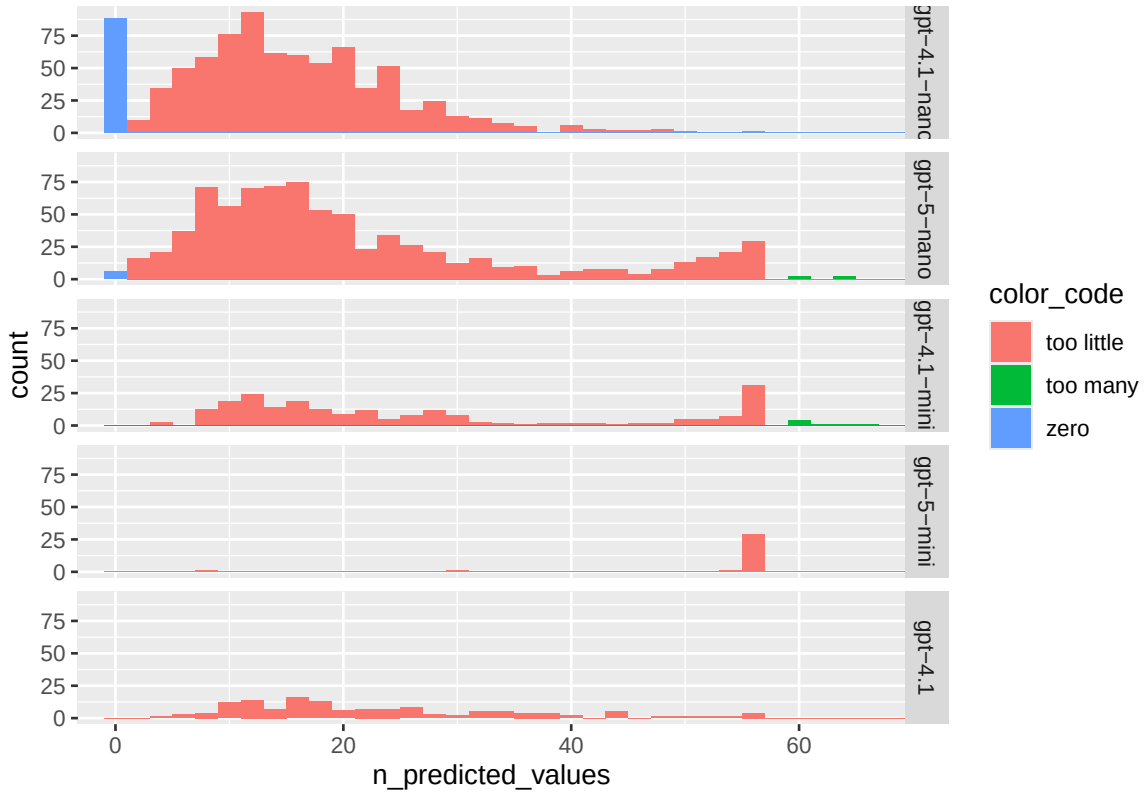


Figure 5.31: Showing the number of predictions OpenAI's models made.

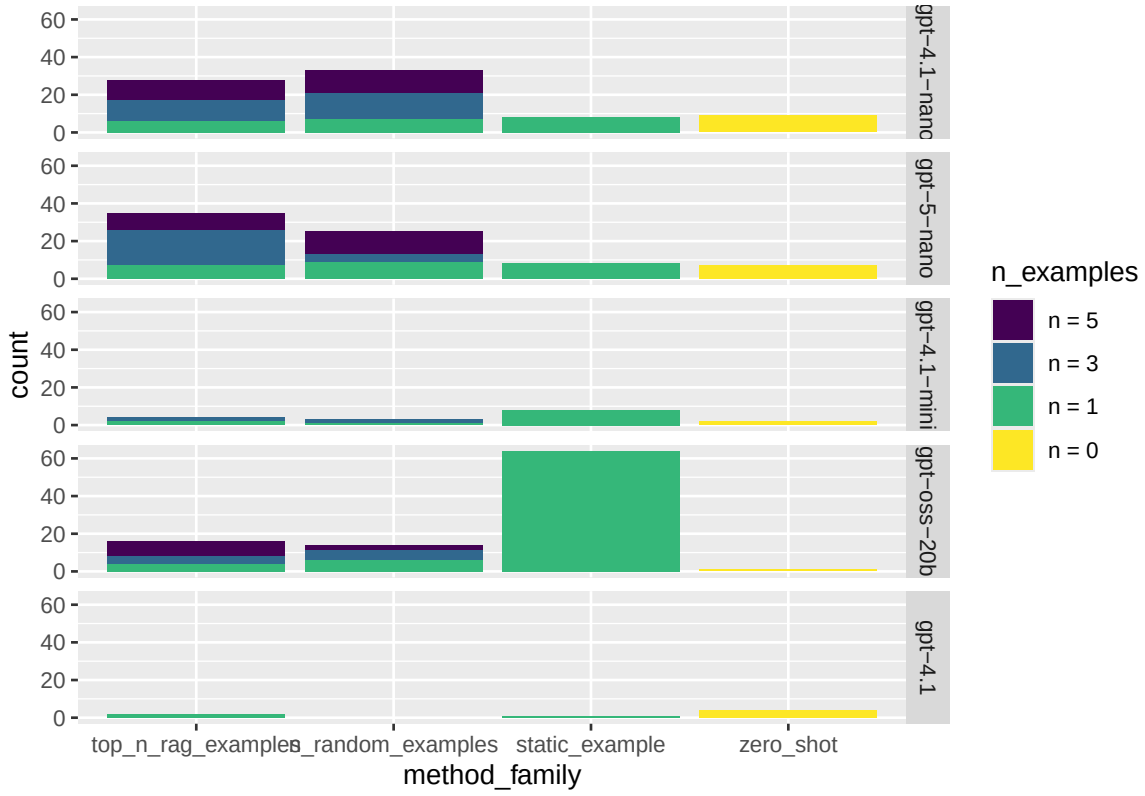


Figure 5.32: Showing the number of predictions OpenAI's models made.

5.2

Table 5.24: 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
chatgpt	0.12	0.17
Qwen 2.5	0.1	0.12
tiiuae	0.09	0.07
mistralai	0.08	0.07
Llama3	0.07	0.05
microsoft	0.07	0.06
openai	0.05	0.07
Llama4	0.03	0

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.

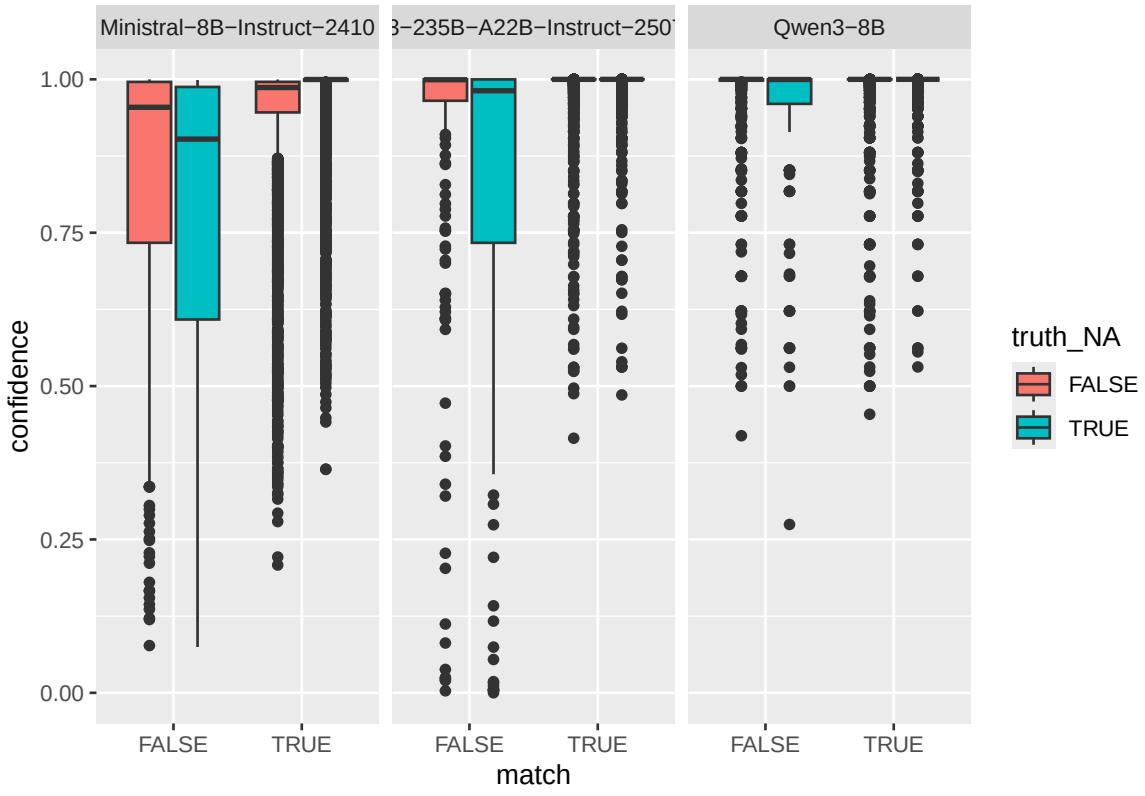


Figure 5.33: Comparing the reported confidence scores for the table extraction task on real dataset for the Mistral and Qwen 3 with 8B parameters.

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 5.25 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

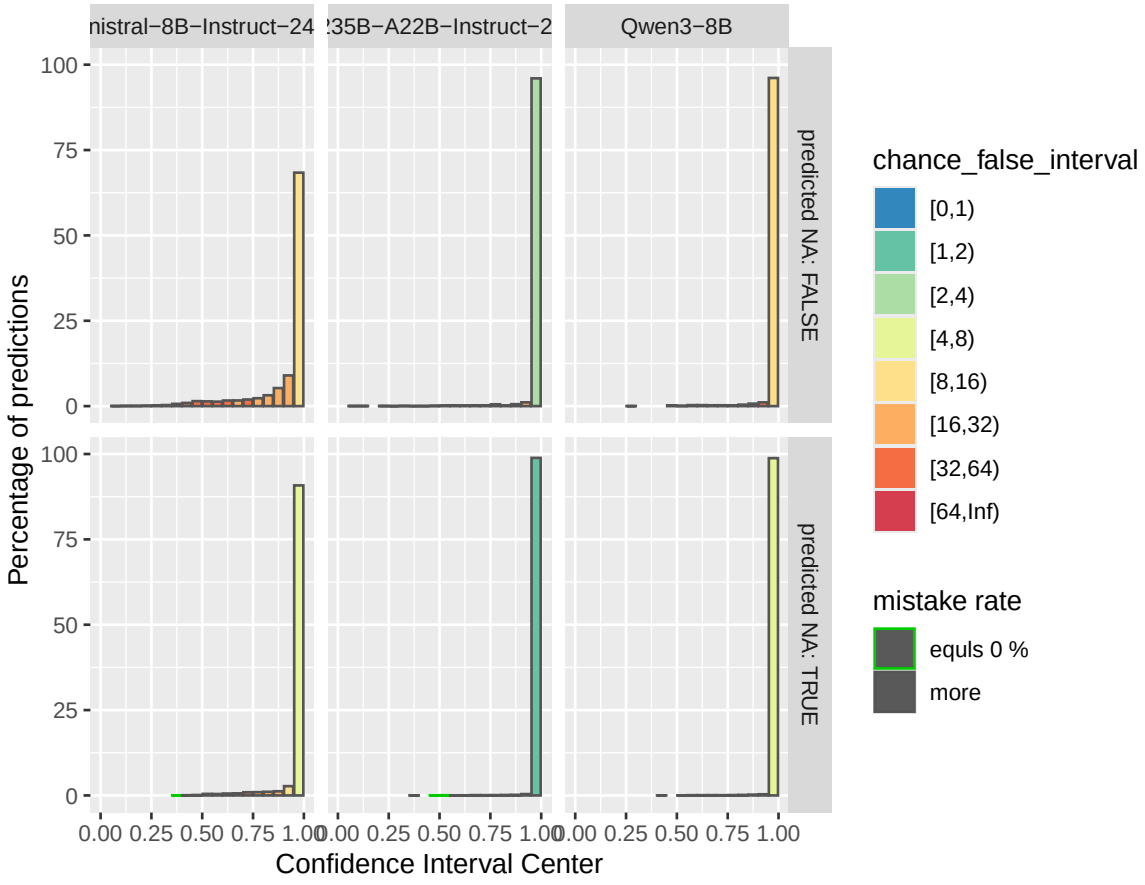


Figure 5.34: Estimating the relative frequency to find a wrong classification over different confidence intervals.

Table 5.25: Comparing the formulated hypotheses and the found results for the table extraction on real Aktiva tables the LLM approach.

	F1		% c
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_random best**
n_examples	**positive**	1 and 3 best (five bad for Llama4)	**positive**
n_columns	neutral	neutral	neutral
sum_same_lin	neutral	negative (i.e. if prev year not T€)	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	neutral
vis_separated_rows	neutral	neutral	neutral
label_length			
label			
missing			
confidence			

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 C.22

5.2.2.2 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.

Table 5.26: 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	m
Qwen 3	Qwen3235BA22BInstruct2507	top_n_rag_examples	3	0.992	{1}	
Qwen 2.5	Qwen2.572BInstruct	top_n_rag_examples	3	0.981	{1}	
mistralai	MistralLargeInstruct2411	top_n_rag_examples	5	0.981	{1}	
Llama-4	Llama4Maverick17B128EInstructFP8	top_n_rag_examples	1	0.972	{1}	
google	gemma327bit	top_n_rag_examples	5	0.928	{1}	
Llama-3	Llama3.170BInstruct	top_n_rag_examples	3	0.889	{1}	

Table 5.27: 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_runtim
Qwen 3	Qwen38B	top_n_rag_examples	3	0.954	{1}	21:39
Qwen 2.5	Qwen2.57BInstruct	top_n_rag_examples	5	0.928	{0.966}	15:26
mistralai	Ministral8BInstruct2410	top_n_rag_examples	5	0.875	{0.974}	47:2
google	gemma312bit	top_n_rag_examples	1	0.857	0.931	13:16
Llama-3	Llama3.18BInstruct	top_n_rag_examples	5	0.842	0.931	21:5

5.2

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 5.2.2. 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 5.26 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. 70 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 5.26 shows, that Qwen3-235B-A22B-Instruct performs best. Llama 4 Scout also performs very good but is three times faster. Table 5.27 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[30].

Detail:

Figure C.24 shows, that Llama 3.3 70B never manages to reduce the spread in the numeric prediction performance.

[30]When we say, it runs well, it gets rated as *okay* on the LLM Inference: VRAM & Performance Calculator.

Confidence Figure 5.35 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 C.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.

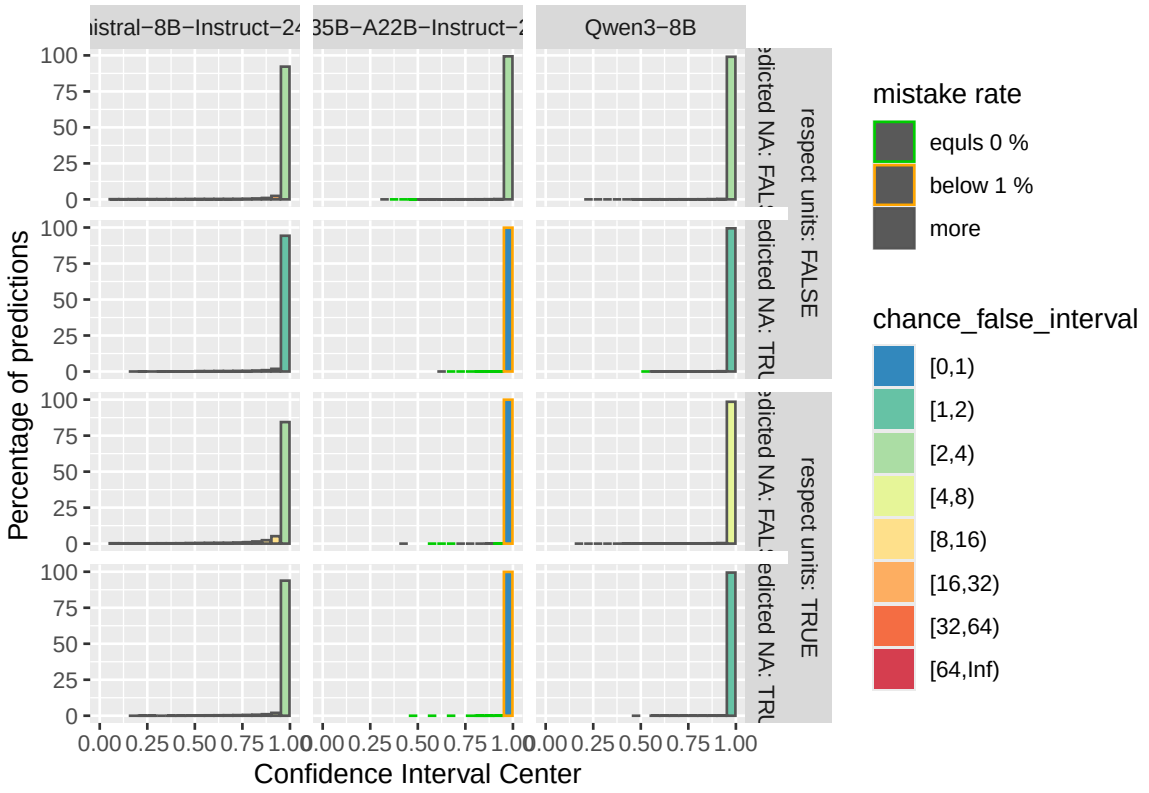


Figure 5.35: Estimating the relative frequency to find a wrong extraction result over different confidence intervals for predictions for the synthetic table extraction task.

Hypotheses Table 5.28 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 5.36 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 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 5.35 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)

Table 5.28: 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 html a
sum_same_line	neutral	neutral	negative
sum_same_line:input_format	neutral	neutral	better for htm
header_span	neutral	neutral	neutral
header_span:input_format	Can't be evaluated		Can't be evalu
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 htm
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			

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)

5.2.2.3 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.

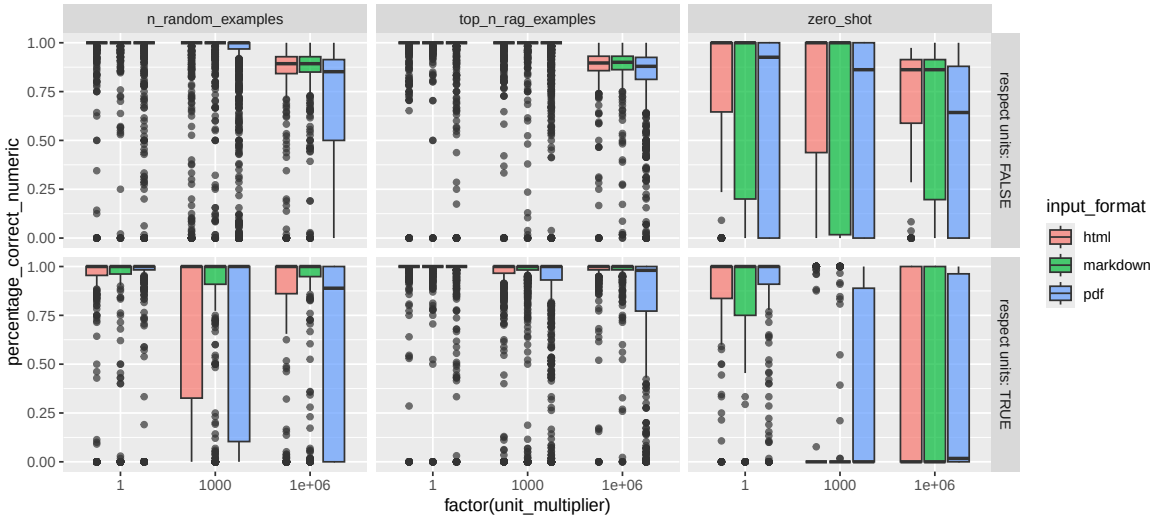


Figure 5.36: Comparing the percentage of correct extracted numeric values grouped by input format, method family and the fact, if currency should be respected.

