Extraction of tabular data from annual reports with LLMs

Using in context learning with open source models and RAG

submitted by

Simon Schäfer

Matr.-Nr.: 944 521

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

Master of Science (M.Sc.)

in the field of

Data Science

Date of submission September 1, 2025

Gutachter

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

Berliner Hochschule für Technik

Studiere Zukunft

Abstract

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 three 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.

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

Contents

Contents i

1 Introduction 1

1.1 Motivation 1
1.2 Objectives 1
1.3 Methodology (1 p) 2
1.4 Thesis Outline (0.5 p) 2
1.5 To place in chapters above 2
1.6 RHvB 3
1.7 Datenverfügbarkeit 3
1.8 Unstrukturierte Daten 3
1.8.1 Portable Document Format 3

2 Literature review (less than 10 p) 5

2.1 NLP history 5
2.2 Basic terms 5
2.3 Supervised Learning Approaches 5
2.3.1 Generalized Linear Models 5
2.3.2 Random Forest 5
2.3.3 Large Language Models 6
2.3.4 Information extraction 6
2.4 Data balancing 6
2.4.1 Under sampling, oversampling 6
2.5 Evaluation Metrics 6
2.5.1 For classification 6
2.5.2 For regression 6
2.6 Technological topic (related work) 6
2.7 Term frequency 7
2.7.1 Extraction of numeric values 7

2.8 optinal more topics like previous . . . 7
2.9 Summary (0.5 p) . . . 7
2.10 To place in chapters above . . . 7
2.11 Table extraction tasks . . . 7
2.11.1 Difficulties . . . 7
2.12 Document Extrtaction Process . . . 7
2.12.1 Document Layout Analysis . . . 7
2.12.2 . . . 8
2.13 Tools . . . 8
2.13.1 TableFormer . . . 8

3 Methods **9**
3.1 Data . . . 9
3.2 Page identification . . . 9
3.2.1 Baselines . . . 9
3.3 Table detection . . . 10
3.3.1 LLM . . . 10
3.3.2 Vision Model . . . 10
3.3.3 Docling and Co . . . 10
3.4 Information extraction . . . 10
3.4.1 Baselines . . . 10
3.4.2 Simple pipeline . . . 10
3.4.3 Sophisticated approaches . . . 10

4 Implementation (max 5p) **11**
4.1 Speedup with vLLM and batching . . . 11
4.2 Setup (Dockerfile and PV) . . . 11

5 Results **13**
5.1 Page identification . . . 13
5.1.1 Baseline: Regex . . . 15
5.1.2 Table of Contents understanding . . . 17
5.1.3 Classification with LLMs . . . 23
5.1.4 Term frequency based classifier . . . 39
5.1.5 Comparison . . . 44
5.2 Table extraction . . . 47
5.2.1 Baseline: Regex . . . 48
5.2.2 Extraction with LLMs . . . 51
5.2.3 Comparison . . . 64

6 Discussion **67**
6.1 Limitations . . . 67
6.1.1 table extraction . . . 67
6.1.2 classification . . . 67
6.2 Not covered . . . 67
6.3 Outlook . . . 68
6.3.1 Table extraction . . . 68

7 Conclusion **69**

References **71**

List of Figures **73**

List of Tables **77**

Glossary **79**

A Appendix **81**
A.1 Local machine . . . 81
System Details Report . . . 81
A.2 Benchmarks . . . 82
A.2.1 Text extraction . . . 82
A.2.2 Table detection . . . 82
A.2.3 Large language model process speed . . . 86
A.3 Prompts . . . 86
A.3.1 TOC understanding . . . 86
A.3.2 Classification . . . 87
A.4 Regular expressions . . . 97
A.5 Tables . . . 98
A.5.1 Classification . . . 98
A.6 Figures . . . 98
A.6.1 Page identification . . . 98
A.6.2 Table extraction . . . 103
A.7 Annual Comprehensive Financial Report Balance Sheet . . . 118
A.8 Extraction framework flow chart . . . 118
A.9 Table extraction with regular expressions . . . 118

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

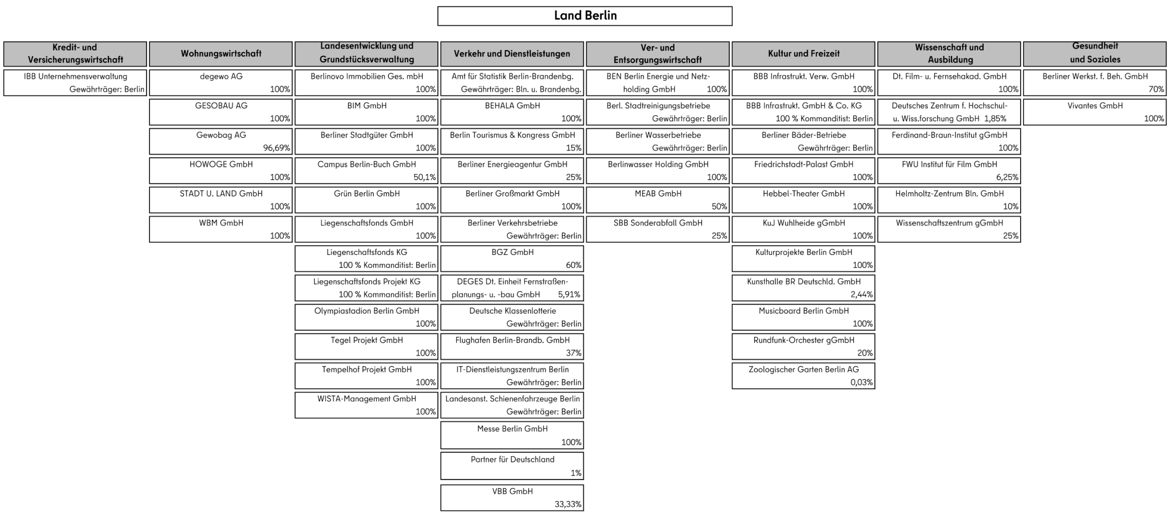


Figure 1.1: Overview of companies Berlin holds share at

1.2 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 (large language model) 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

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.

2.3.3 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

Methods

norm gpu hours

3.1 Data

- companies Beteiligungsbericht
- number found Jahresberichte
- number used Jahresberichte first rows
- number used Jahresberichte Aktiva Tabellen

3.2 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.

3.2.1 Baselines

3.2.1.1 Regex based

results potentially dependend on package used for text extraction (Auer et al., 2024, p. 2 f.)

- PyMuPDF
- pypdf
- docling-parse
- pypdfium
- pdfminer.six

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

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

3.2.1.2 Term frequency based

VLLM based was not implemented

3.3 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

3.3.1 LLM

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

3.3.2 Vision Model

Yolo

3.3.3 Docling and Co

VLLM based was not implemented

3.4 Information extraction

3.4.1 Baselines

simple regex?

3.4.2 Simple pipeline

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

3.4.3 Sophisticated approaches

not implemented

- with pipelines
- Nougat
- maker
- Azure
- docling

Chapter 4

Implementation (max 5p)

4.1 Speedup with vLLM and batching

4.2 Setup (Dockerfile and PV)

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.23). For longer documents they propose to use the TOC (table of contents) 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.

[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.

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.

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.

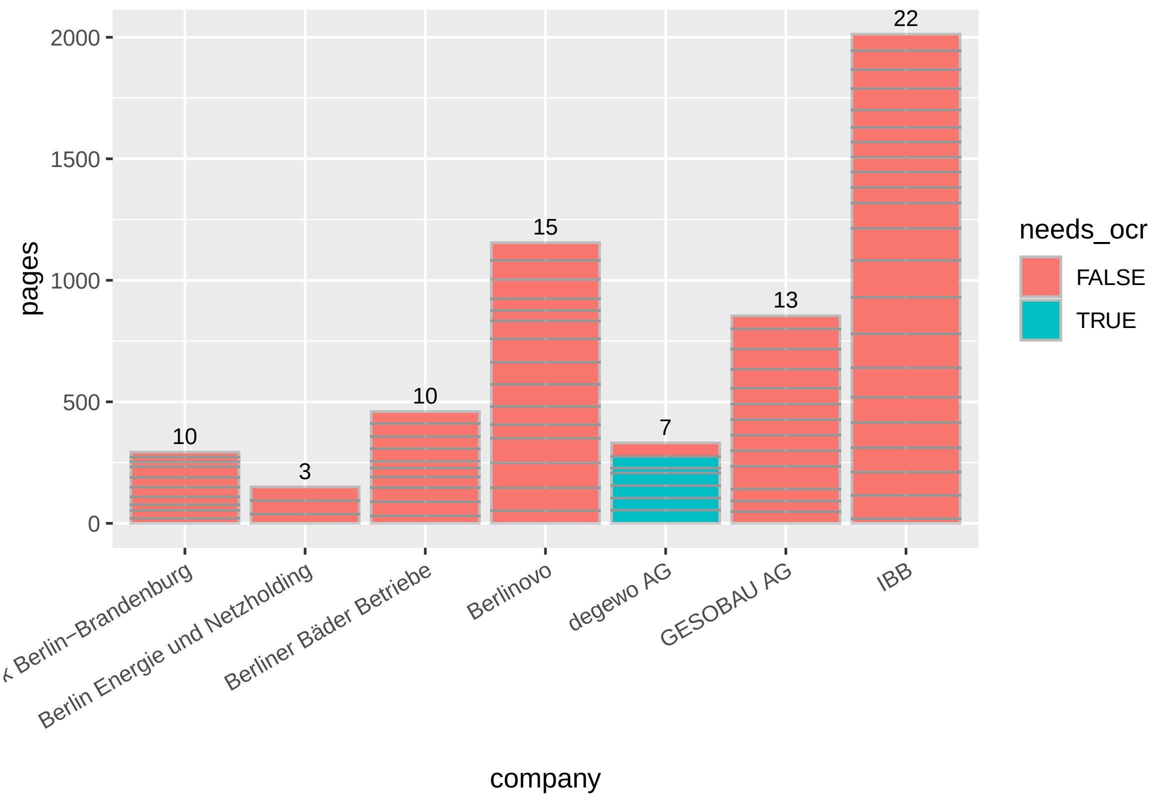


Figure 5.1: Showing the number of pages (bar height) and number of documents (number above the bar) per company for the data used for the page identification task. Some documents would require ocr before being processed and were not used.

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.

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.

[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 5.1: 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

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

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?

[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 A.1 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().