5.2

Table 5.29: 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	delta_rate
Qwen3235BA22BInstruct2507FP8	1_random_examples	{0.966}	{0.966}	{0.897}	
Llama4Scout17B16EInstruct	5_random_examples	{0.966}	0.931	0.448	
MistralLargeInstruct2411	5_random_examples	{0.966}	0.922	0.776	
Qwen38B	5_random_examples	0.94	0.802	0.336	
Llama3.18BInstruct	5_random_examples	0.836	0.776	0.552	
Ministral8BInstruct2410	5_random_examples	0.897	0.767	0.552	
gemma327bit	3_random_examples	0.828	0.724	0.207	
gemma312bit	top_1_rag_examples	0.862	0.586	0.543	

Performance Table 5.29 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 B.2 shows that these observations are valid for the improvement with in the models independent from the selected method. Figure C.27 shows, that the improvement for using one or three synthetic examples is biggest for Qwen3-8B.

Learning to respect currency units Table 5.30 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€*.

Table 5.30: Comparing extraction performance for real Aktiva extraction task dependent on the promt addition to respect currency units

model	n_cols_T_EUR_0	n_cols_T_EUR_1	n_cols_T_EUR_2
Llama3.18BInstruct	0.03	0.02	0.01
Llama4Scout17B16EInstruct	{0}	0.28	0.89
Ministral8BInstruct2410	0.08	0	0.79
MistralLargeInstruct2411	{0}	0.39	0.9
Qwen3235BA22BInstruct2507FP8	{0}	{0.44}	{0.96}
Qwen38B	0.48	0	0.28
gemma312bit	0.03	0.17	0.48
gemma327bit	{0}	0.06	0.02

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 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 5.37 shows, that the performance of Llama 3.1 8B and gemma3 27B on colums with currency units does not change.

5.2

C.28, C.29 and C.30

Thus, synthetic data can be used to solve new tasks and substitute missing data for rare classes.

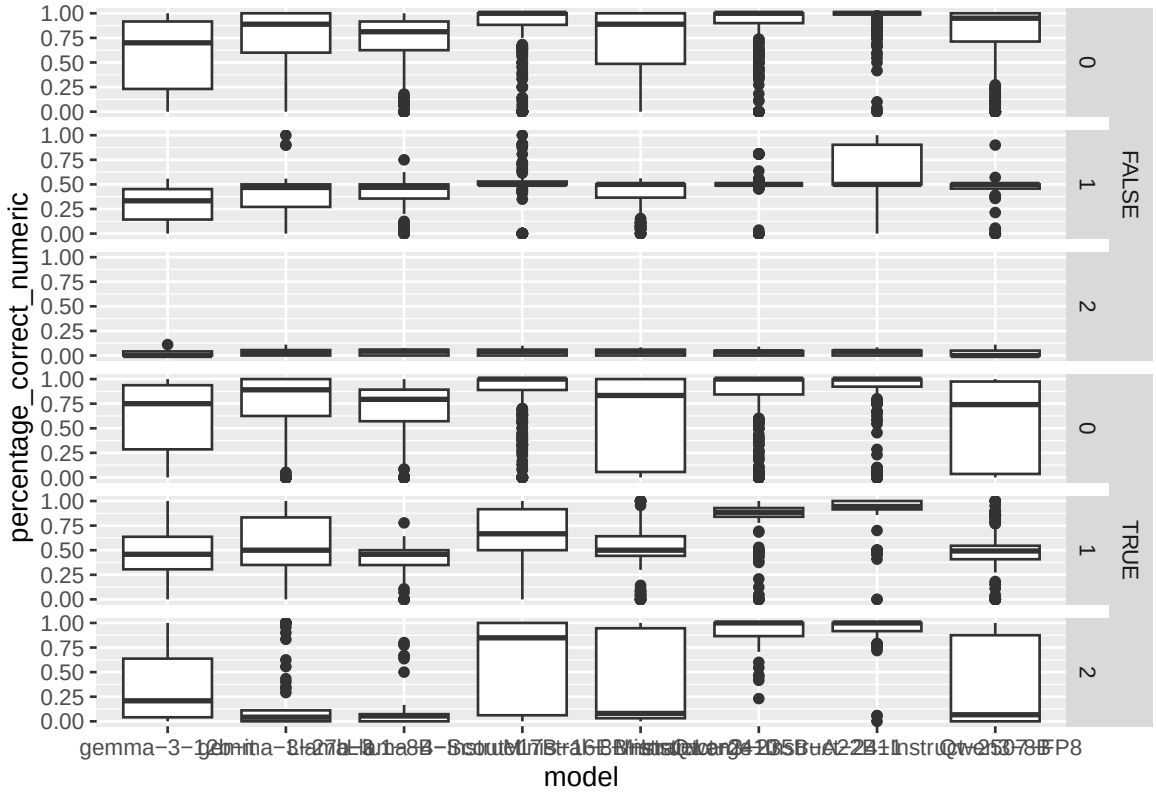


Figure 5.37: Comparing the numeric prediction performance for the hybrid approach, based on the fact, if the LLM is prompted to respect currency units.

Confidence Figure 5.38 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

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.

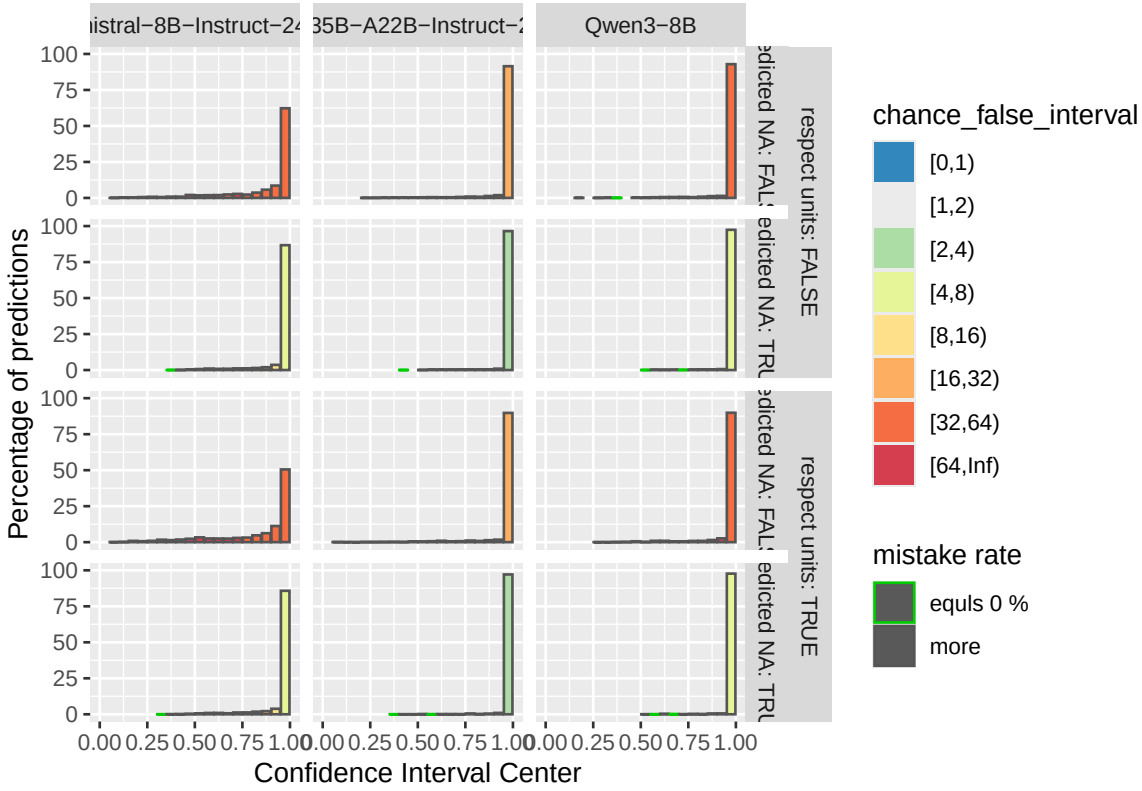


Figure 5.38: Estimating the relative frequency to find a wrong extraction result over different confidence intervals for predictions based on synthetic examples for in-context learning.

Hypotheses Table 5.31 shows only one unsupported hypothesis for a predictor with a strong effect: *method_family*. Figure 5.39 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.

5.2.3 Comparison

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.

hypotheses

Performance Table 5.32 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 %. The extraction performance may get higher, if the in-context learning examples show how to deal with columns that have a currency unit. But the strong performance observed right now is already based on implicit numeric transformations, since 19.2 % of all numeric values have *T€* as unit. Furthermore, both measures, percentage of correct numeric predictions (98.0 %) and F1 score (98.1 %) are not perfect yet and could be improved.

Table 5.31: Comparing the formulated hypotheses and the found results for the table extraction on real Aktiva tables with the hybrid LLM approach.

	F1		% corr
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 5.39: Estimating the relative frequency to find a wrong extraction result over different confidence intervals for predictions based on synthetic examples for in-context learning.

Table 5.32: Comparing the mean percentage of correct predictions total among all approaches and table types.

approach	strategy	table_type	percentage_correct_total
human	manual	real	97.6
regex		real	68.6
llm	Qwen3-235B, top_5_rag_examples	real	97.0
llm	Qwen3-8B, top_5_rag_examples	real	92.7
llm	Qwen3-235B, top_5_rag_examples, synth examples	real	91.8
regex		synth	96.9
llm	Qwen3-235B, top_5_rag_examples, respect_units	synth	99.9
llm	Qwen3-8B, top_5_rag_examples	synth	94.6

On synthetic tables its mean performance is almost perfect, if currency units get respected. The 0.1 % incorrect prediction from the PDF documents could be caused by faulty text extracts by *pdfium*. With HTML documents we find 100 % correct predictions. With Markdown documents we find 99.9 % correct predictions as well. Figure C.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. Checking the extracted values takes up to three minutes. This totals in 300 minutes prediction checking. Thus, selecting a smaller model that is finishing after 2:30 minutes is not speeding up the process a lot. Once we get a sufficient good performance with the big models the prediction checking can be dropped. This would bring th real benefit.

Hypotheses The predictor that showed 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.

Especially for the approaches that use synthetic tables show that the input format could also have a meaningful effect. It seems important to prevent erroneous text extraction and converting the extracted text in HTML might be helpful to eliminate last unclarities. But the question, if a perfect text extract would be as good as HTML or Markdown, is not answered yet.

5.2

Chapter 6

Discussion

6.1 Interpretations

6.1.0.1 Table extraction

Regular expression approach Some possible explanations for the different performance on the extracted texts are:

- a duplicated row name[1]
- numeric columns extracted separated from row names by extraction libraries
- sums in the same row as the single values[2]
- with pdfium: missing white space[3]
- with pdfium: random line breaks[4]

You can find some examples for incorrect extracted texts in section A.7.

The random line breaks result in some missed row names which is reflected by the bigger spread for NA precision with *pdfium* on the synthetic dataset (see Figure 5.28 B). Nevertheless, the NA precision for the majority of the cases is perfect. This is different with the real dataset. The NA precision is found to be at only 0.68.

6.2 Limitations

6.2.1 table extraction

- found mistakes in gold standard with the llm results; mistakes found by human double check
- new lines / splitted lines
- test synthetic hypothesis with pymupdf extract
- 2.4 % wrong gold standard creation
- confidence intervals based on company (know which formats are tricky)

6.2.1.1 Regex baseline

- 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?

[1] The row *Geleistete Anzahlungen* can be found in two parts of the table and the simple approach just matches the numbers to the first found entry.

[2] In this case the regex takes the sum as the value for the previous year.

[3] This can form unexpected numeric patterns or prevent the row names to be recognized.

[4] The approach takes care of line breaks between words, but not within. This leads to unrecognized row names as well.

6.2.2 classification

- Qwen 2.5 hat zweiseitige GuV von IBB entdeckt und zur Anpassung der Ground Truth
- predictor: n_big_tables (tf or llm relevant?)
- Why it is important to have a good recall (or top n accuracy)
- bad performance for Maverick with more models relied to FP8 model version? No. Same reults with FP16

One could build an application that is not asking for a human intervention for reported confidences over 0.9 and then give the possibility to change the page to extract information from later on.

For humans: Easily identifiable if page has a big table with numbers but not so easy to spot the Aktiva / Passiva label.

6.3 Not covered

- OCR
- fine-tuning
- using something smaller (e.g. LSTMs) instead LLMs
- building application, UX design (ref Ambacher 2024)
- table extraction (either VLLMs or classic approaches <– tried tabula but was not successful (because of missing visual traits)?) to prevent wrong text flow and have clear cell borders

in company document next / previous year more helpful than years further away?

6.4 Outlook

- ensemble from multiple models or are errors systematic? (e.g. Wohnungsbaugenossenschaften splitting some rows in multiple and none is picked?)
- check for halluzination vs wrong placed / repeated numbers
- no perfect score even with synthetic data
- flexible extraction (name something, find it, get it)
-

6.4.1 Table extraction

building a document extraction database document by document can improve performance taking advantage of same-company rag in-context learning

predictions for barrierefreie documents of WBM empty, one time because the pages showed **GuV**; also no predictions for Zoo 2024 **Passiva**

Chapter 7

Conclusion

References

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

BMI, Referat O2 (Ed.). (2013). *Minikommentar zum Gesetz zur Förderung der elektroni- schen Verwaltung sowie zur Änderung weiterer Vor- schriften.*

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

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

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.*

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

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

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

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

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

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

List of Figures

List of Tables

Glossary

ACFR Annual Comprehensive Financial Report
AUC area under the curve
BHT Berliner Hochschule für Technik
GPU graphics processing unit
GuV Gewinn- und Verlustrechnung
HGB Handelsgesetzbuch
HITL human-in-the-loop
LLM large language model
OCR optical character recognition
PDF Portable Document Format
RHvB Rechnungshof von Berlin
RechKredV Verordnung über die Rechnungslegung der Kreditinstitute, Finanzdienstleistungsinstitute und Wertpapierinstitute
SHAP SHapley Additive exPlanations
TF-IDF Frequency-Inverse Document Frequency
TOC table of contents
ebnf extended Backus–Naur form
glm generalized linear model
json JavaScript Object Notation
mcc multi-class classification
regex regular expression
vLLM Virtual Large Language Model

Chapter A

Appendix

A.1 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

A.2 Benchmarks

A.2.1 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

Table A.1: Comparing extraction time (in seconds) for different Python package

package	runtime in s
pdfium	{14}
pymupdf	22
pypdf	218
pdfplumber	675
pdfminer	752
doclingparse	1621

in this thesis, because Auer et al. (2024) published no quantitative results. The benchmark runs on the local machine described in section A.1. There are 5256 pages to extract the text from.

Table A.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.

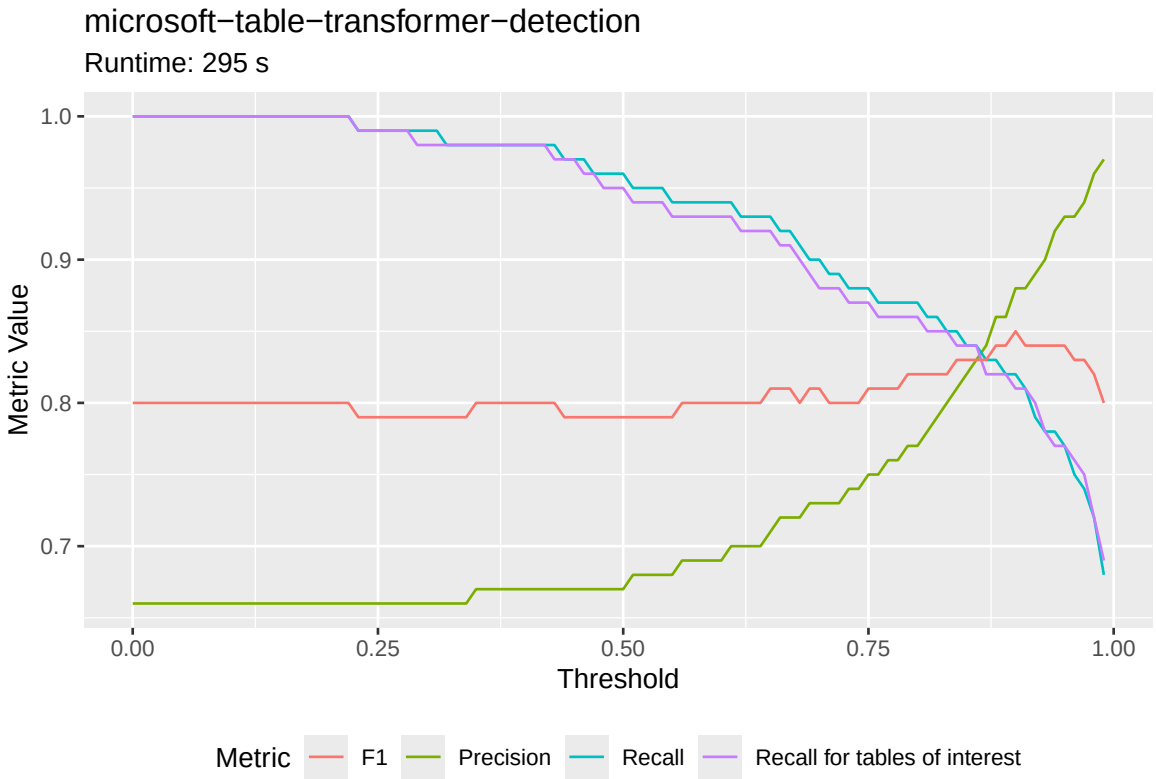
Auer et al. (2024) reports, that *pdfium* occasionally merges text cells that are not close o each other, resulting in unrecoverable quality issues. Thus we checked some of the extracted texts manually and include the PDF extraction backend as a variable in our experiments. The page identification experiment, 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 A.7.

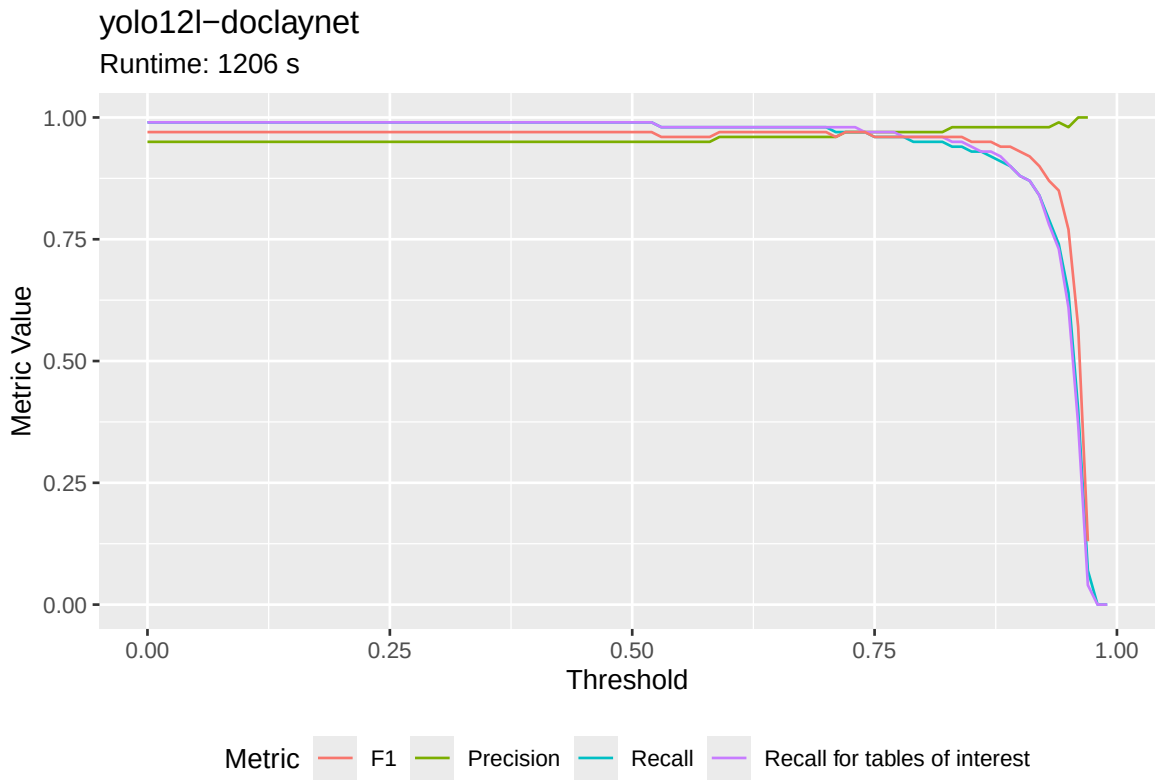
A.2.2 Table detection

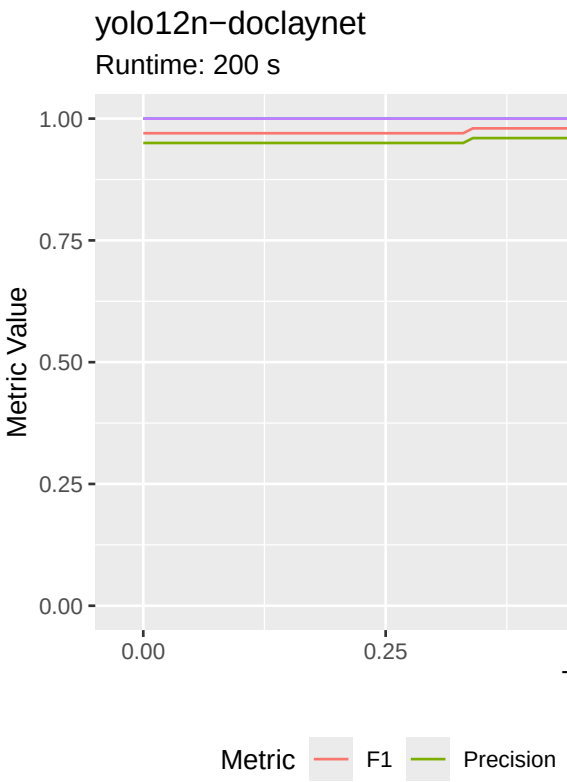
- yolo benchmark and table transformer
- skip classification with llm

not so important anymore

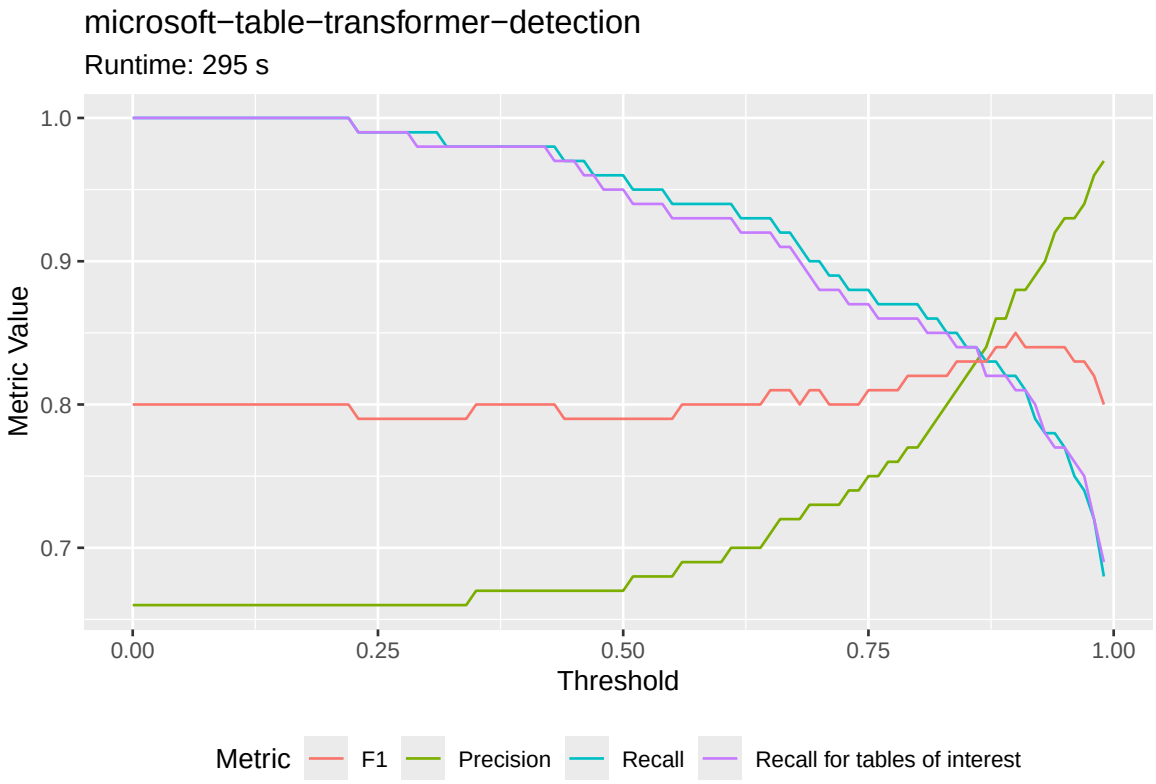


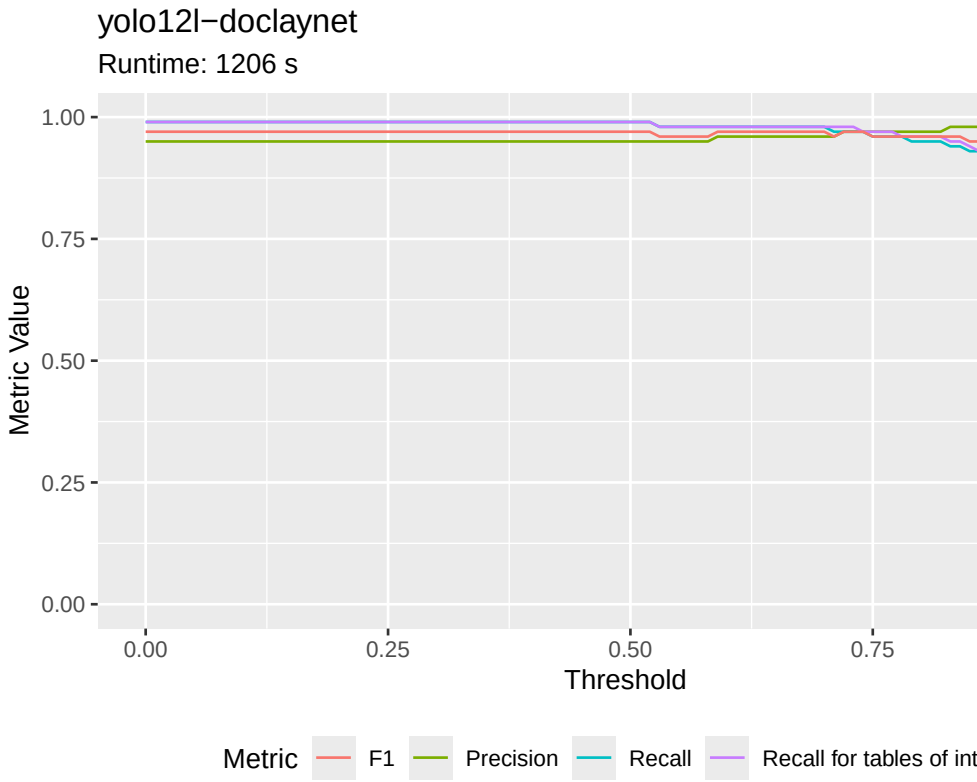
You see the plot for: microsoft-table-transformer-detection. (Click to stop automatic rotation.)





You see the plot for: yolo12l-doclaynet. (Click to stop automatic rotation.)
You see the plot for: yolo12n-doclaynet. (Click to stop automatic rotation.)



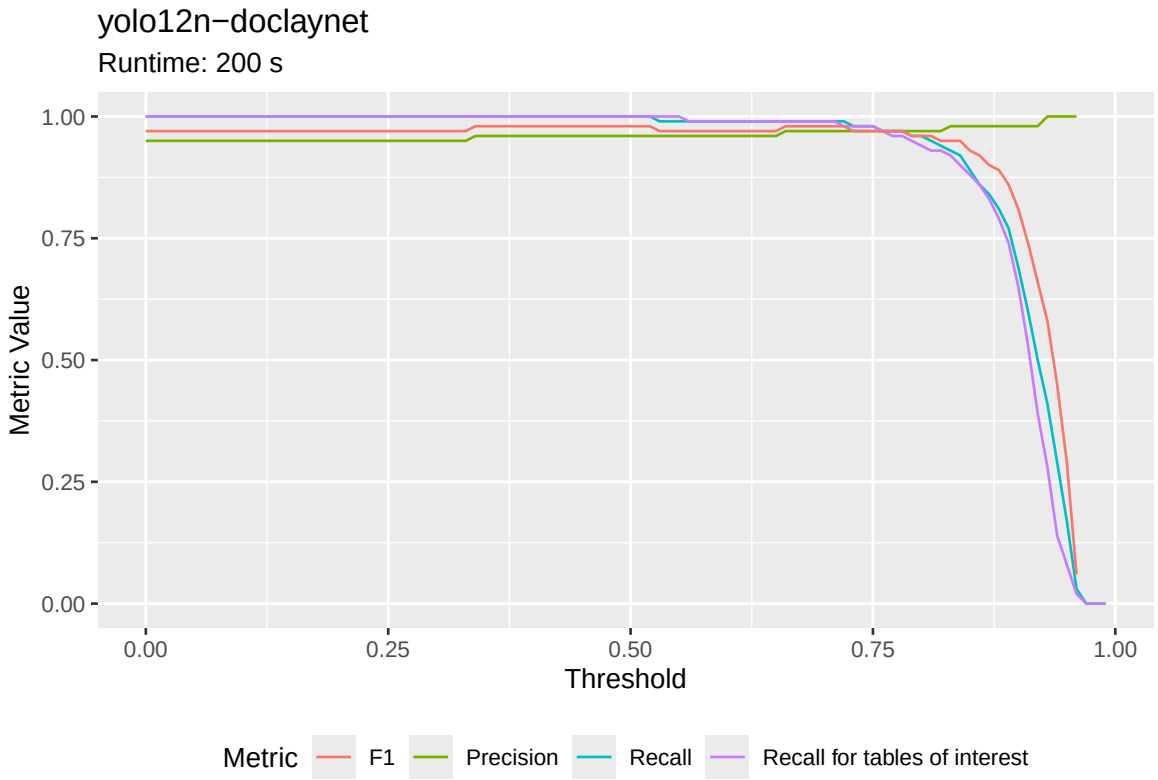


microsoft-table-transformer-detection

A

Table A.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



yolo12l-doclaynet
yolo12n-doclaynet

A

A.2.3 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 A.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 5.1.3). Previous experiments were only conducted on a subset of pages, that were selected based on the results of the *simple regex* approach (see section 5.1.1).

A.3 Prompts

A.3.1 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."},
    ]
```

A.3.2 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
```

```
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'''
"""})
```

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
```

A.4 Regular expressions

Here one can find the three regular expressions used for the benchmarks presented in section 5.1.1.

```
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|Jahre
```

```
        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"
    ]
}
```

A.5 Annual Comprehensive Financial Report Balance Sheet

A.6 Extraction framework flow chart

A.7 Table extraction with regular expressions

Extract by pdfium for '../../benchmark_truth/synthetic_tables/separate_files/final/aktiva_table__3_columns__span_False__thin
€__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
```

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 A.1: Example balance sheet pagefom Californias Annual Comprehensive Financial Report 2023

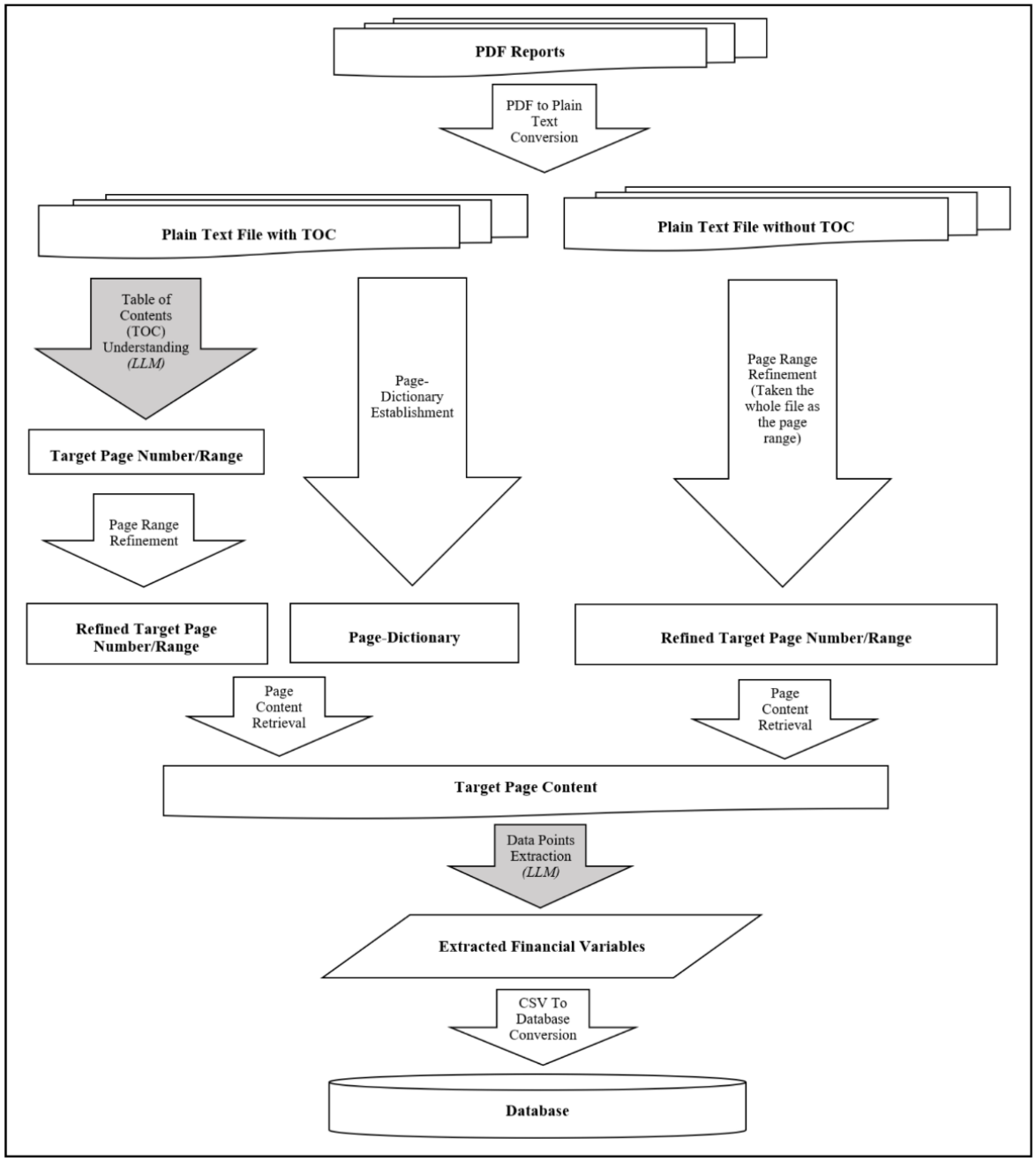


Figure A.2: Flowchart of the extraction framework of Auer et al. (2024)

```
LizenzenansolchenRechtenundWerten

4,646,71

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

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__th
€__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

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

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'
```

Chapter B

Tables

B.0.1 Classification

B.0.2 Table extraction

B.0.2.1 Hybrid approach

Table B.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 B.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_rate_s
Qwen3235BA22BInstruct2507FP8	{0.983}	{0.966}	{0.897}	0.5	
Llama4Scout17B16EInstruct	0.931	0.897	0.448	0.33	
MistralLargeInstruct2411	0.966	0.897	0.776	0.67	
Llama3.18BInstruct	0.828	0.759	0.552	0.286	
Qwen38B	0.931	0.759	0.336	{0.714}	
Ministral8BInstruct2410	0.862	0.741	0.552	0.467	
gemma327bit	0.862	0.672	0.207	0.579	
gemma312bit	0.793	0.5	0.543	0.586	

Chapter C

Figures

NA predicting

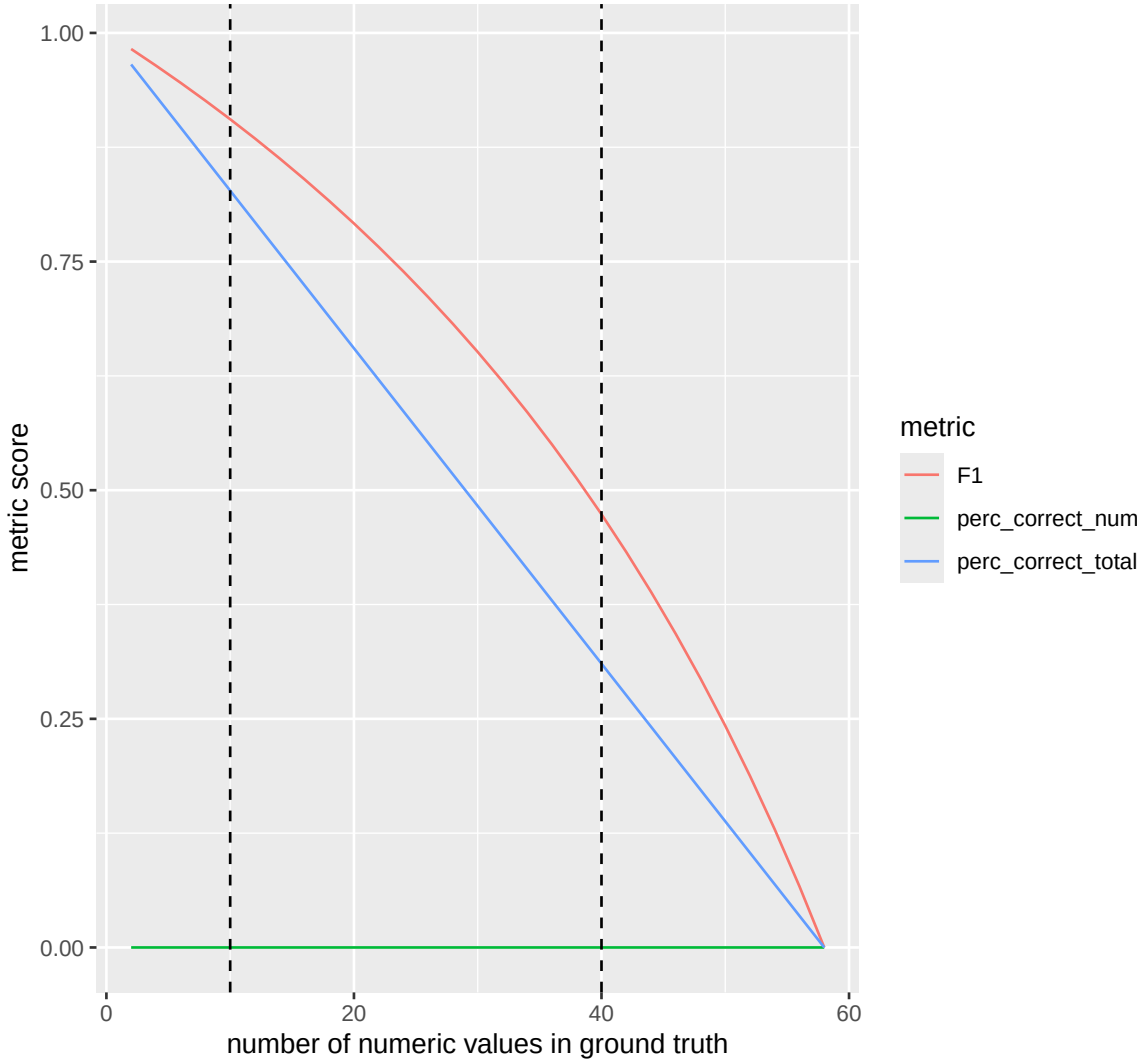


Figure C.1: Displaying the performance metrics a LLMs response would have, if all predictions are 'null'. The area between the two dashed lines shows the number of numeric values found in the real Aktiva tables.