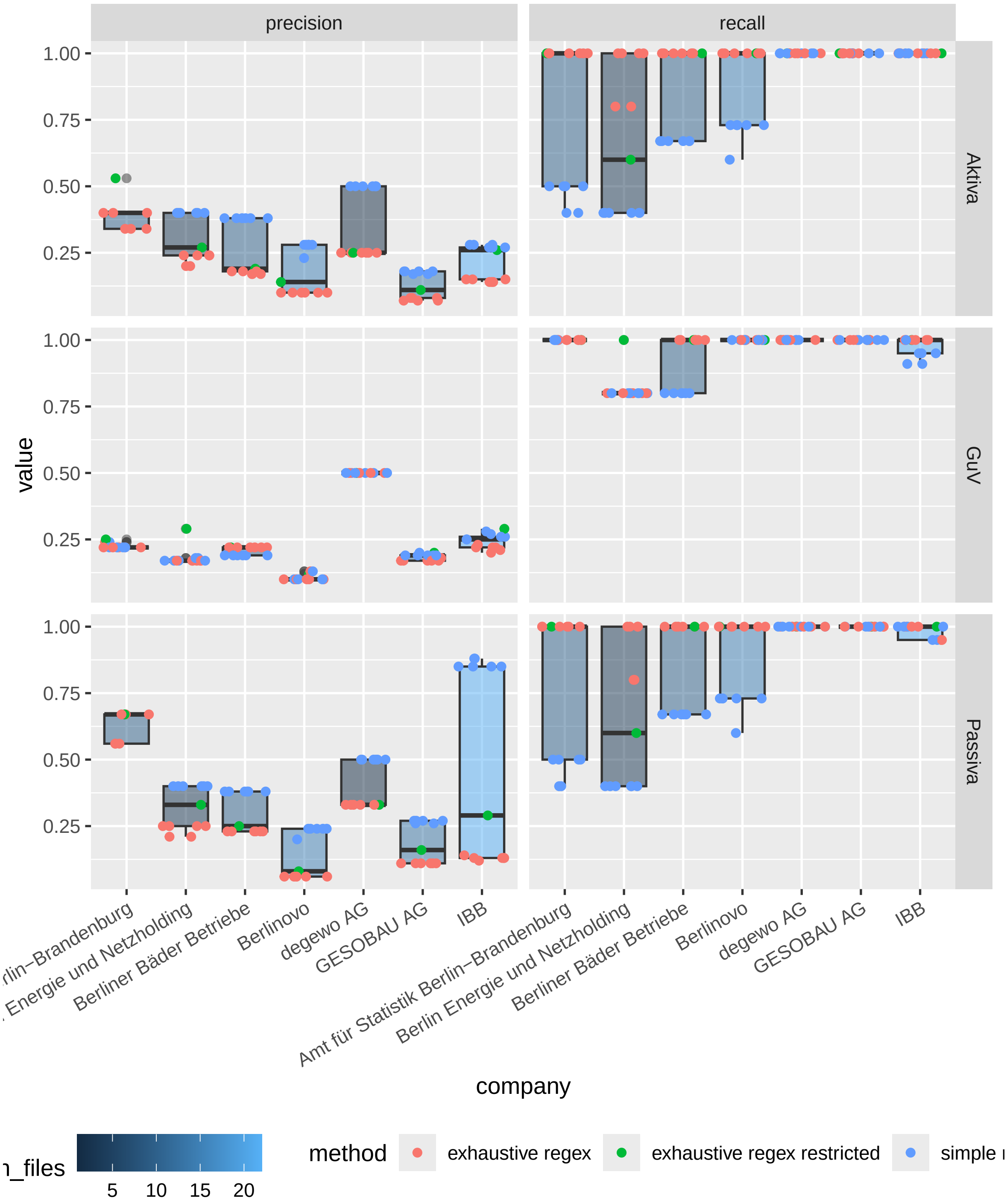


Figure 5.2: Comparing the performance among different companies.

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 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.

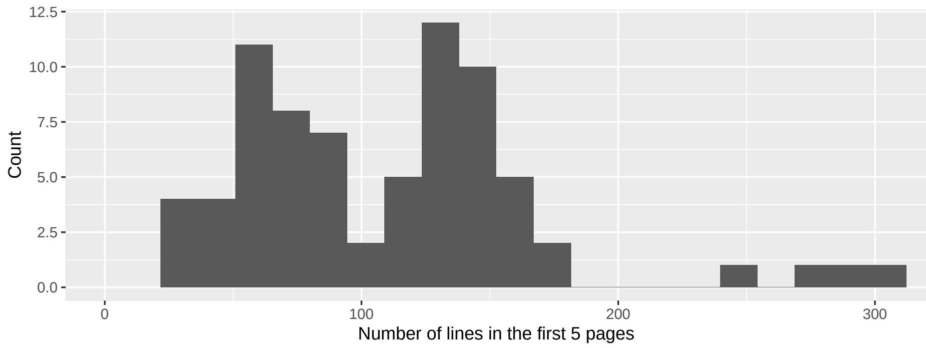


Figure 5.3: Histogram of the number of lines in the first 5 pages of the annual reports

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

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

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

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 %.

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 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.

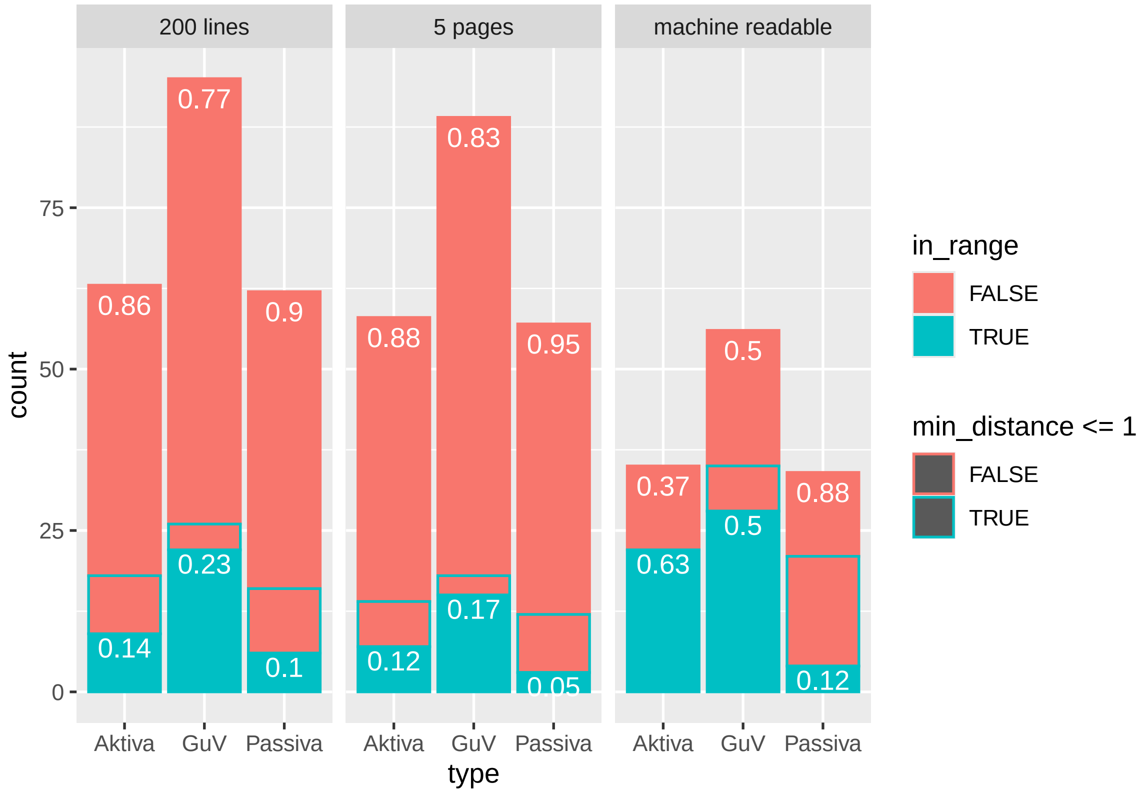


Figure 5.4: Comparing number of found TOC and amount of correct and incorrect predicted page ranges

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 A.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

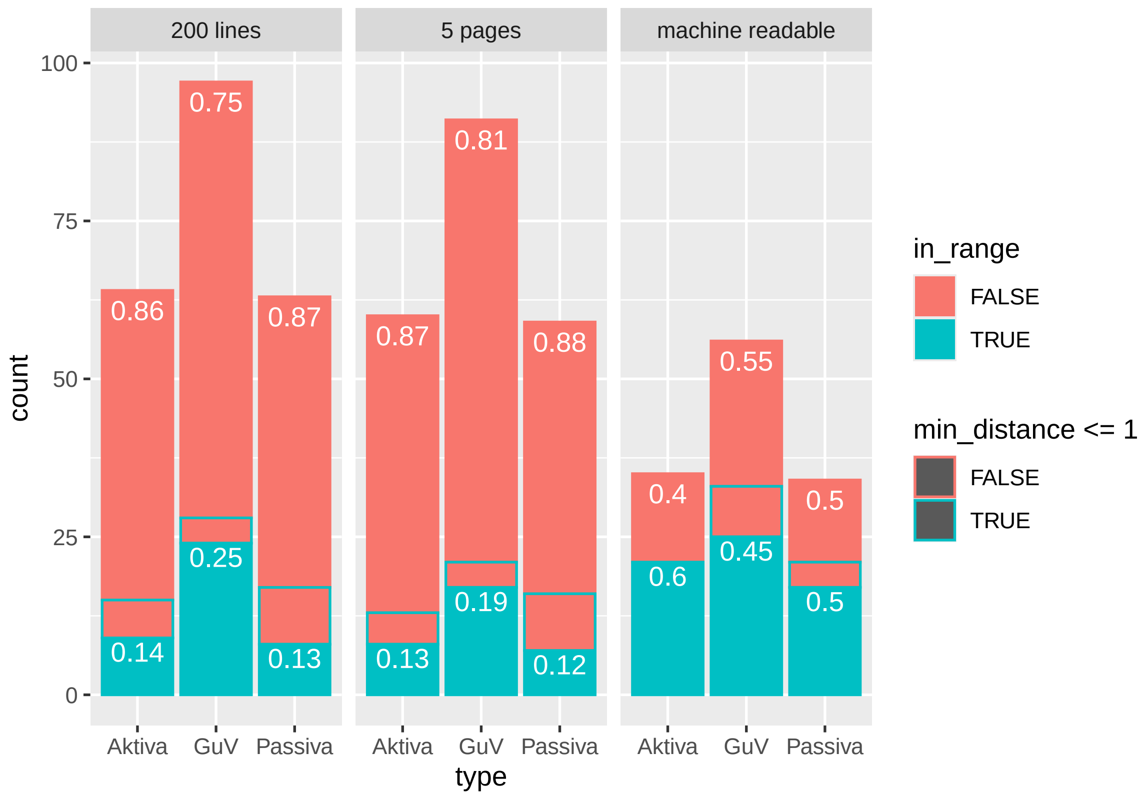


Figure 5.5: Comparing number of fount TOC and amount of correct and incorrect predicted page ranges

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}

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

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.

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.