C.1 Page identification

C.1.1 Regex baseline

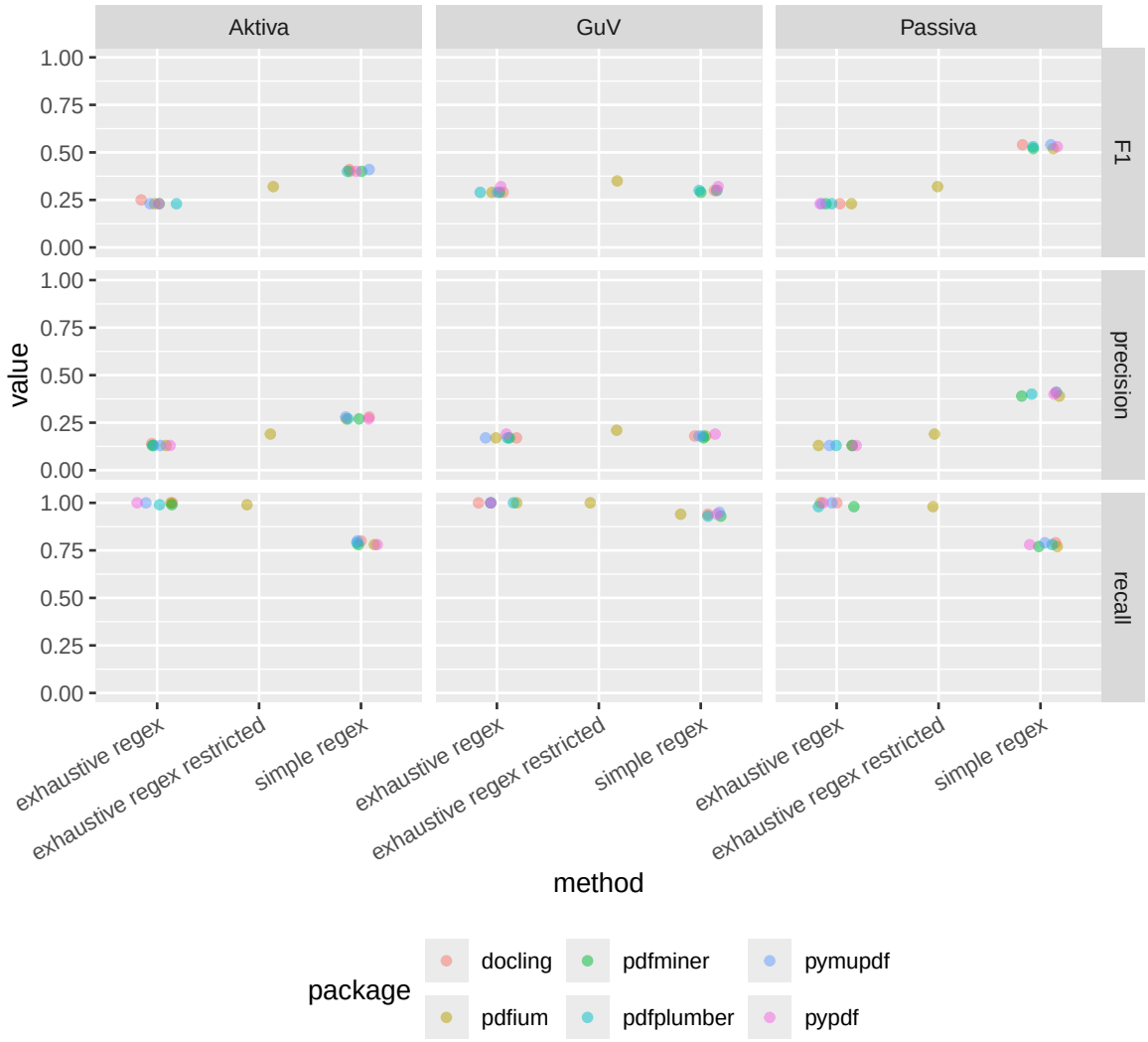


Figure C.2: Comparing page identification metrics for different regular expressions for each classification task by type of the target table.

C.1.2 TOC understanding

C.1.3 Classification

C.1.3.1 Binary

Binary classification F1 score over runtime limited to 60 minutes

Binary classification F1 score over runtime unlimited

C.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

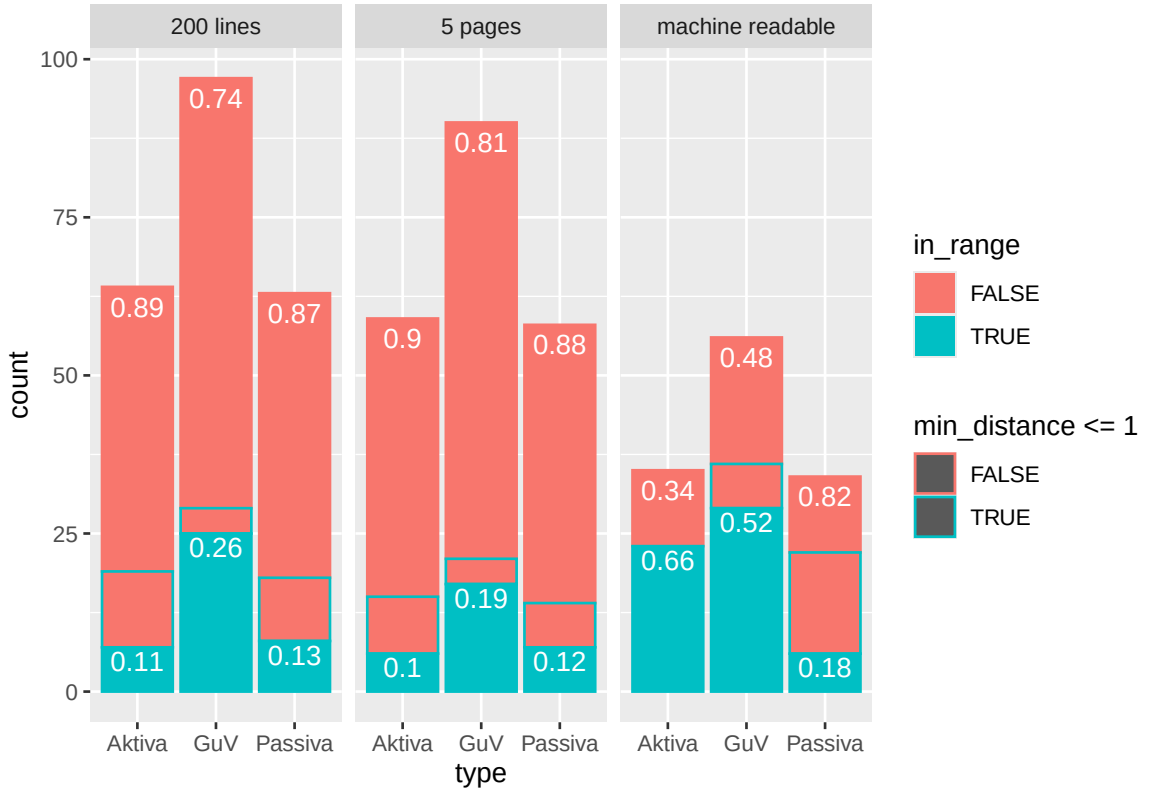


Figure C.3: Comparing number of fount TOC and amount of correct and incorrect predicted page ranges

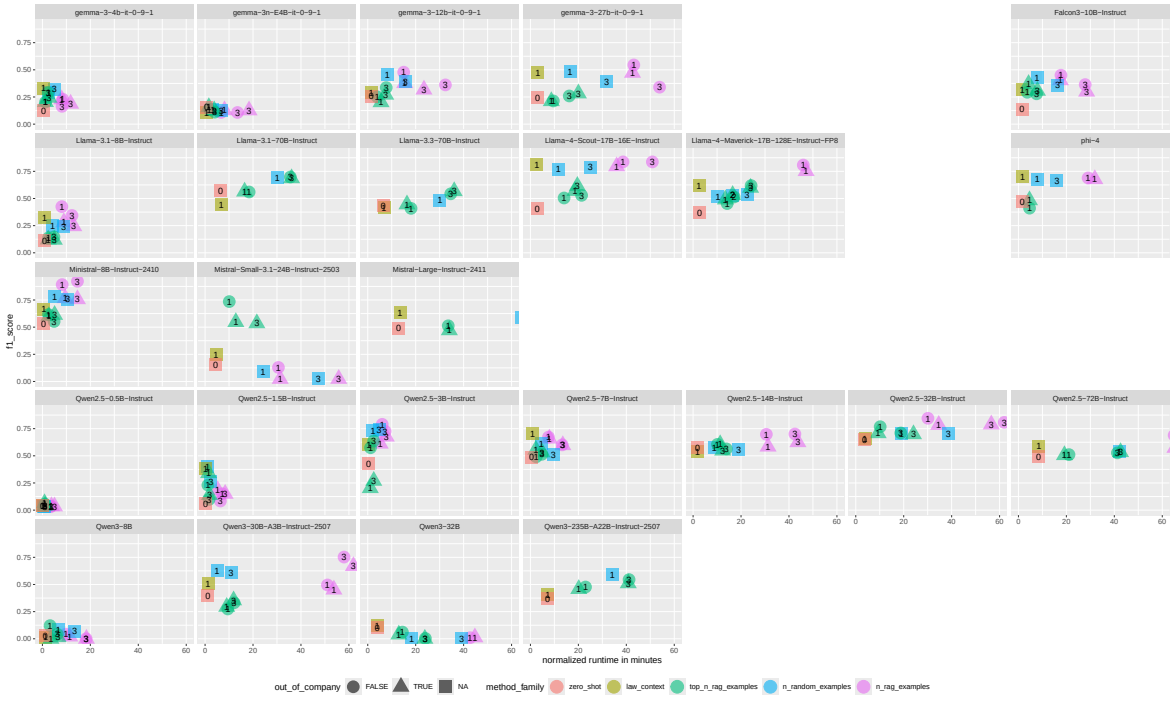


Figure C.4: Comparing F1 score over normalized runtime for binary classification task. The normalized runtime is given in minutes of processing on a single B200. The time to load the model into the VRAM is excluded. Focussing on small models showing only 60 minutes of runtime.

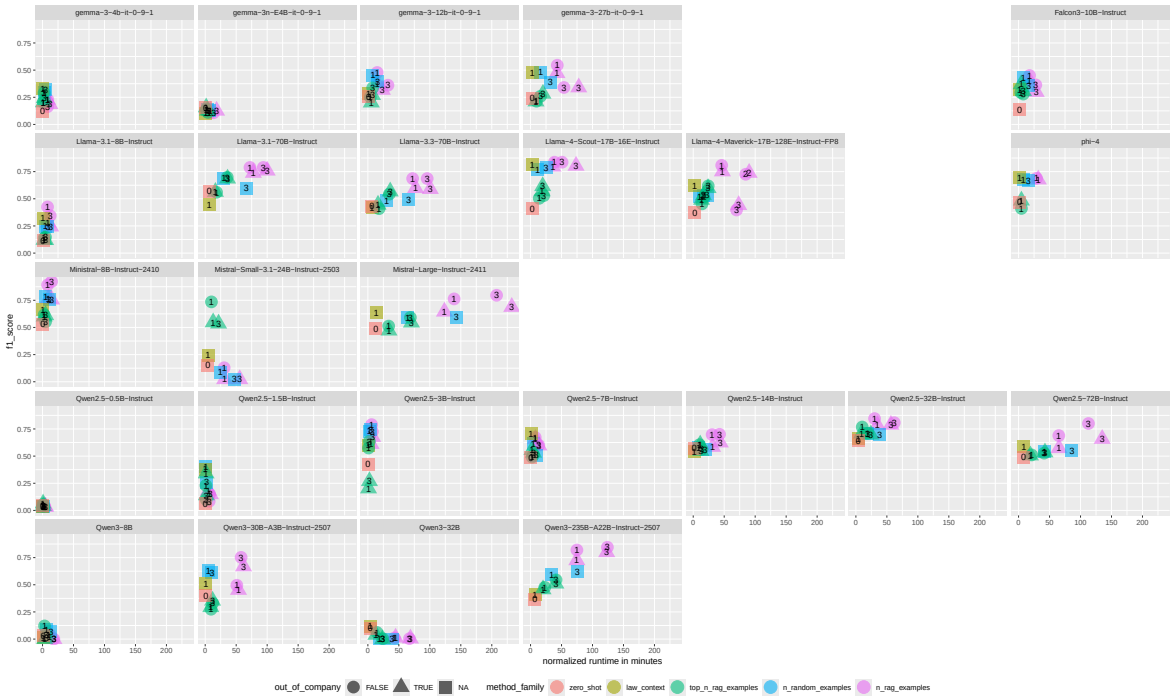


Figure C.5: Comparing F1 score over normalized runtime for binary classification task. The normalized runtime is given in minutes of processing on a single B200. The time to load the model into the VRAM is excluded.

Figure C.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 C.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.

C.2 Table extraction

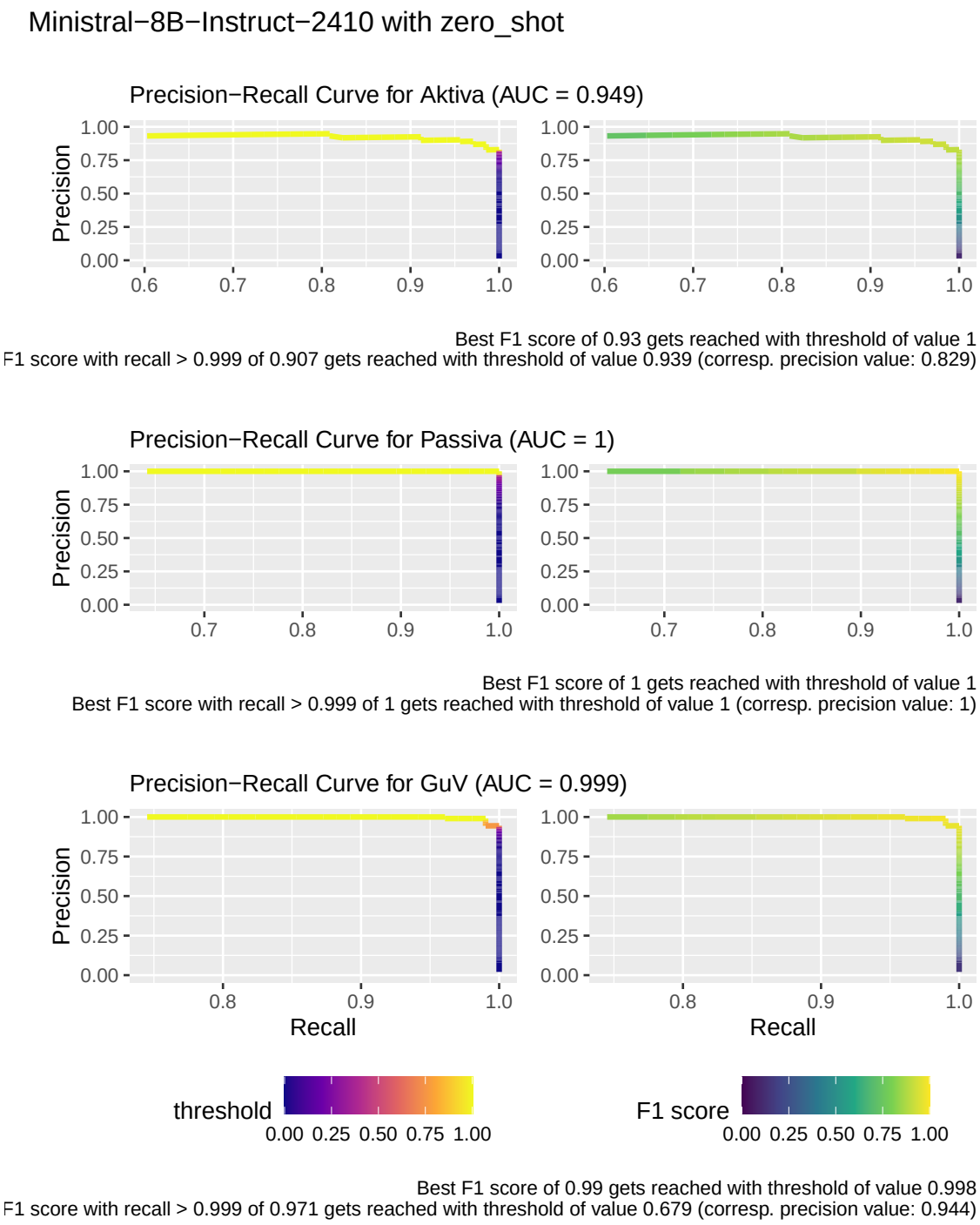


Figure C.8: Showing the precsion-recall-curve for Llama-4-Scout-17B-16E-Instruct.

C.2.1 Regex approach

C.2.1.1 Real tables

C.2.1.2 Synthetic tables

C.2.2 Real tables

C.2.2.1 Examples from same company

C.2.2.2 OpenAI models

C.2.2.3 Hypotheses

C.2.3 Synthetic tables

C.2.3.1 Confidence

C.2.4 Hybrid approach

real_table_extraction_llm_synth_context_shap_plot

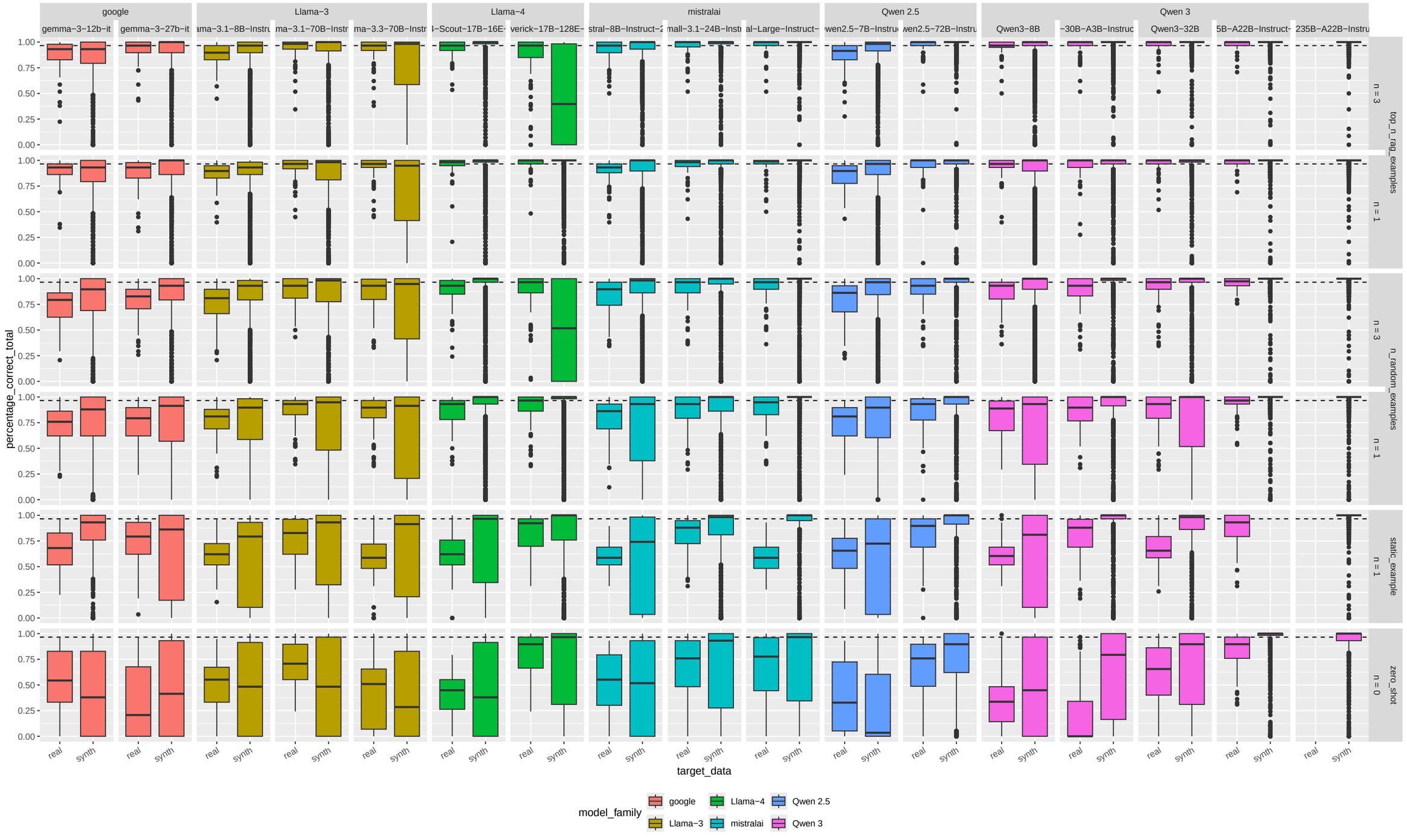


Figure C.9: Comparing the table extraction performance among real and synthetic Aktiva tables

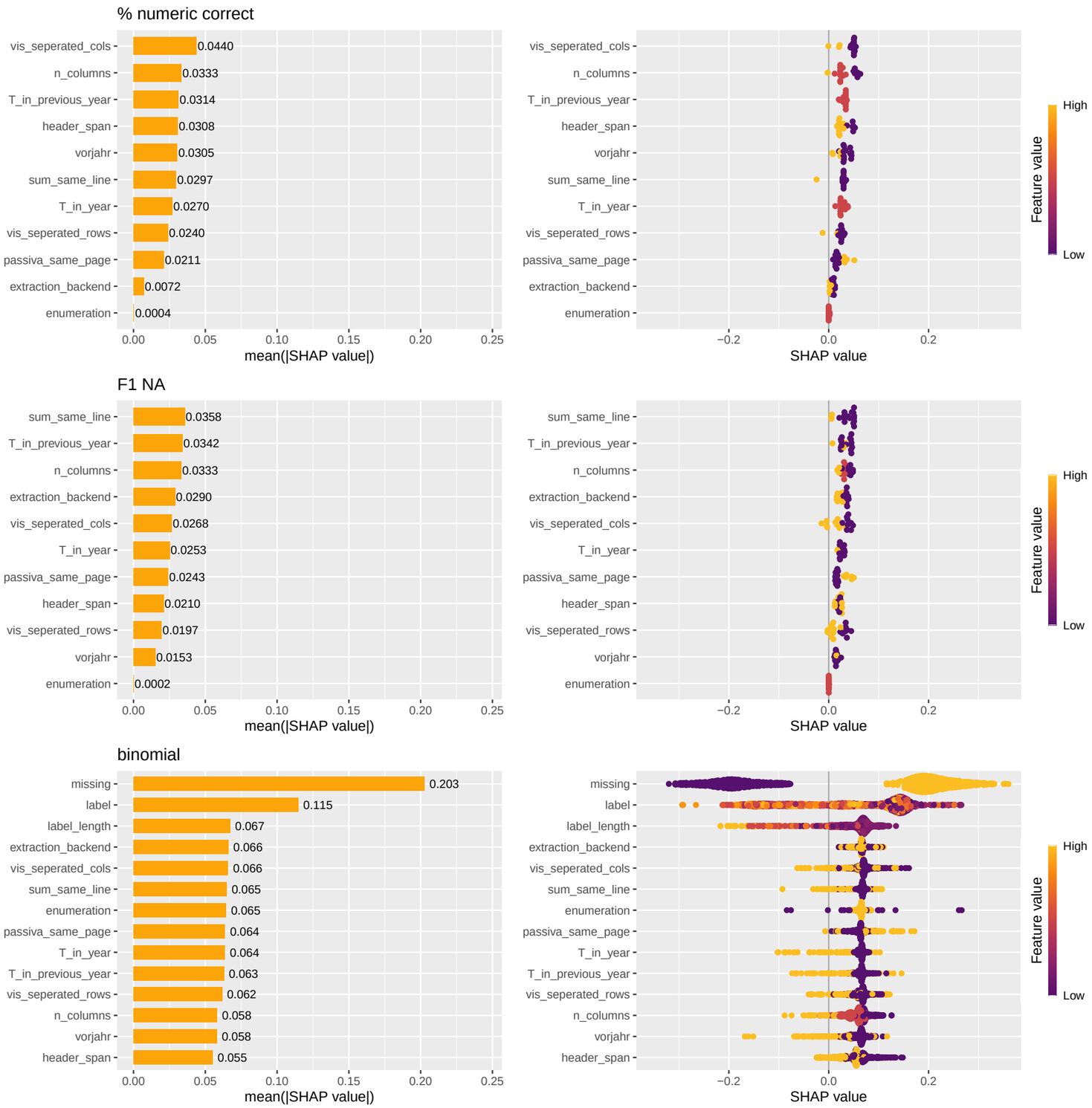


Figure C.10: Mean absolute SHAP values and beeswarm plots for real table extraction with regular expression approach

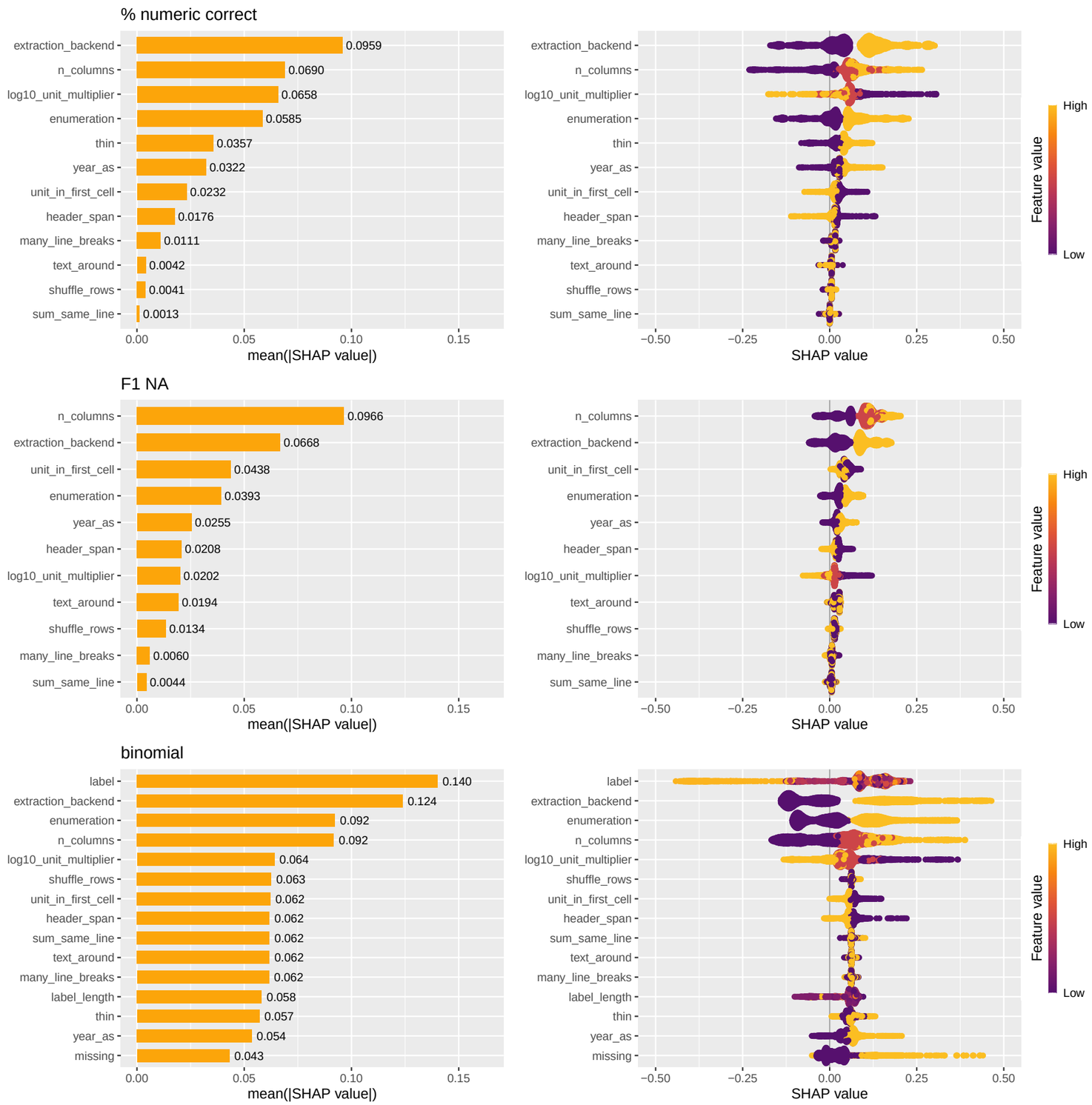


Figure C.11: Mean absolute SHAP values and beeswarm plots for synth table extraction with regular expression approach

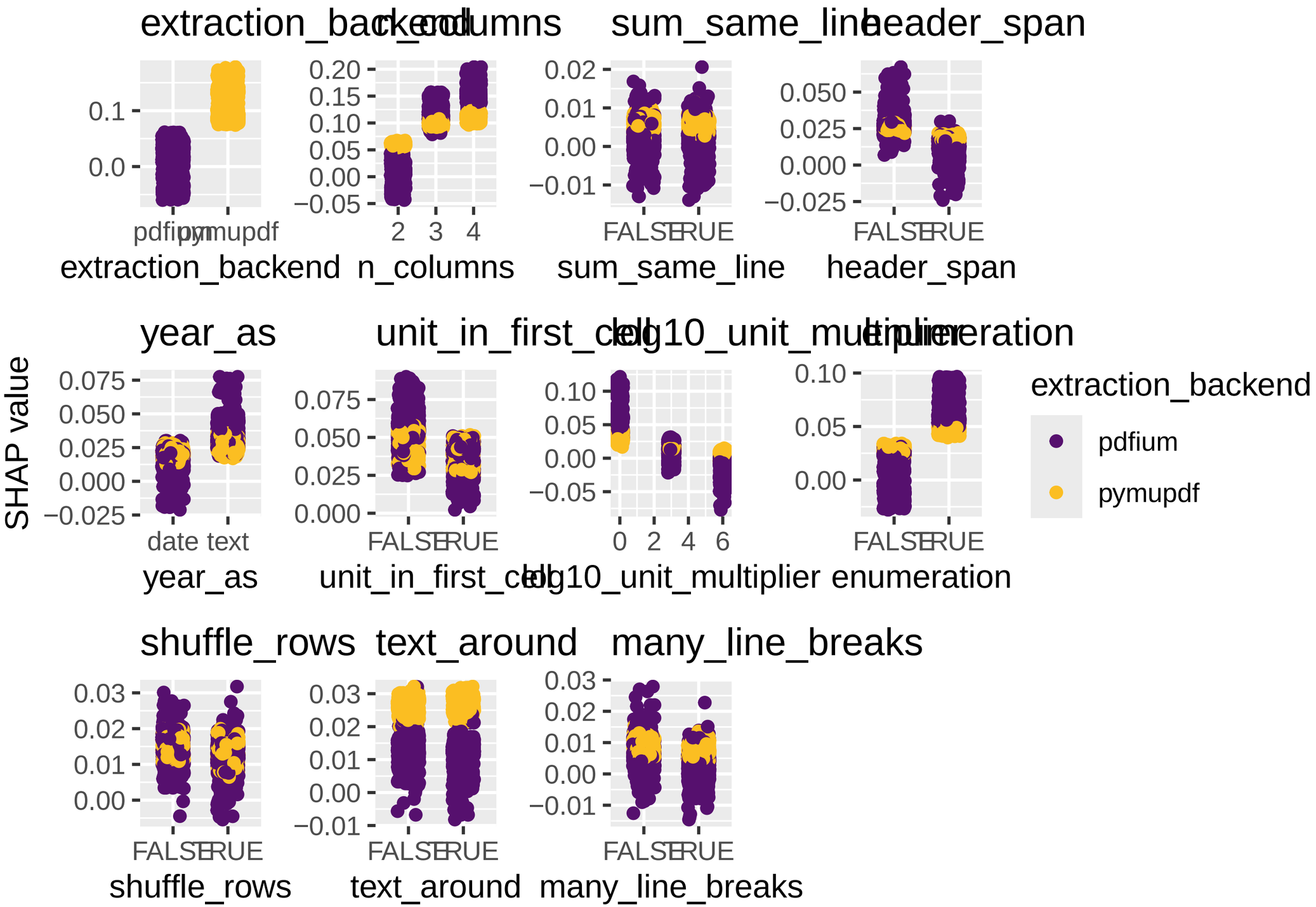


Figure C.12: Showing the interactions of the extraction backend pdfium with the table characteristics for F1 score.

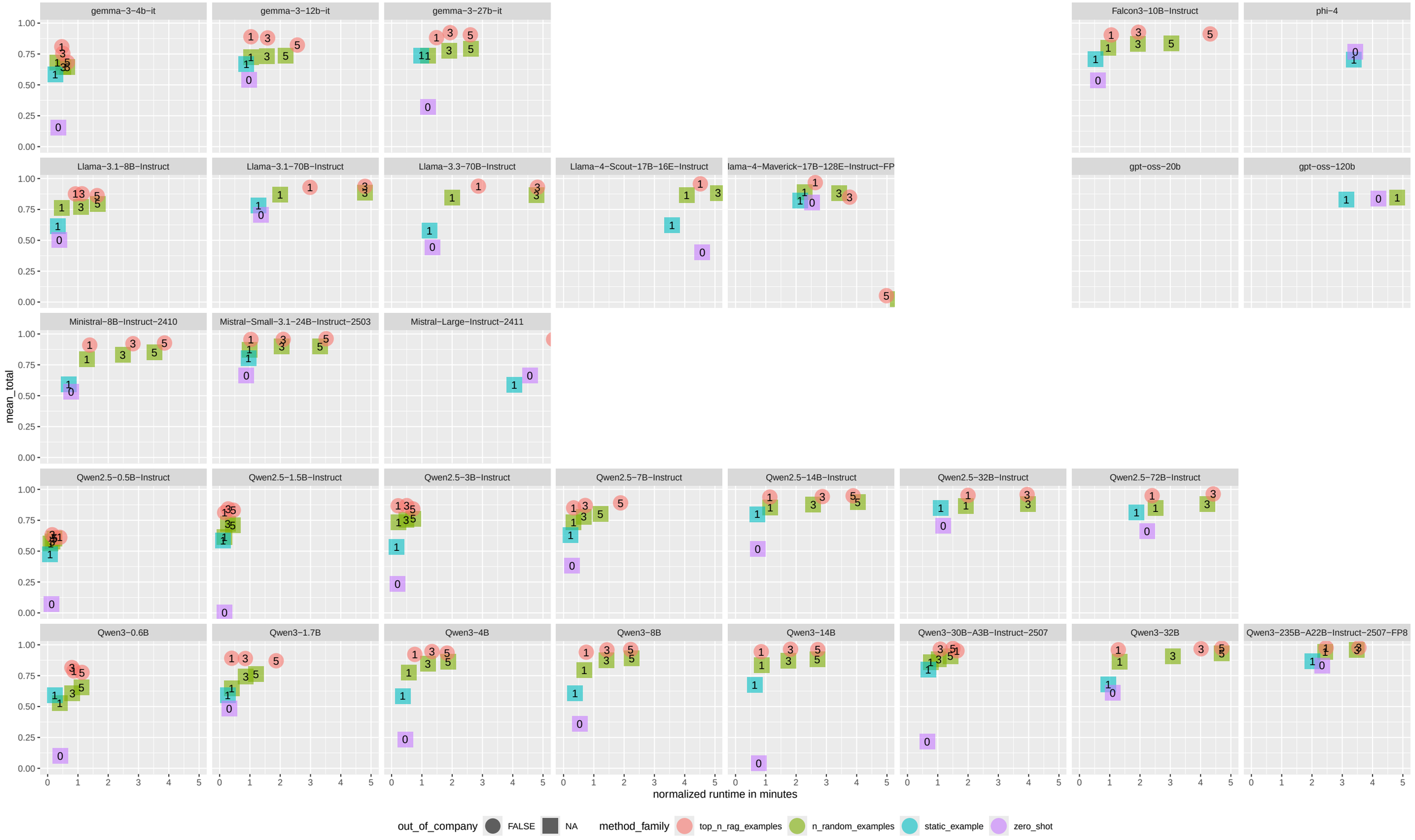


Figure C.13: Comparing percentage of correct predictions total over the normalized runtime. The normalized runtime is given in minutes of processing on a single B200. The time to load the model into the VRAM is excluded. Focussing on small models showing only 5 minutes of runtime.

Figure C.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.