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

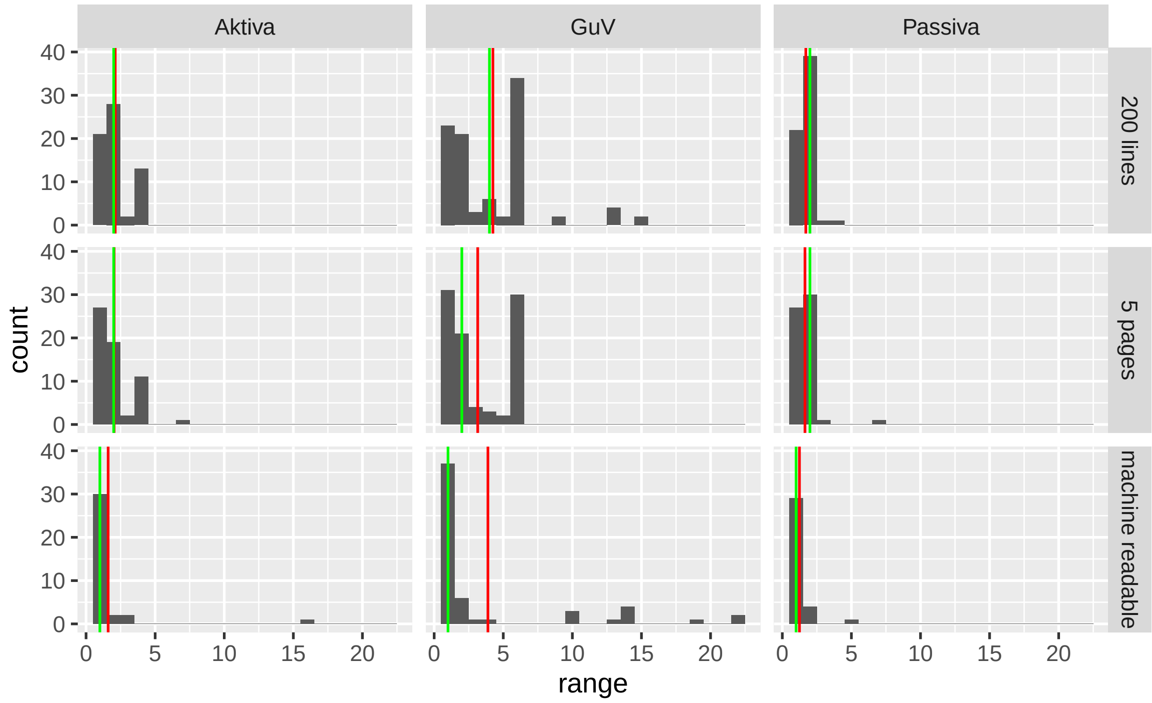


Figure 5.6: Comparint the predicted page range sizes. The red vertical line shows the mean and the green one shows the median of these sizes.

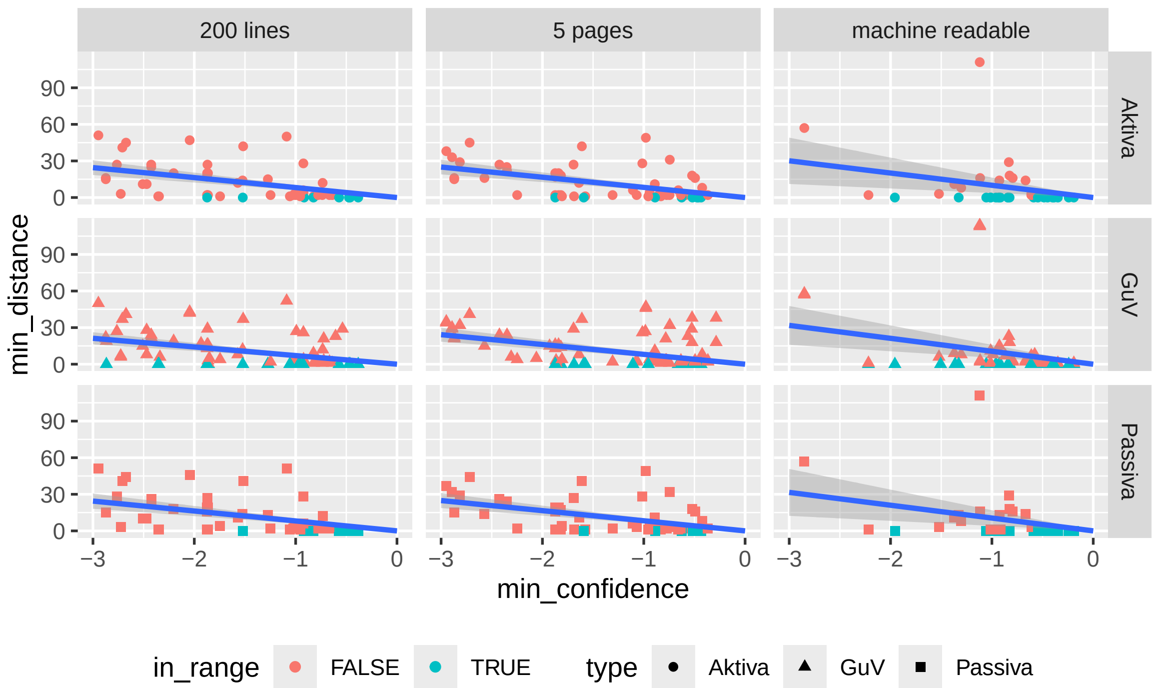


Figure 5.7: Showing the minimal distance of the predicted page range to the actual page number overthe logprobs of the models response confidence.

It has no big influence on the predictions, if the TOC is passed formatted as markdown or json. With the json formatted TOC it found two more correct page ranges[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.

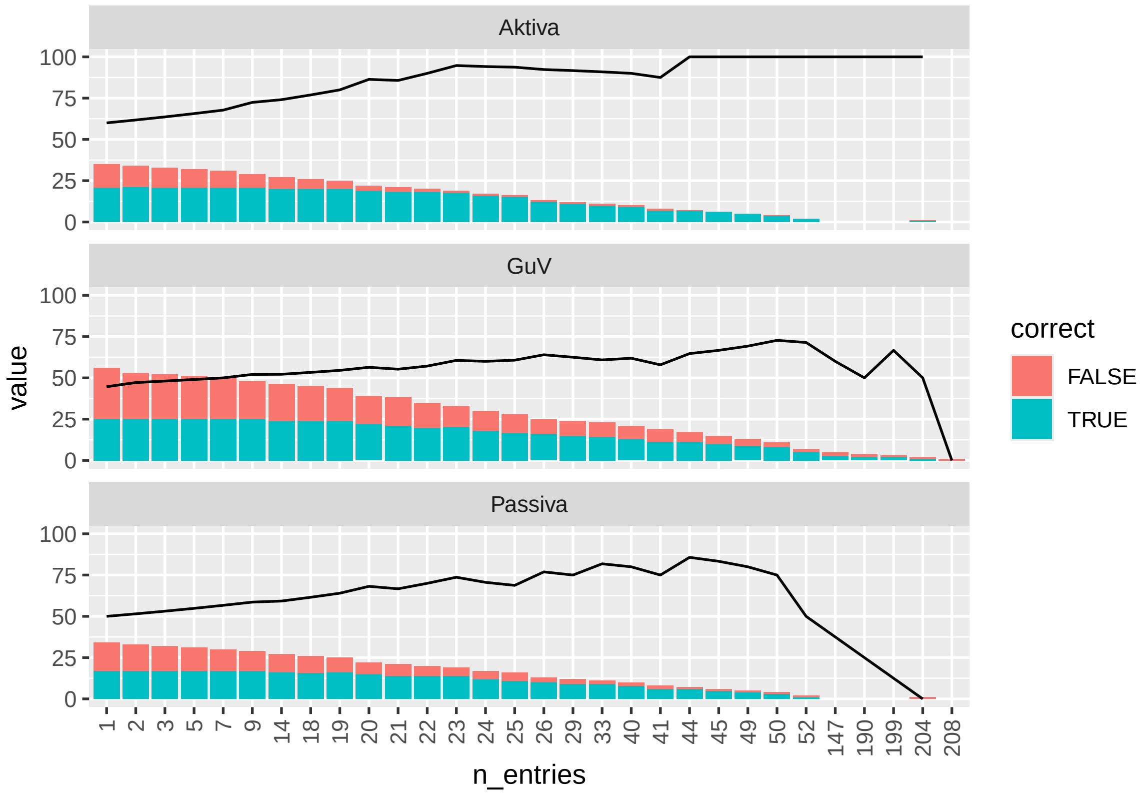


Figure 5.8: Showing the amount of correct and incorrect predicted page ranges (bars) and the percentage of correct predictions (black line).

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*. 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.

[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.

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.

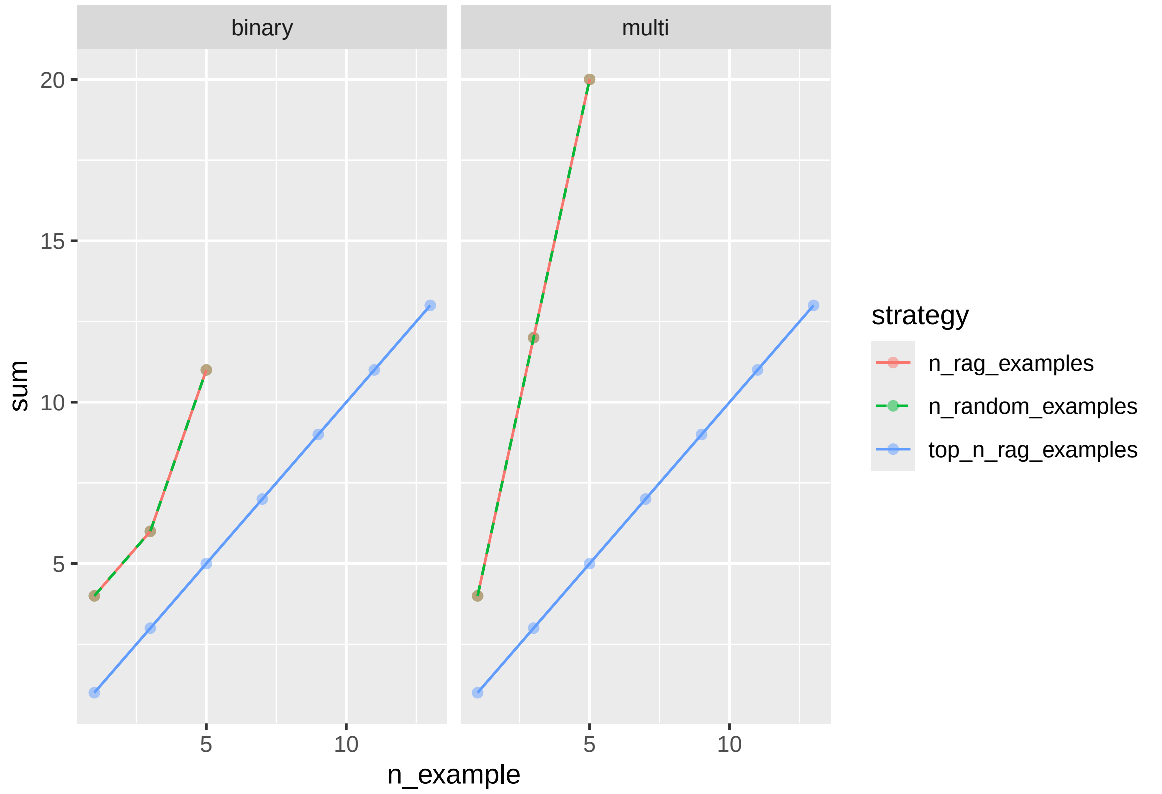


Figure 5.9: Comparing the actual number of provided examples depending on the classification type, example selection strategy and chosen parameter n-examples. The slope for the top-n-rag-examples strategy is the same for both approaches. The line for the strategies n-random-examples and n-rag-examples 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 (Virtual Large Language Model) library, accessed via hugging face and fits into the combined VRAM of

[10]See also Table A.3.