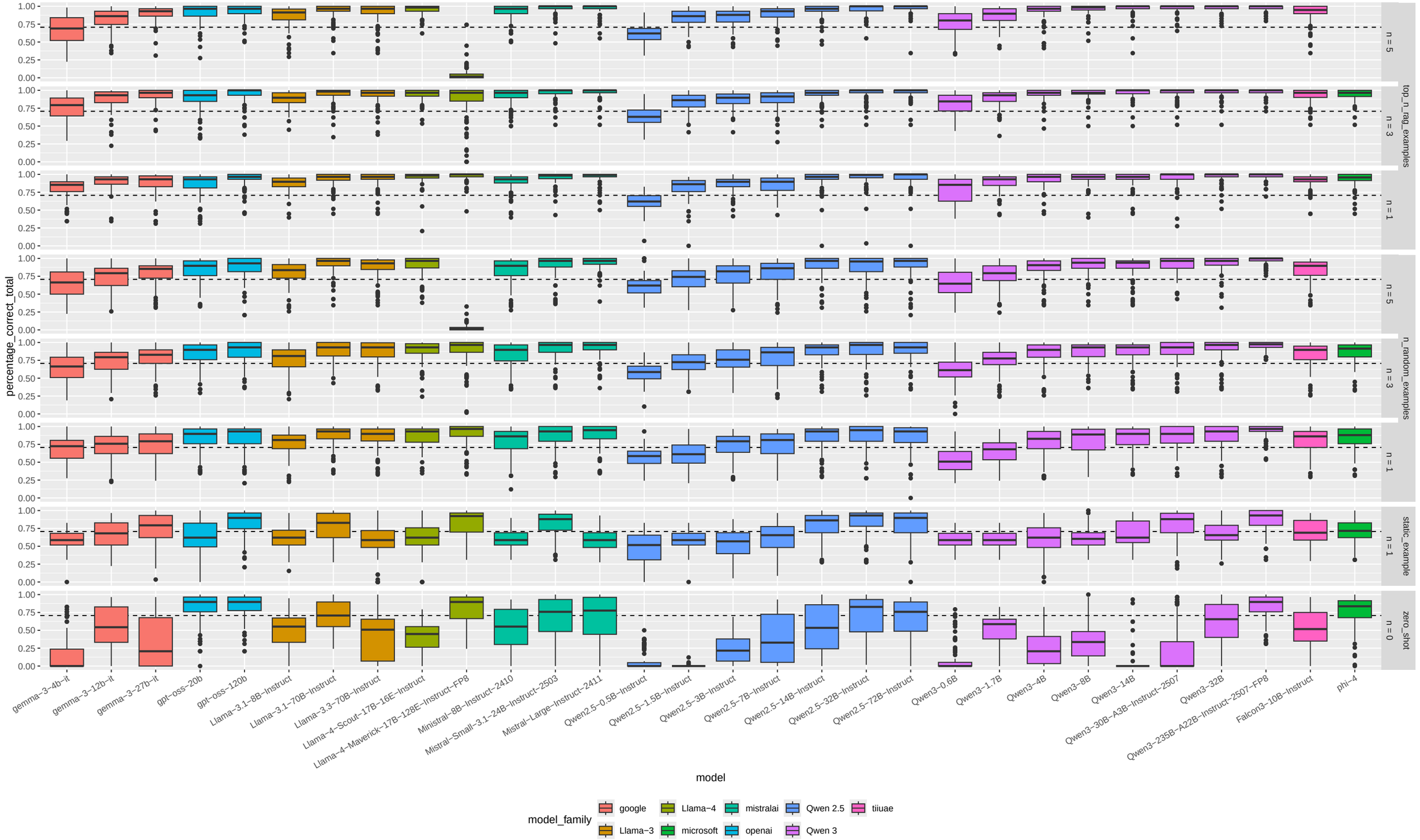


Figure C.15: Percentage of correct extracted or as missing categorized values for table extraction task on real Aktiva tables

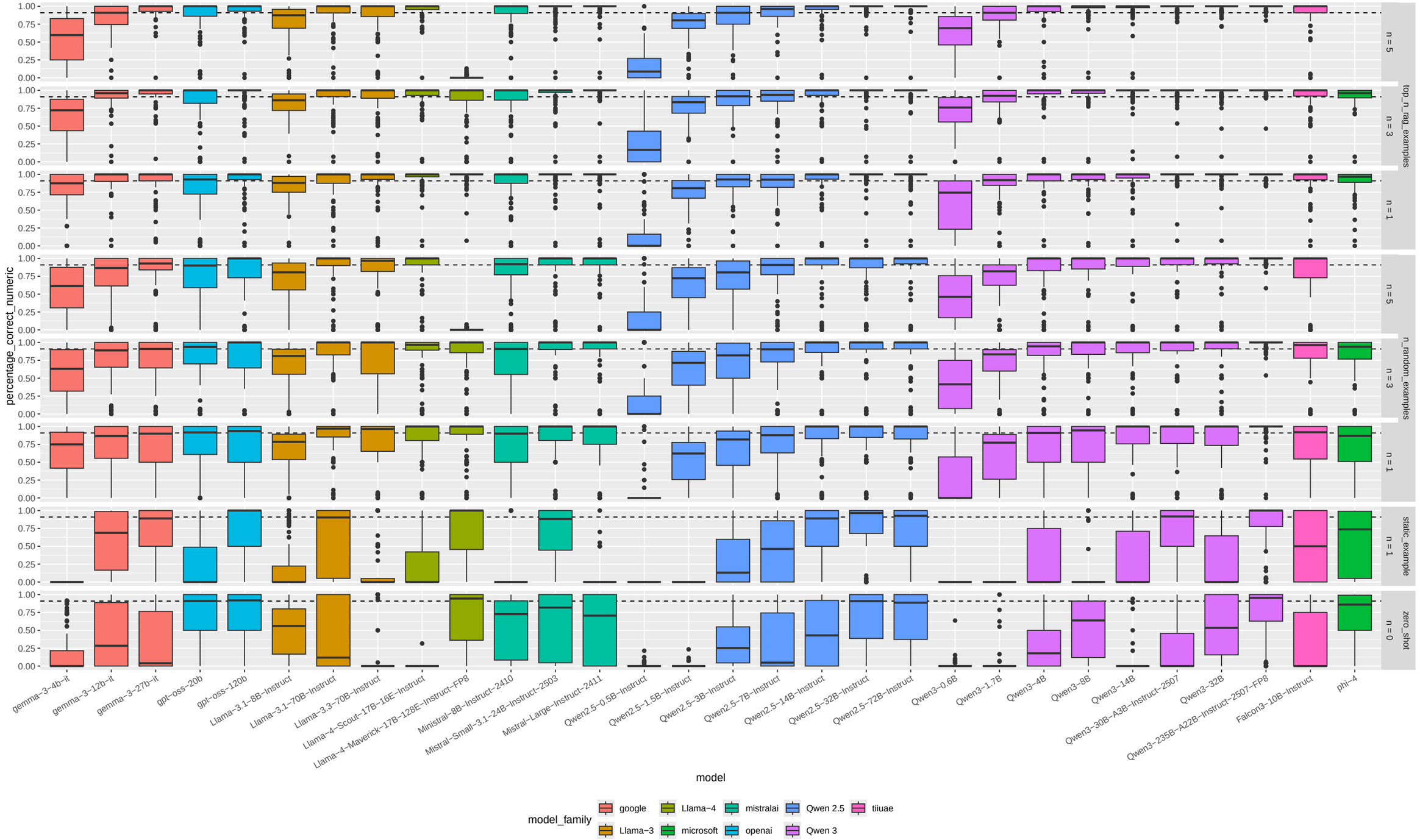


Figure C.16: Percentage of correct extracted numeric values for table extraction task on real Aktiva tables

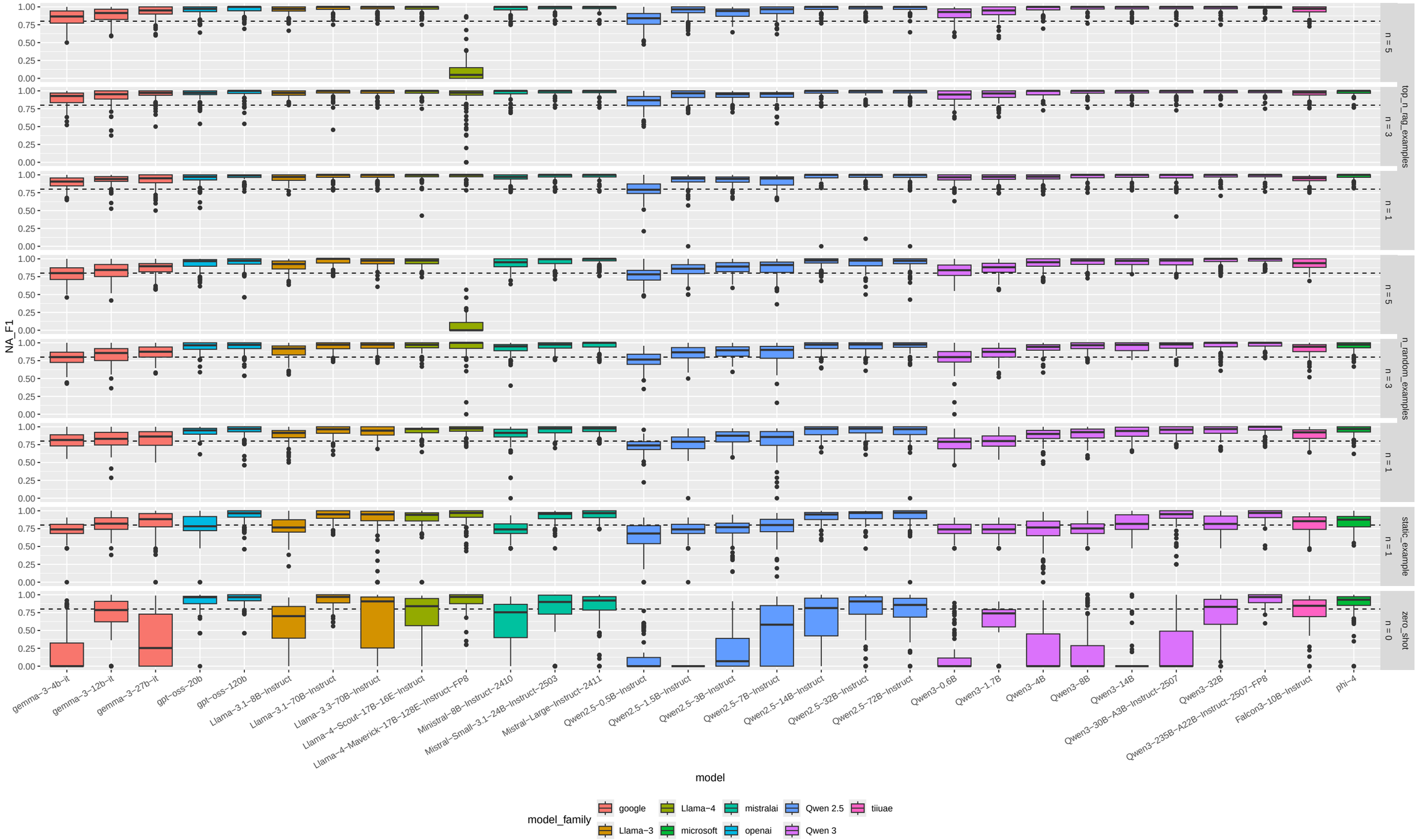


Figure C.17: F1 score for the missing classification if a value is missing for table extraction task on real Aktiva tables

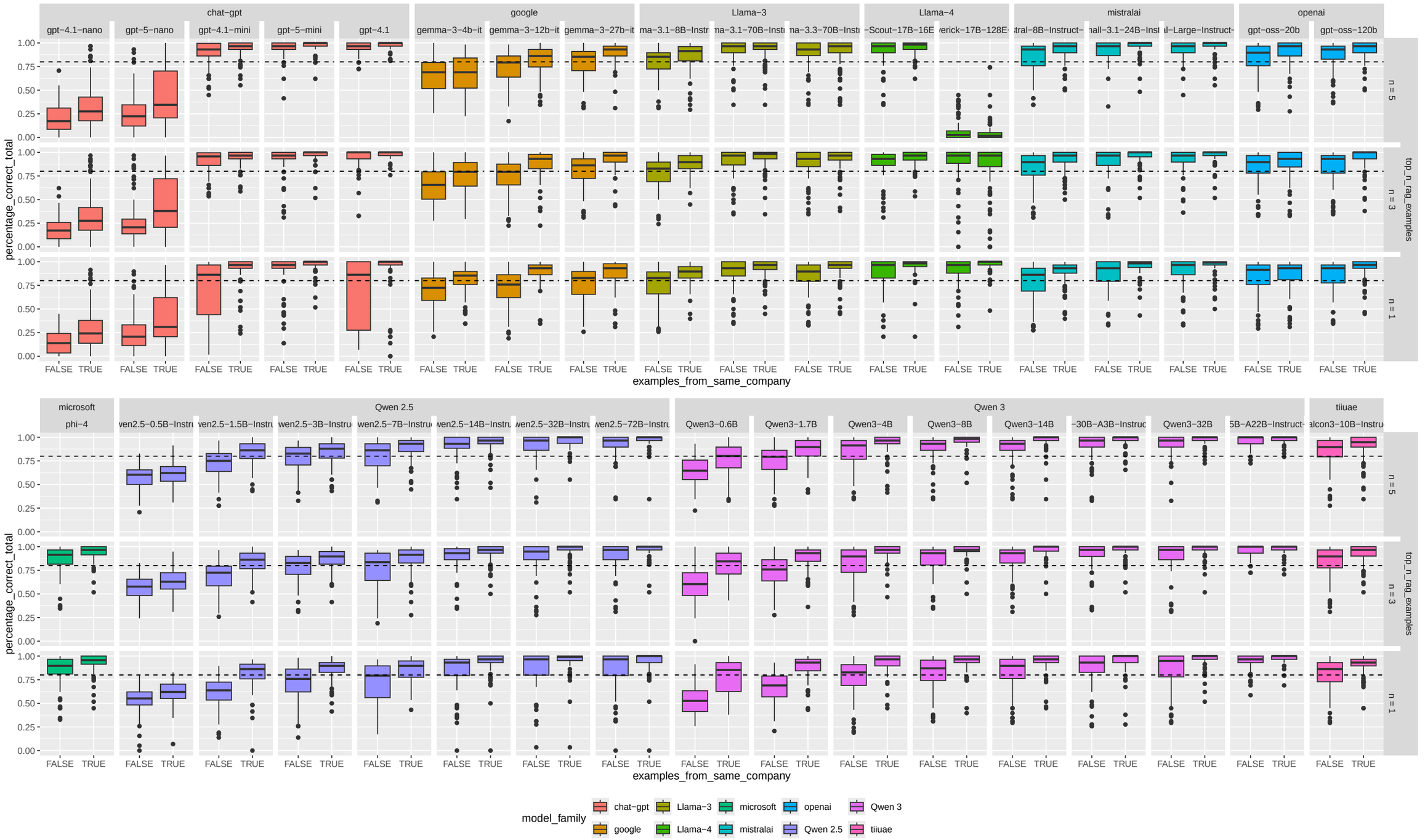


Figure C.18: Comparing the overall extraction performance depending on the condition if examples from the same company can be used.

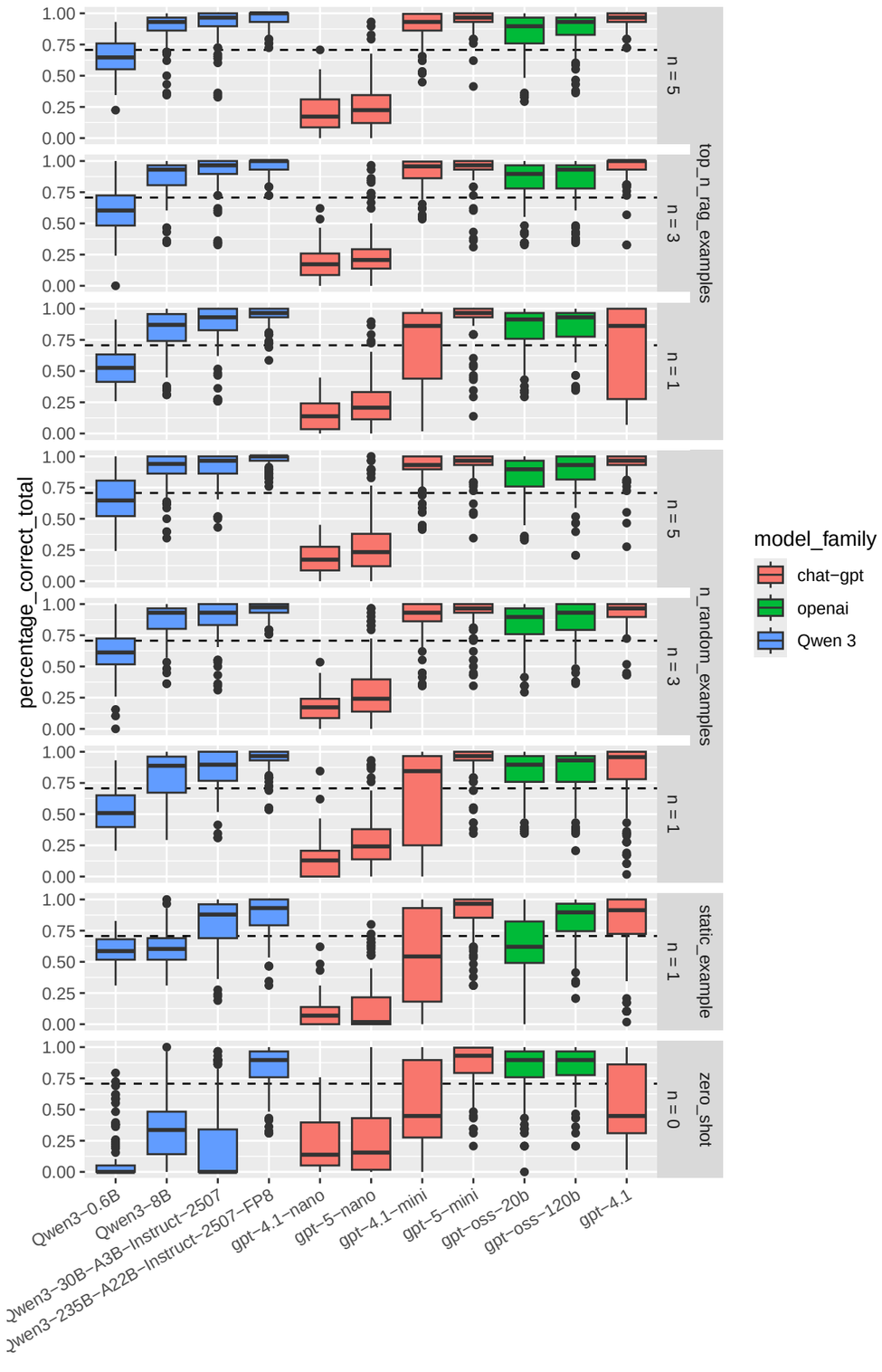


Figure C.19: Comparing the percentage of correct predictions overall for OpenAi's LLMs with some Qwen 3 models

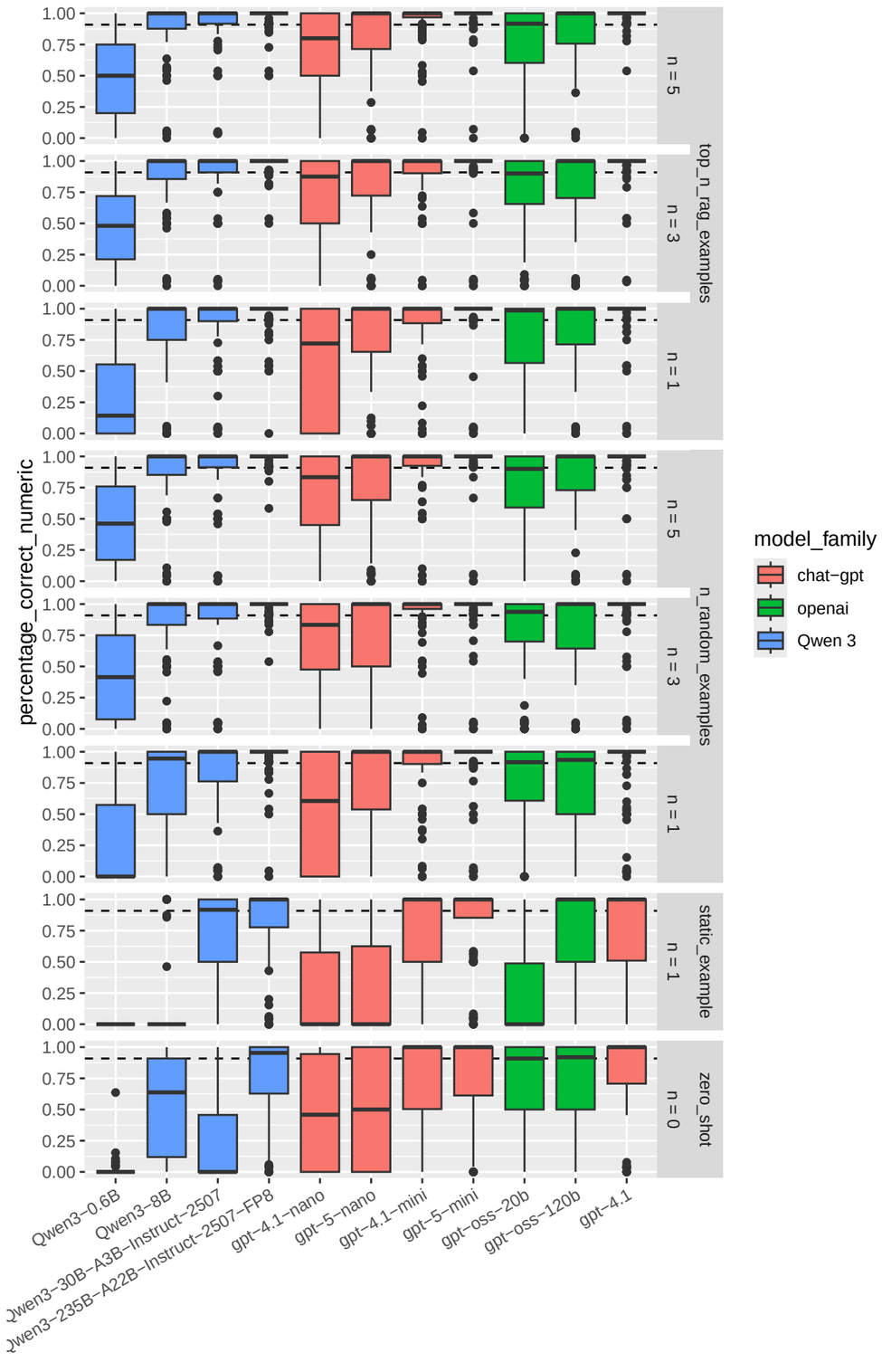


Figure C.20: Comparing the percentage of correct numeric predictions for OpenAi's LLMs with some Qwen 3 models

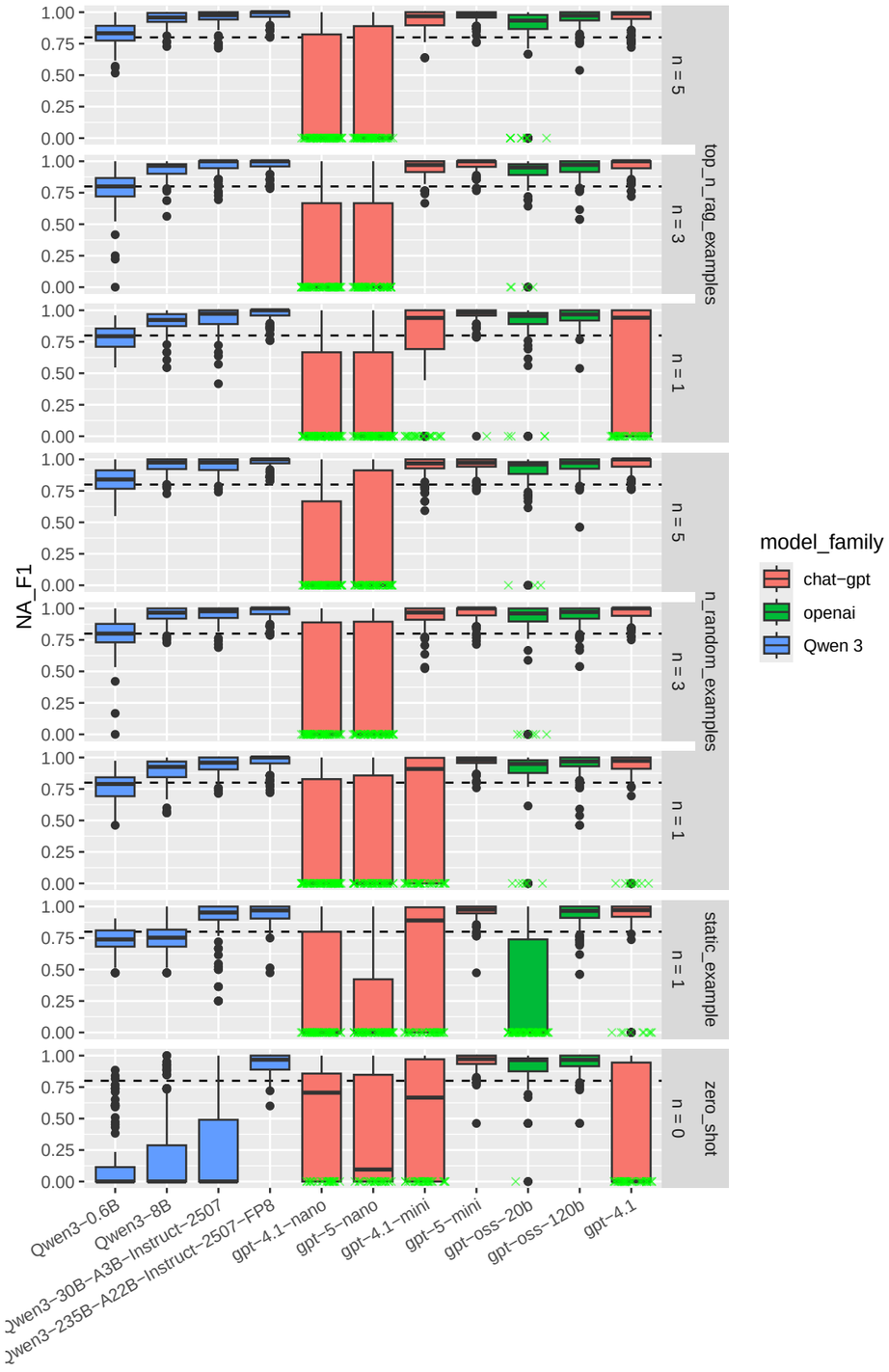


Figure C.21: Comparing the F1 score for predicting the missingness of a value for OpenAi's LLMs with some Qwen 3 models. The green crosses indicate results where a model has predicted only numeric values even though there have been missing values.

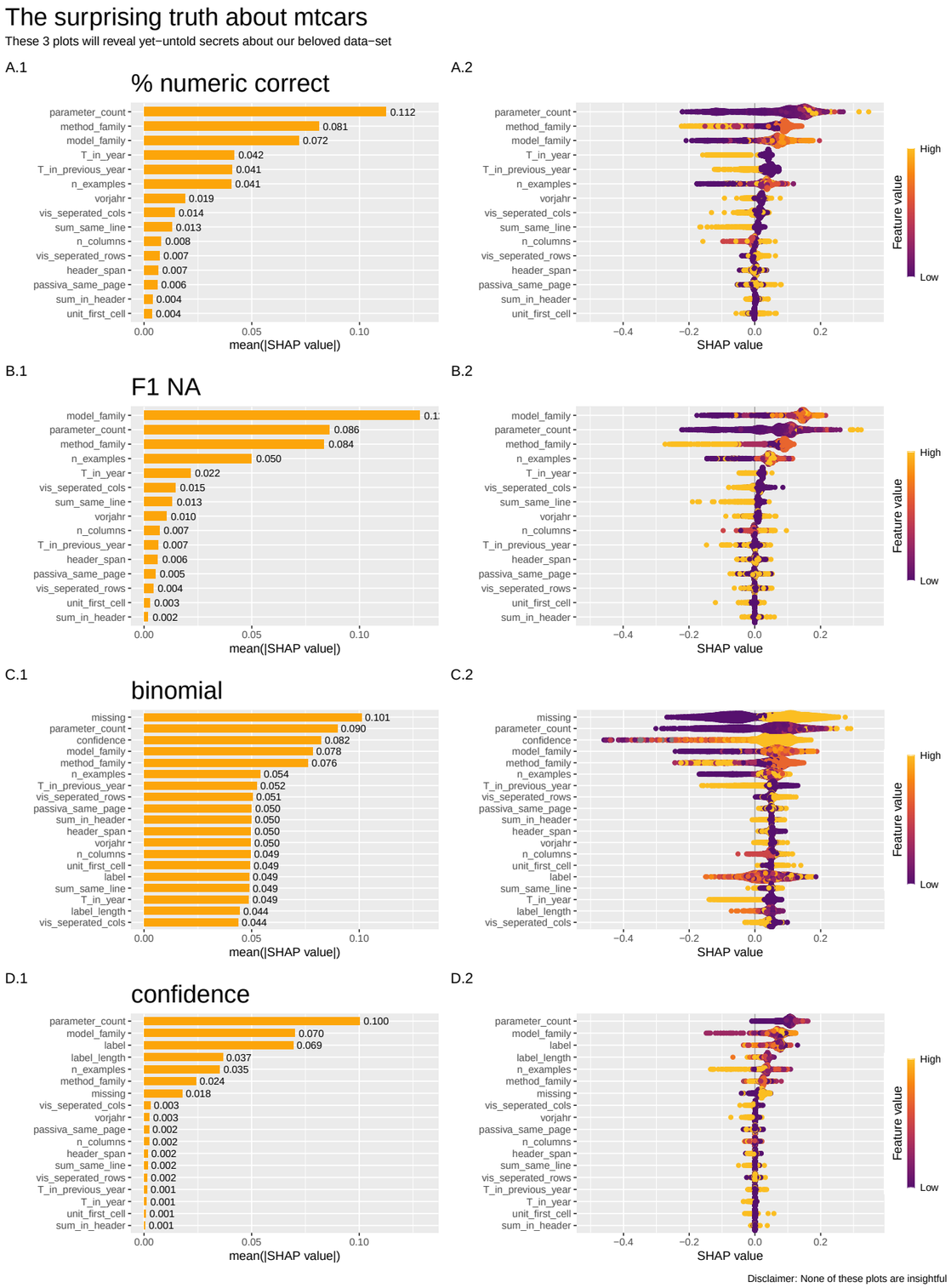


Figure C.22: Mean absolute SHAP values and beeswarm plots for real table extraction with LLMs

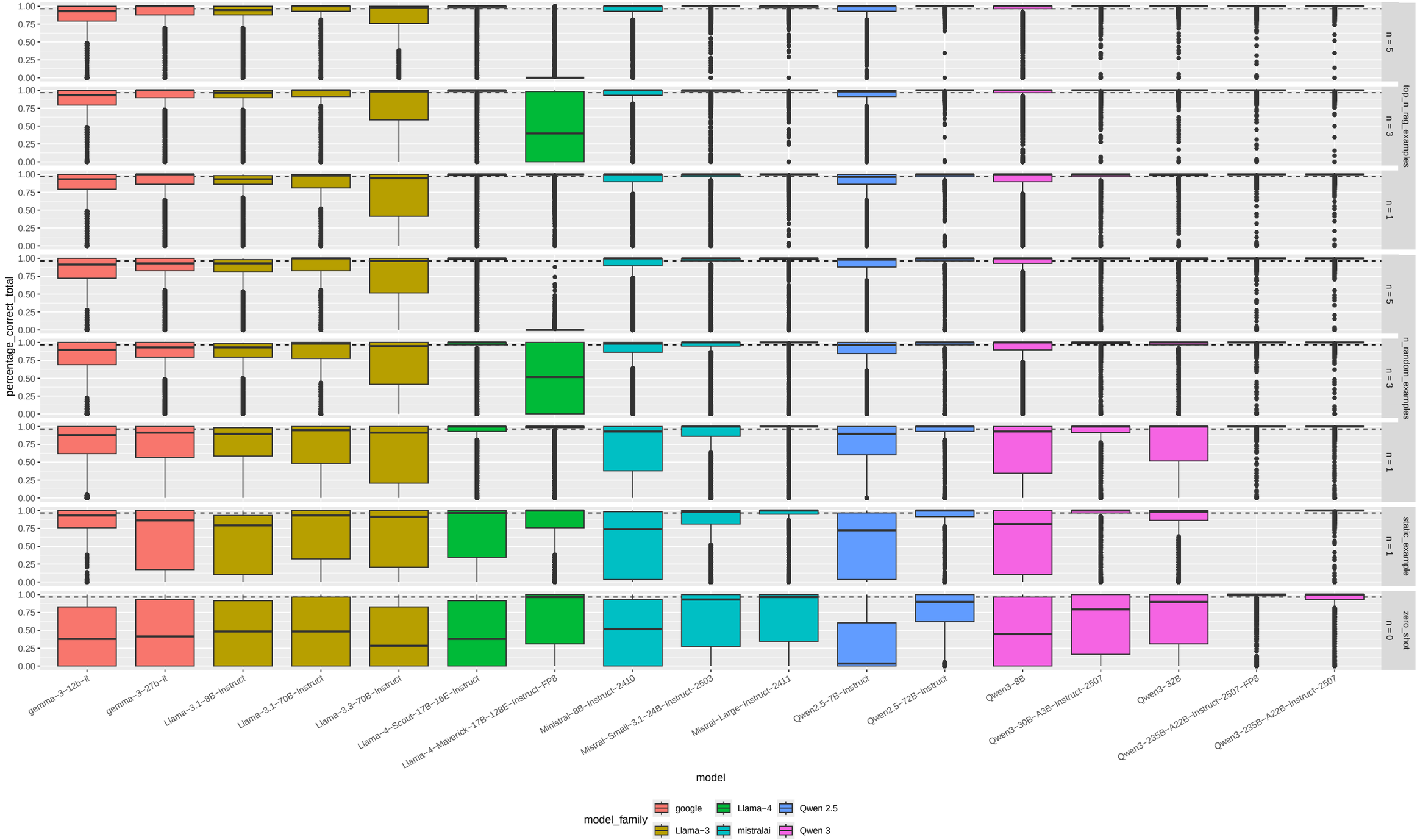


Figure C.23: Percentage of correct extracted or as missing categorized values for table extraction task on synthetic Aktiva tables

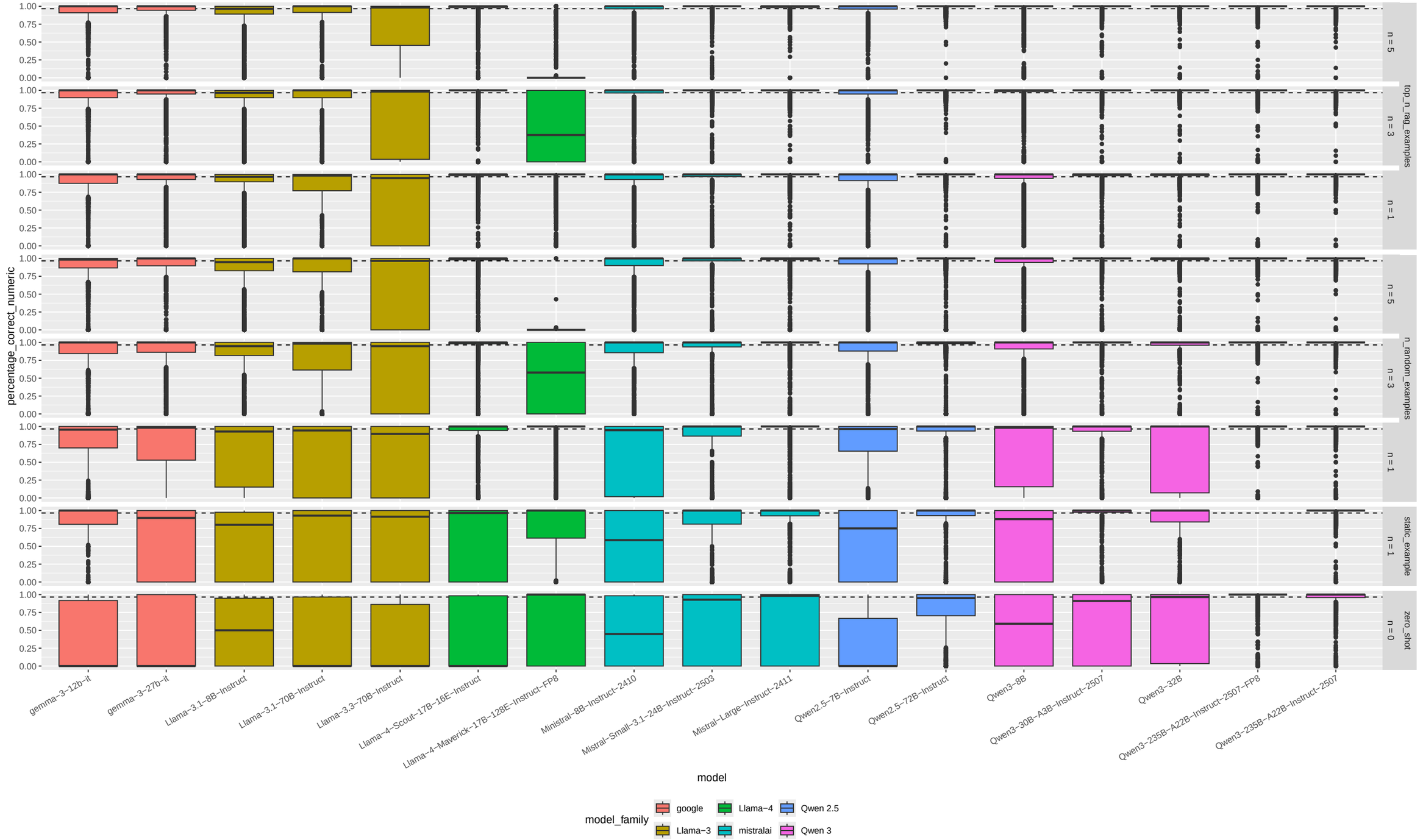


Figure C.24: Percentage of correct extracted numeric values for table extraction task on synthetic Aktiva tables

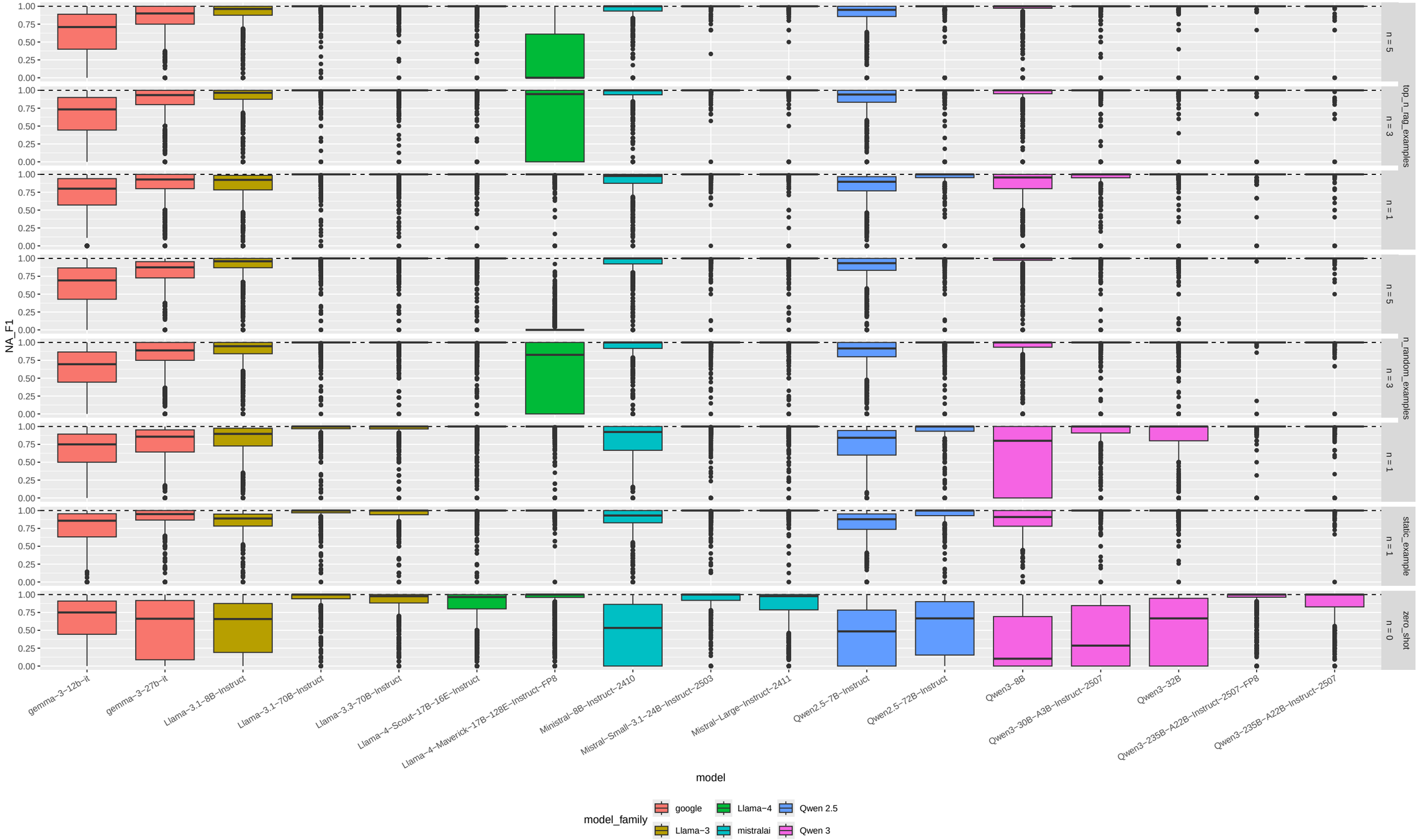


Figure C.25: F1 score for the missing classification if a value is missing for table extraction task on synthetic Aktiva tables