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

8 nvidia B200 graphic cards (1.536 TB). The models cover a wide range of (active) parameter sizes. Especially for the Qwen family many models of different parameter sizes are used in the benchmark, to investigate if there is a clear minimum amount of parameters needed, to solve the classification task.

The results of the benchmarks have been logged as json (JavaScript Object Notation) 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

[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.

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	runtim
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	

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 (*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 A.3.

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

[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.

[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.

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.

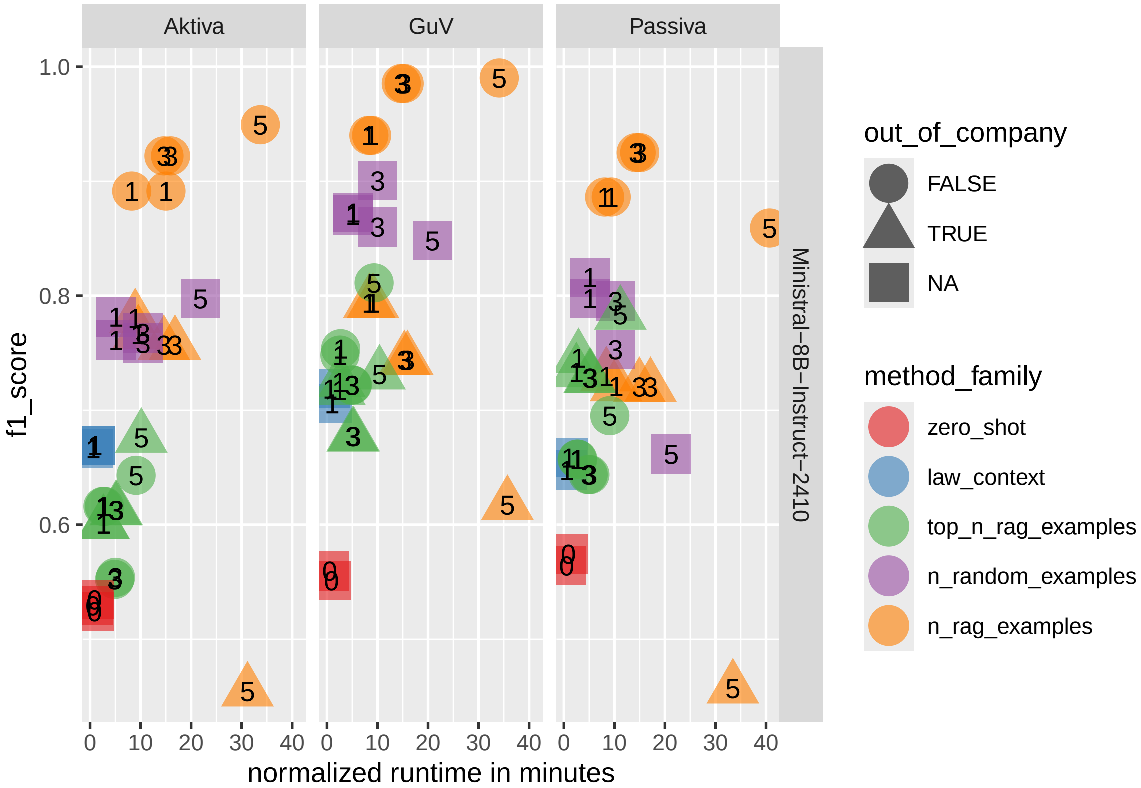


Figure 5.10: Showing F1 score performance over normalized runtime for binary classification for Mistral-8B-Instruct-2410.

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 comparability among the different strategies. Additionally, it shows results for the *top_n_rag_example* strategy with *n_examples* up to 13. The F1 score of the *top_n_rag_example* strategy stays lower than the F1 score of the *n_rag_examples* strategy, even though there are more examples used. This is mainly caused by lower precision scores, probably because there are no contrasting examples provided.

Figure A.4 and Figure A.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

[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.

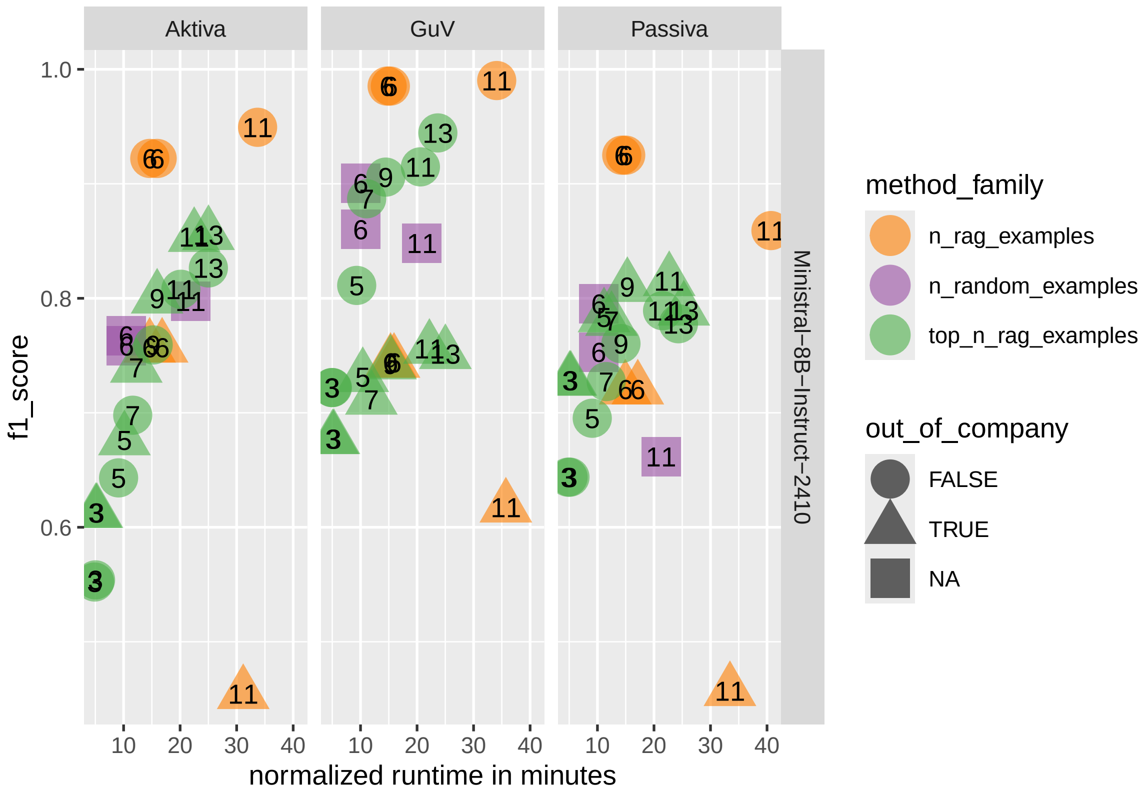


Figure 5.11: Showing F1 score performance over normalized runtime for binary classification for Mistral-8B-Instruct-2410. Comparing the performance based on the real number of provided examples.

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*.

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 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**.

[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.

[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.

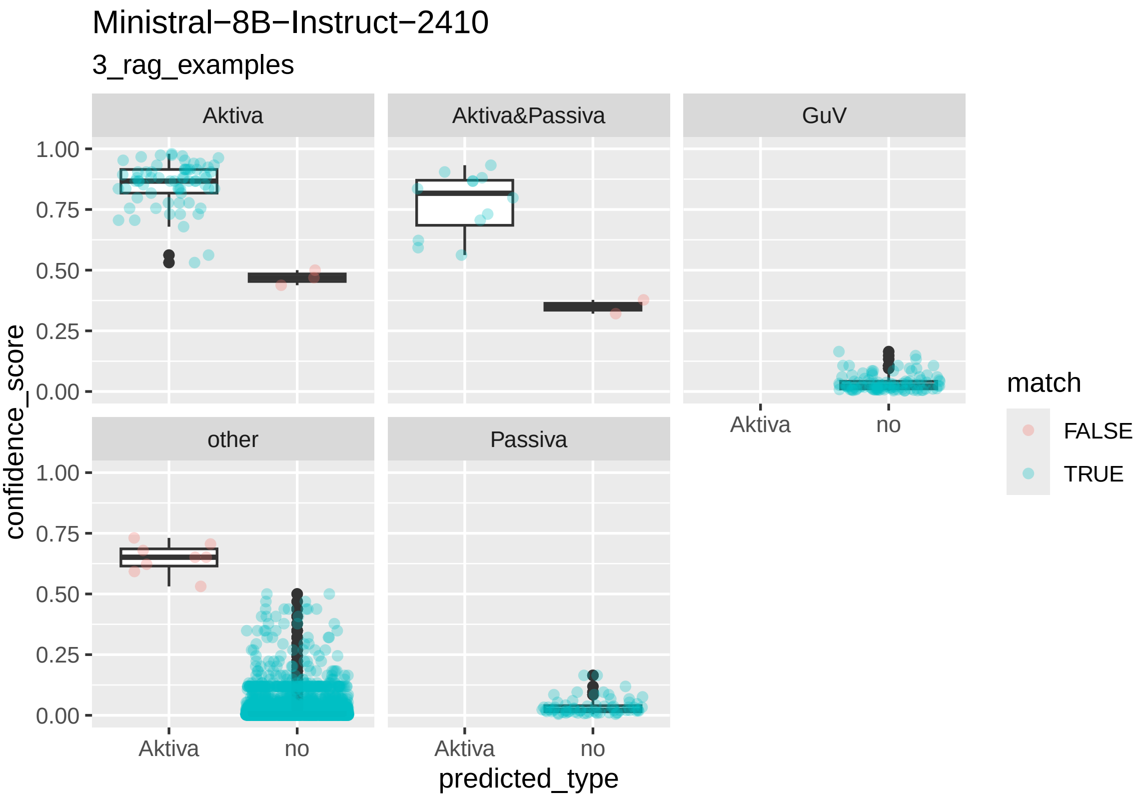


Figure 5.12: Showing the confidence score for the Aktiva classification task grouped by table type and correctness for Mistral-8B-Instruct-2410.

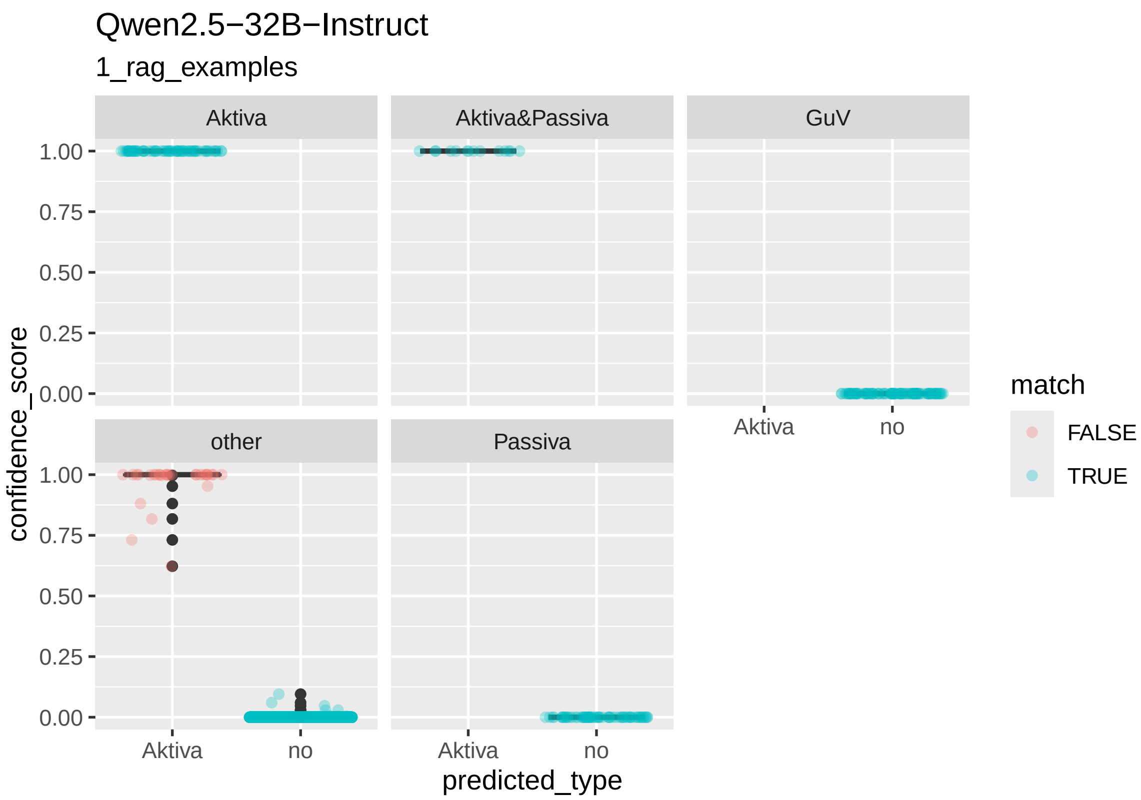


Figure 5.13: Showing the confidence score for the Aktiva classification task grouped by table type and correctness for Qwen-2.5-32B-Instruct.

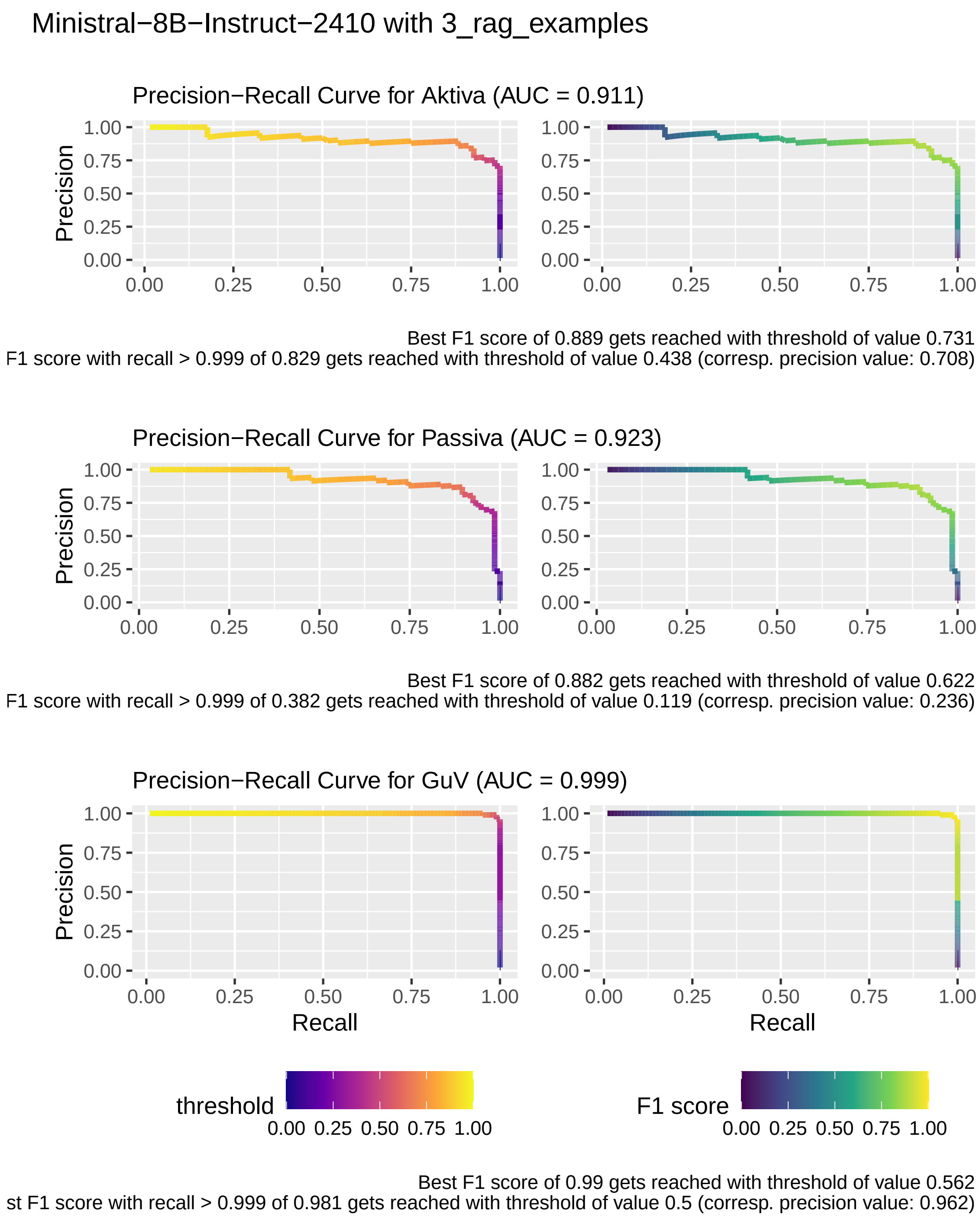


Figure 5.14: Showing the precsion-recall-curve for the best performing model.

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.

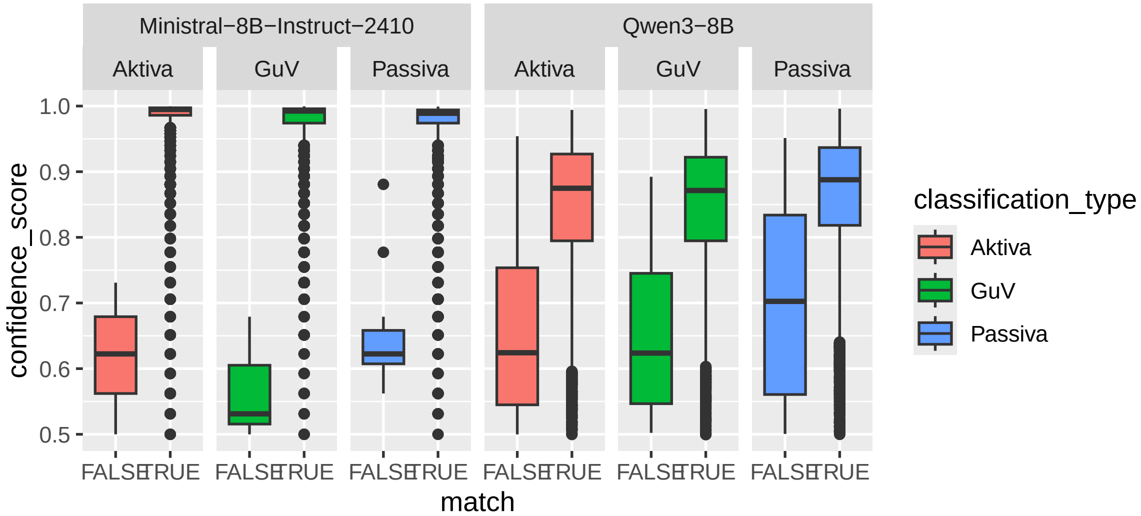


Figure 5.15: Comparing the reported confidence scores for the page identification task for the Mistral and Qwen 3 with 8B parameters.

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 5.17 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

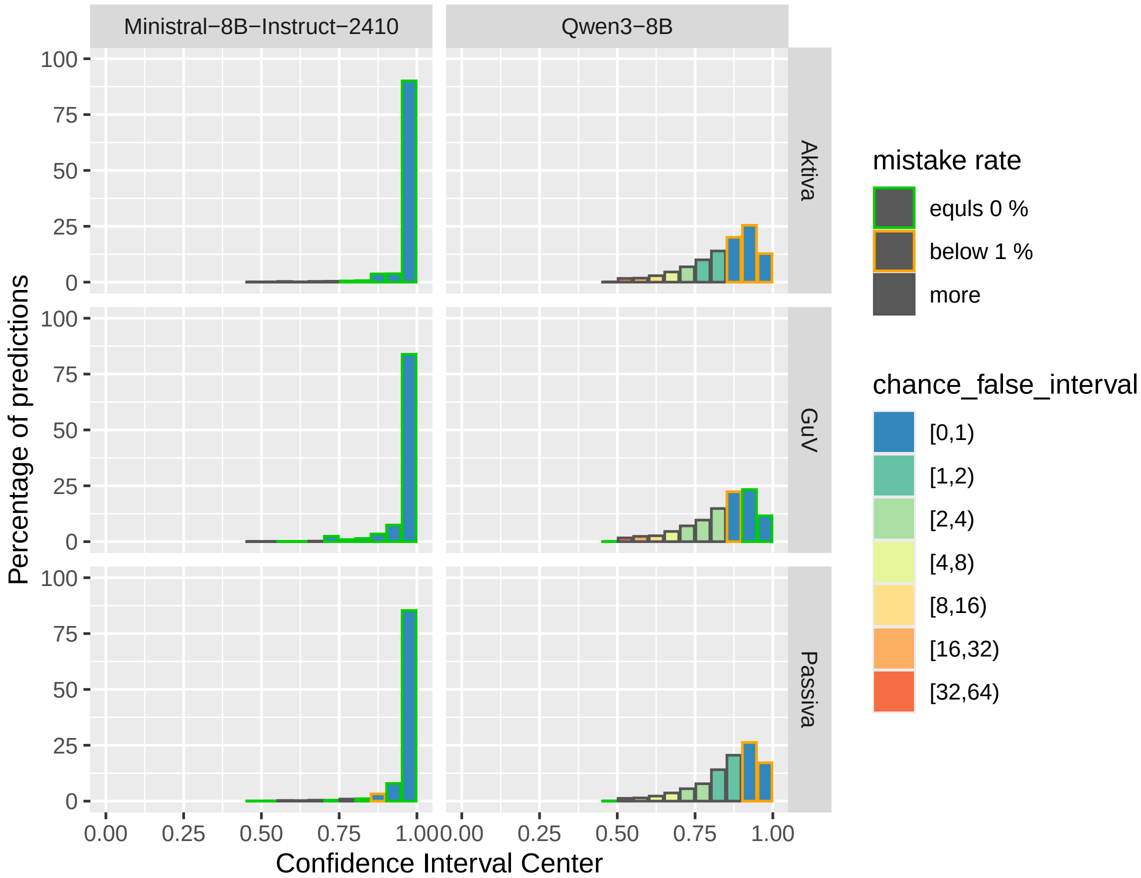


Figure 5.16: Estimating the relative frequency to find a wrong classification over different confidence intervals