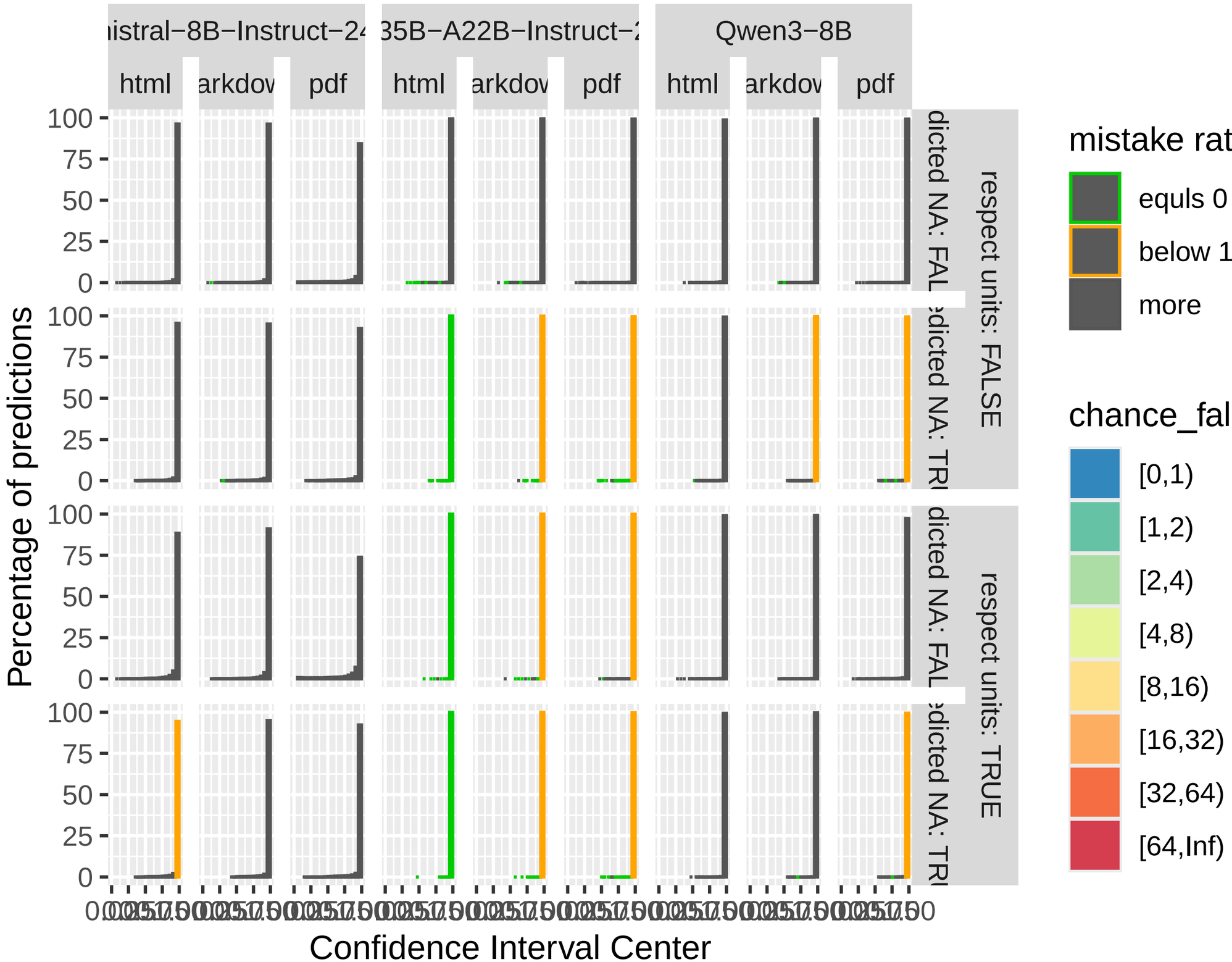
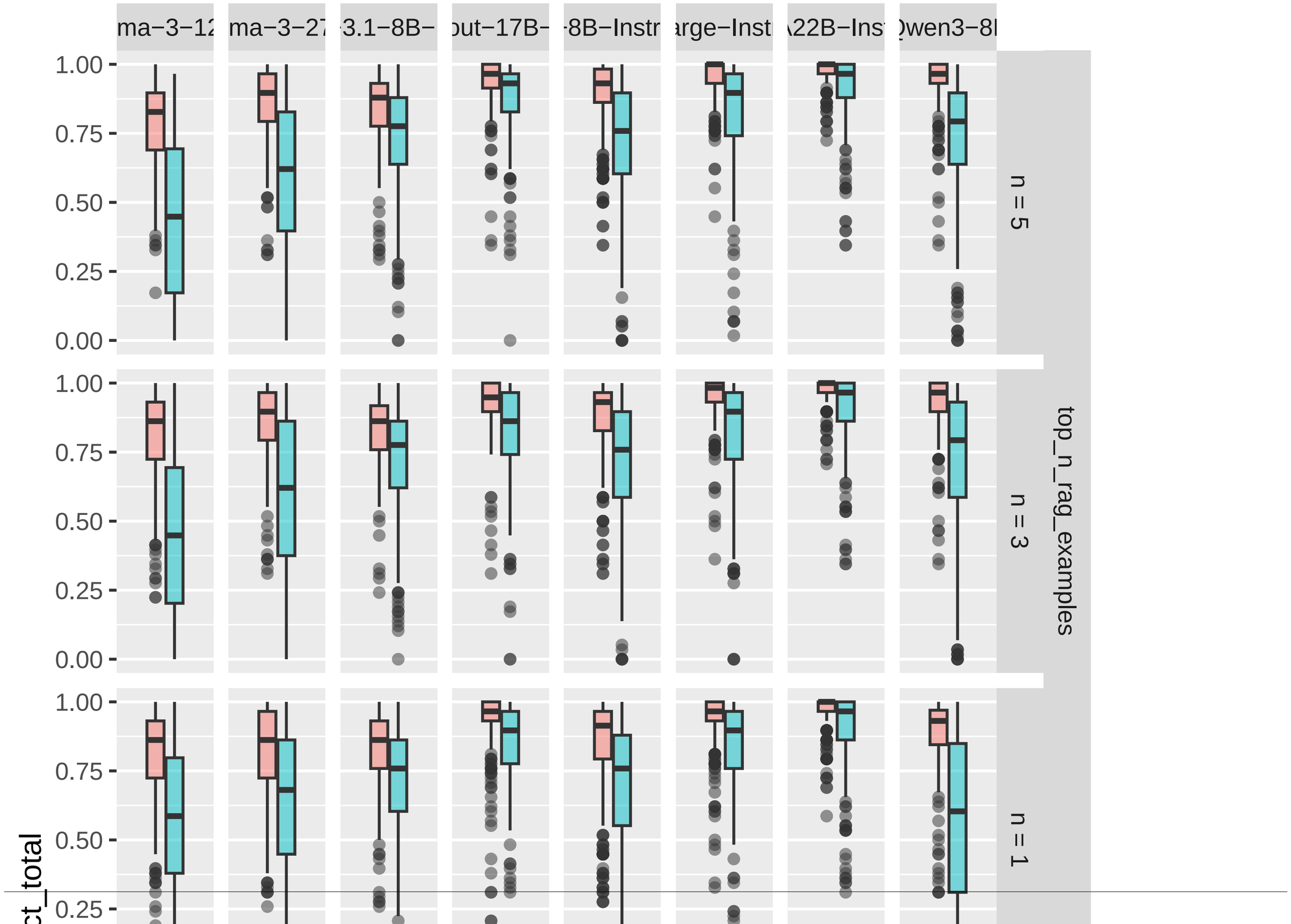
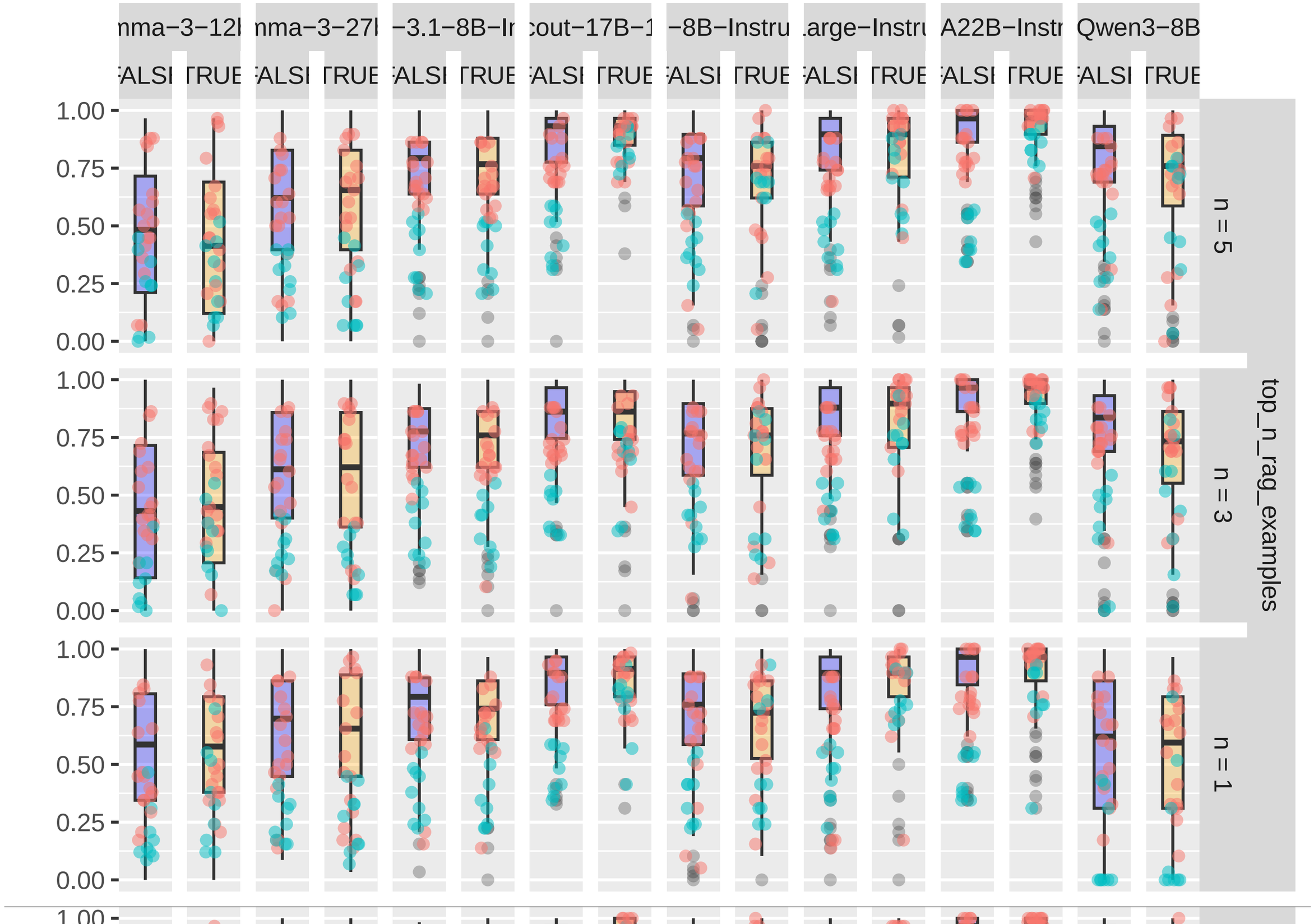
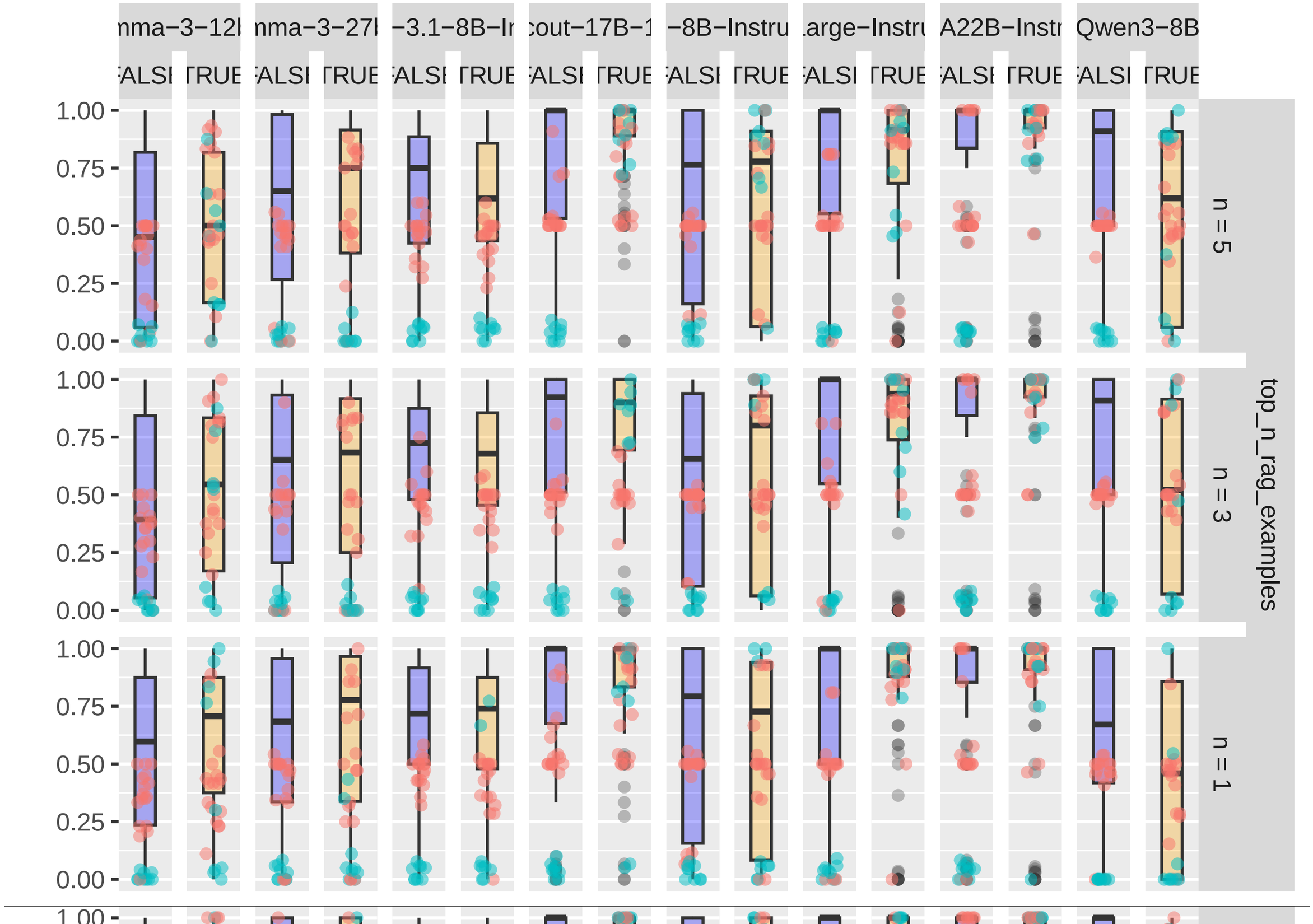
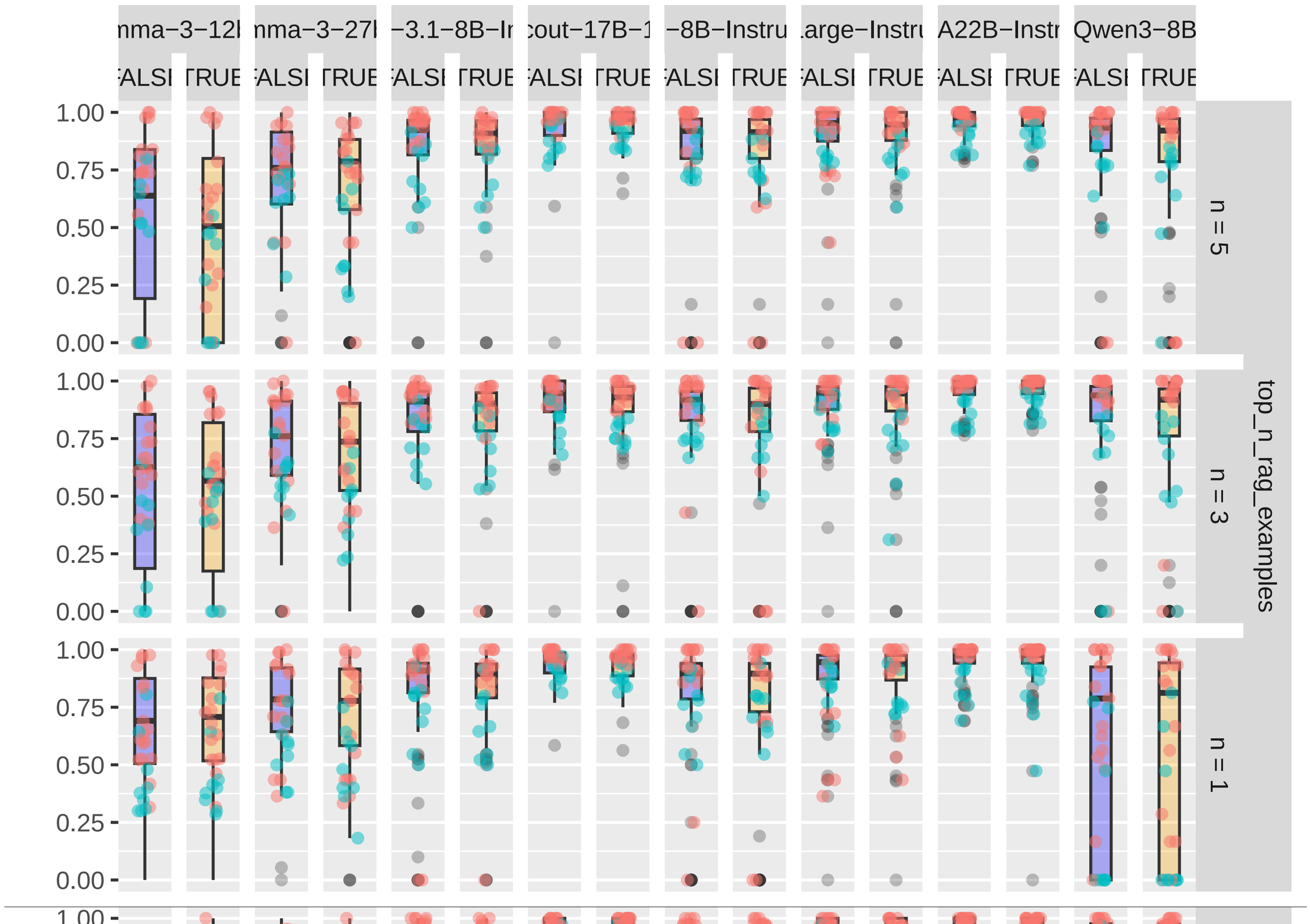


Figure C.26: Estimating the relative frequency to find a wrong extraction result over different confidence intervals for predictions for the synthetic table extraction task. Additionally grouped by input format.


ma−3−12
ma−3−27
3.1−8B−
out−17B−
8B−Instr
arge−Inst
22B−Inst
Qwen3−8B
1.00
0.75
0.50
0.25
0.00
n = 5
n = 3
n = 1
top_n_rag_examples
ct_total


nma−3−12
nma−3−27
−3.1−8B−In
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−3−12
nma−3−27
−3.1−8B−In
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−3−12
nma−3−27
−3.1−8B−In
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

Chapter D

Layout testing

12sdgsdgj

Landscape Landscape

Hello