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 in s
meta-llama	Llama4Scout17B16EInstruct	Aktiva	n_rag_examples	3	1	4451
meta-llama	Llama4Scout17B16EInstruct	GuV	n_rag_examples	1	1	2541
mistralai	MistralLargeInstruct2411	Passiva	n_rag_examples	1	0.99	7061
meta-llama	Llama4Scout17B16EInstruct	Passiva	n_rag_examples	3	0.99	4451
mistralai	MistralLargeInstruct2411	Aktiva	n_rag_examples	3	0.98	18744
Qwen	Qwen2.532BInstruct	Aktiva	n_rag_examples	3	0.98	5664
Qwen	Qwen3235BA22BInstruct2507	GuV	n_rag_examples	3	0.97	11164
Qwen	Qwen2.572BInstruct	Passiva	n_rag_examples	1	0.97	3908
mistralai	MistralLargeInstruct2411	GuV	n_rag_examples	1	0.96	7061
google	gemma327bit091	Aktiva	n_rag_examples	3	0.89	4297
google	gemma327bit091	Passiva	n_rag_examples	1	0.82	2606
google	gemma327bit091	GuV	n_rag_examples	1	0.79	2606
tiiuae	Falcon310BInstruct	GuV	n_rag_examples	1	0.71	868
tiiuae	Falcon310BInstruct	Aktiva	n_rag_examples	3	0.71	2393
microsoft	phi4	Passiva	n_rag_examples	2	0.67	1660
microsoft	phi4	Aktiva	n_random_examples	1	0.6	493
tiiuae	Falcon310BInstruct	Passiva	top_n_rag_examples	3	0.59	494
microsoft	phi4	GuV	n_rag_examples	1	0.46	1725

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

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 A.3.

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

[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.

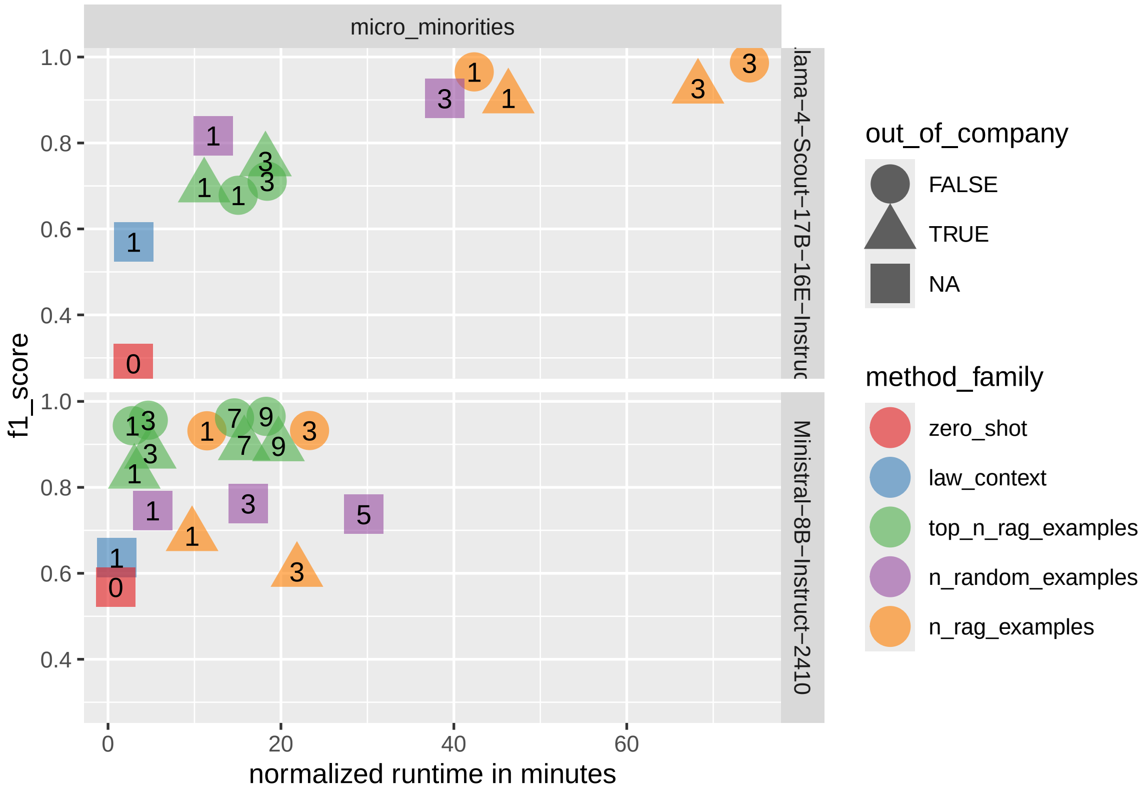


Figure 5.17: Comparing F1 score micro averaged for the minority classes for two models over their normalized runtime.

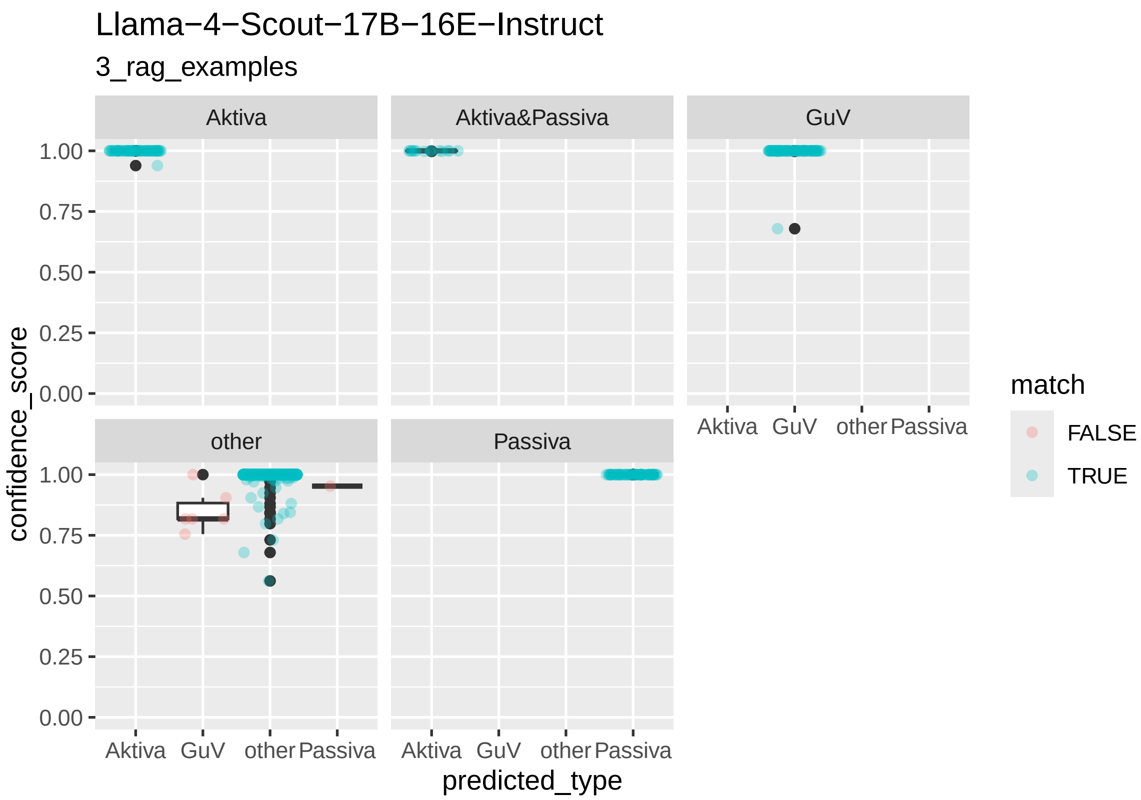


Figure 5.18: Showing the reported confidence scores for all predictions of Llama 4 Scout grouped by the true target type. Errors have only been made within the majority class.

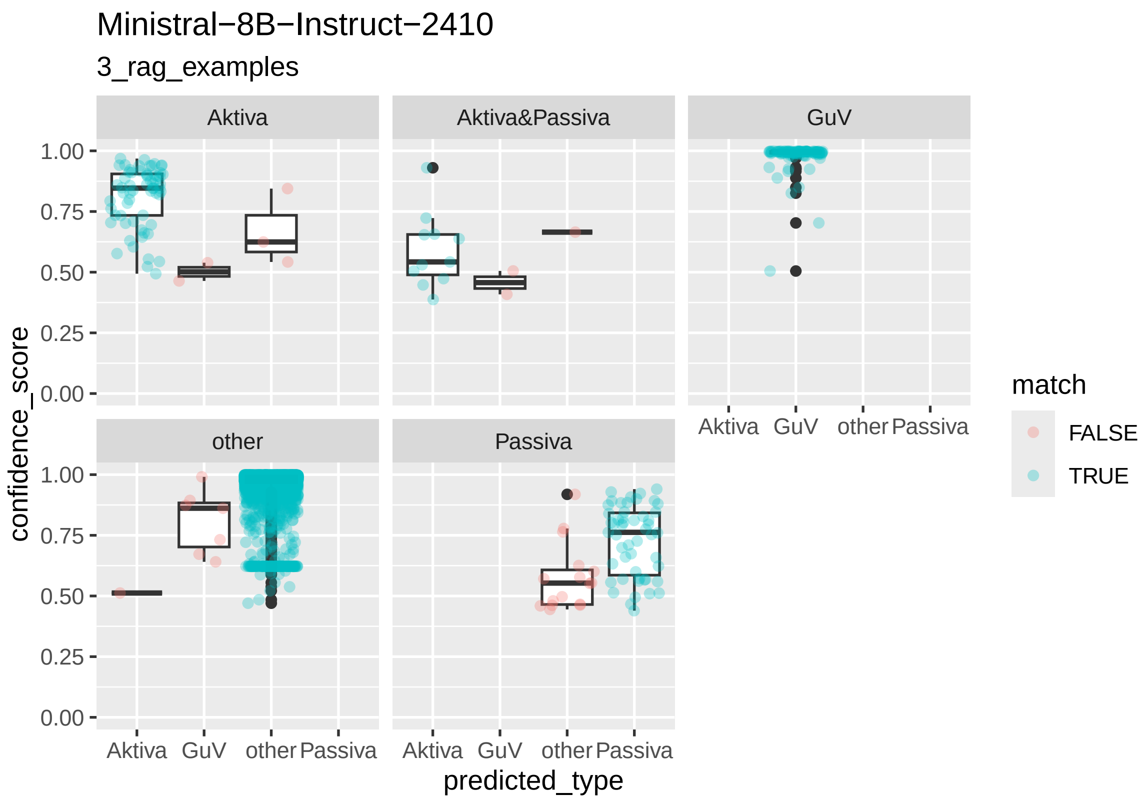


Figure 5.19: Showing the reported confidence scores for all predictions of Ministral 8B grouped by the true target type. Errors have only been made within the majority class.

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 A.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 %.

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.

[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.

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

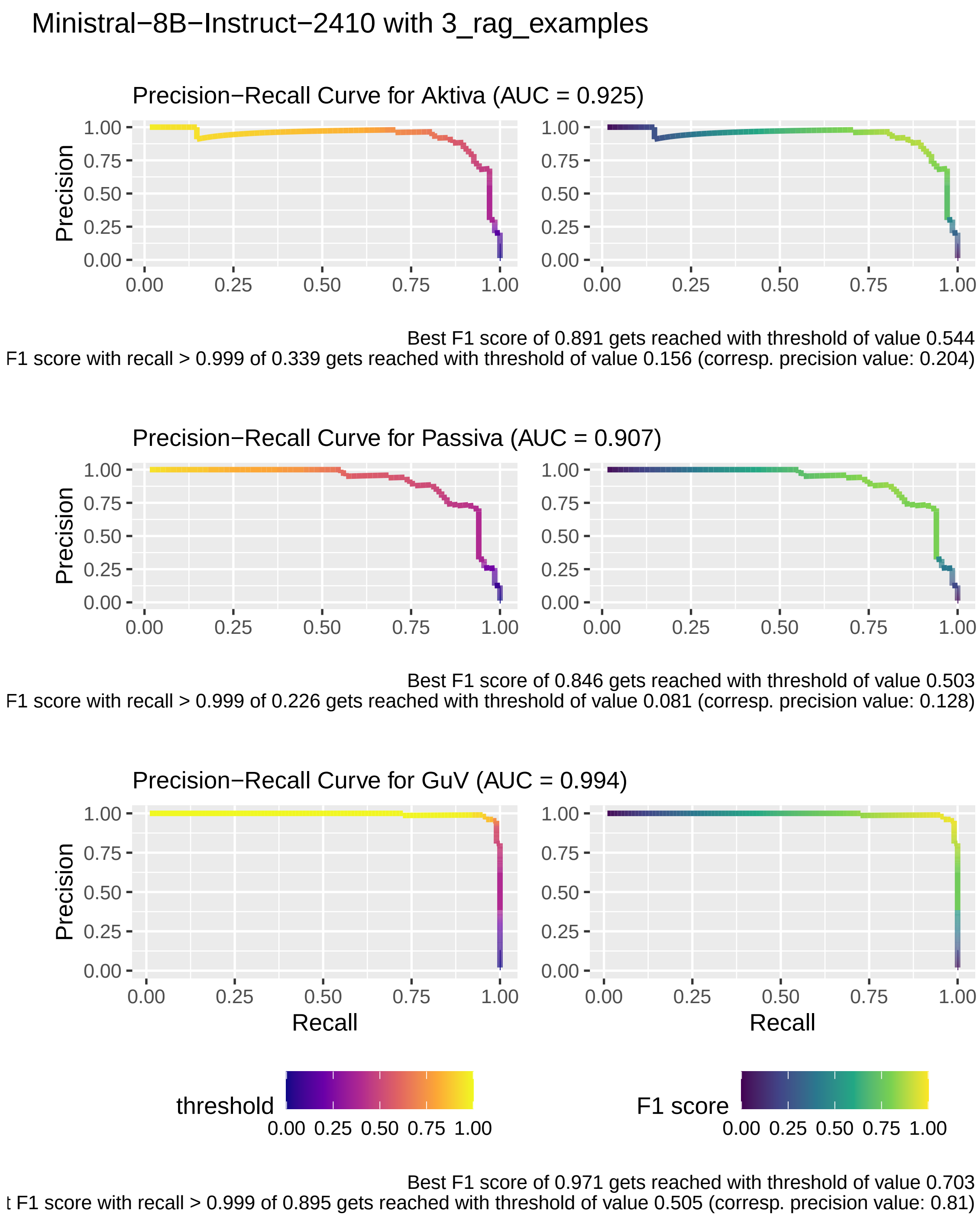


Figure 5.20: Showing the precsion-recall-curve for Ministral-8B-Instruct-2410.

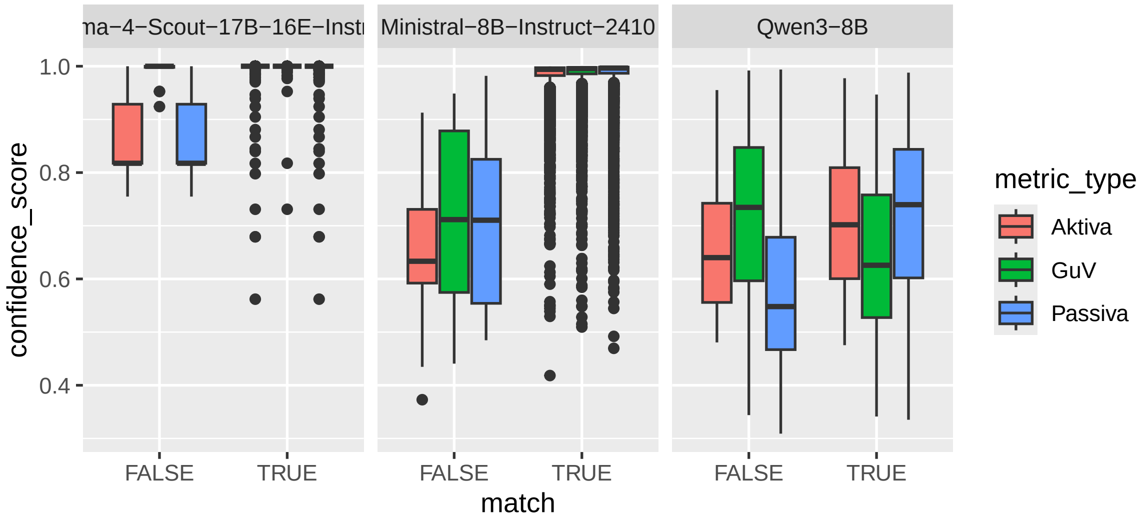


Figure 5.21: Comparing the reported confidence scores for the multi-class page identification task for the Mistral and Qwen 3 with 8B parameters.

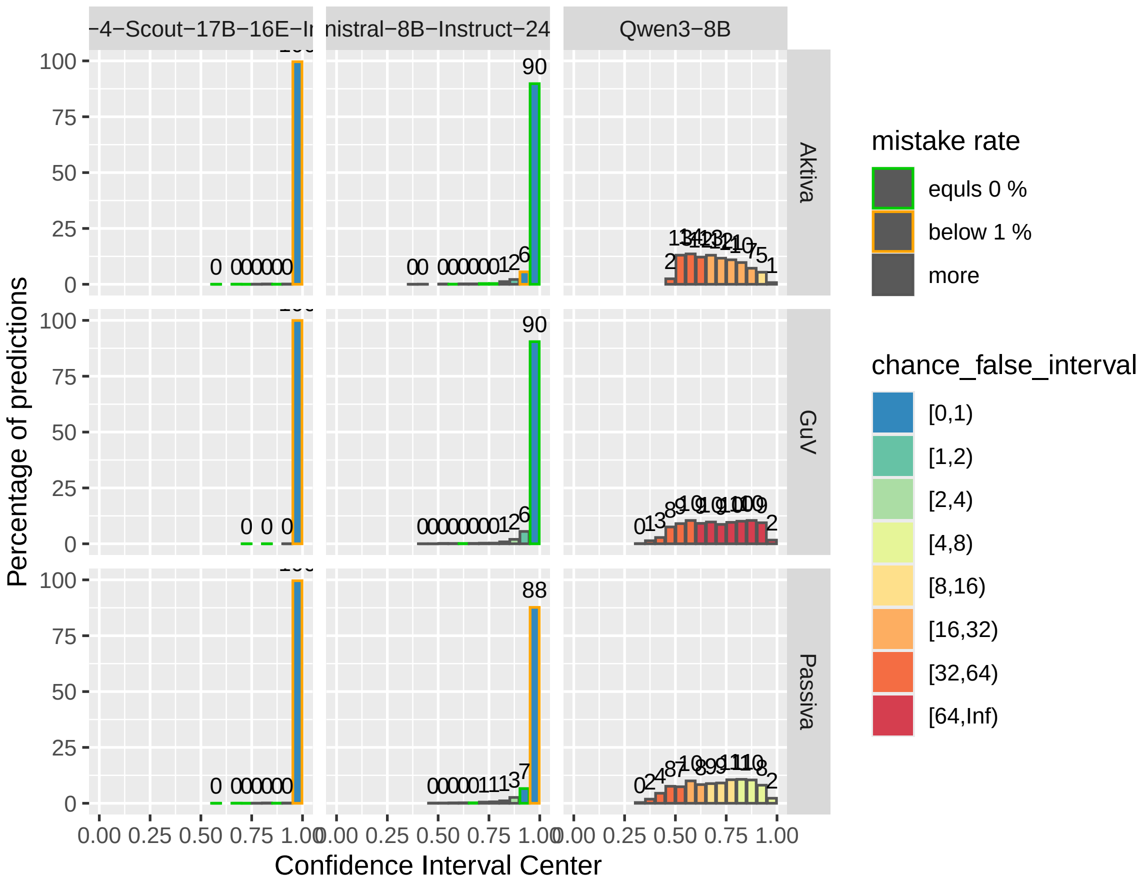


Figure 5.22: Estimating the relative frequency to find a wrong classification over different confidence intervals for the multi-class classification task.

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?)

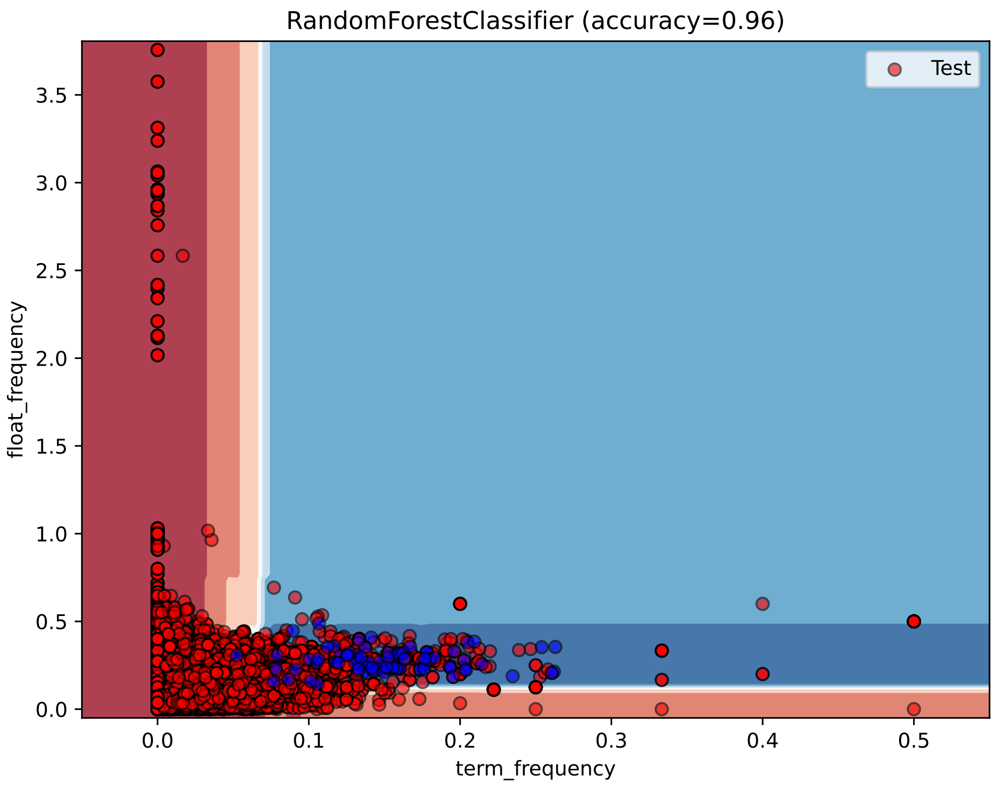


Figure 5.23: Classification map showing which score a data point gets based on its term and float frequency and which type the data points in the test dataset actually have.

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 $n = 7$.

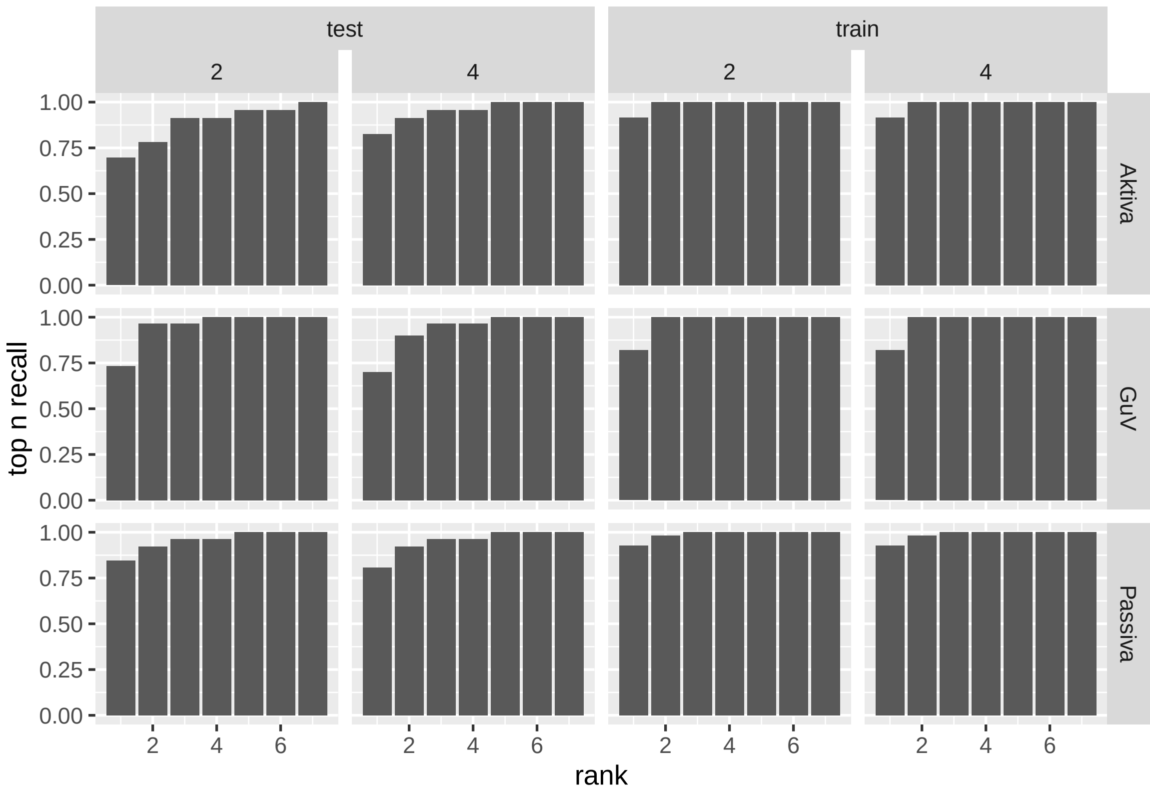


Figure 5.24: Comparing the top n recall on training and test dataset among the random forest with two and four predictors.

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.

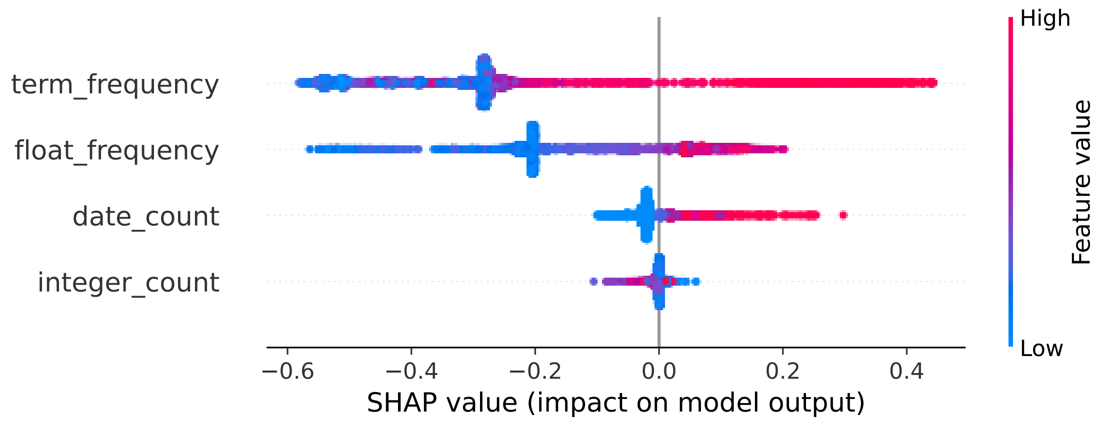


Figure 5.25: Beeswarm plot of SHAP importance values for the four predictors of the second random forest classifier.

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.

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

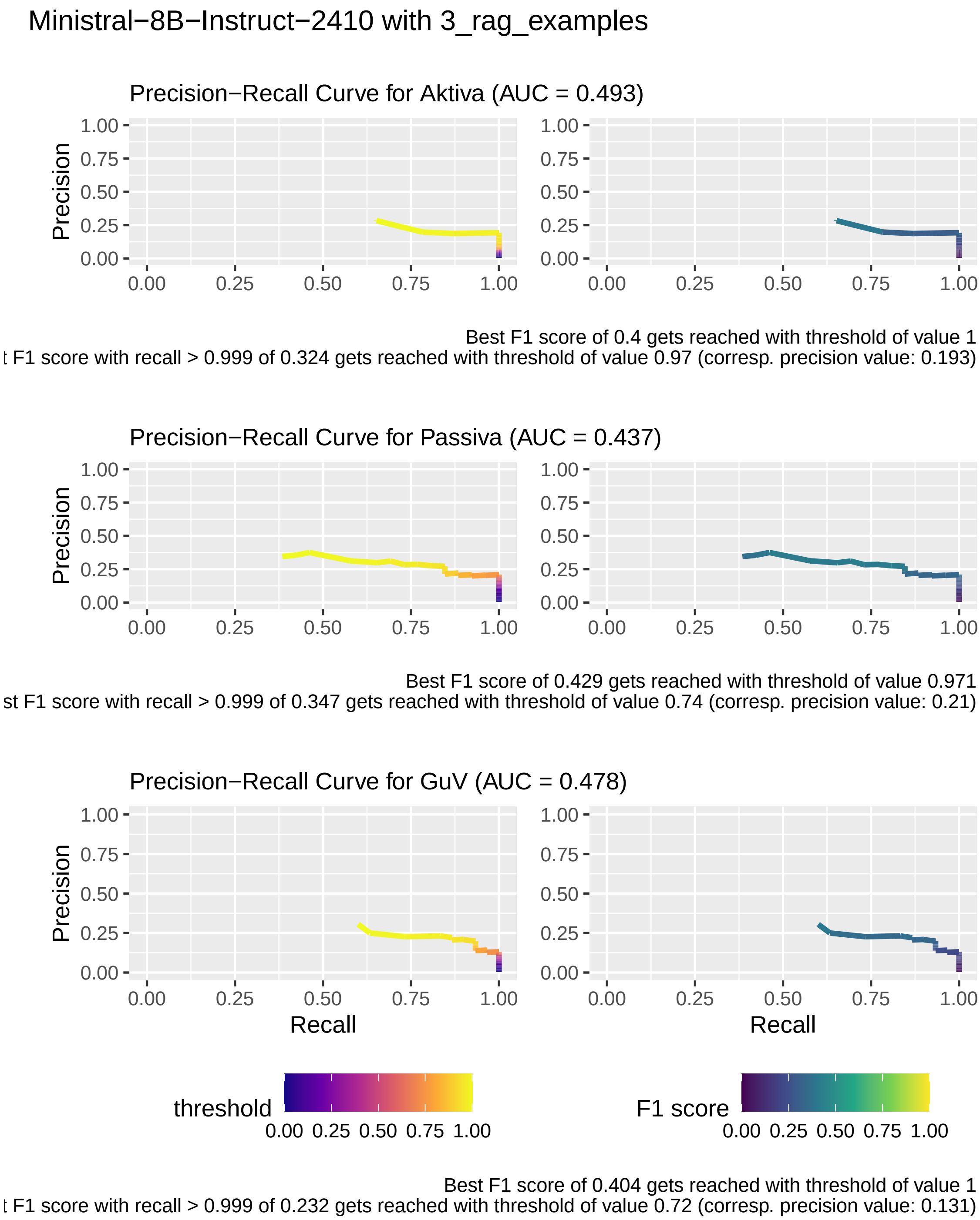


Figure 5.26: Showing the precsion-recall-curve for the random forest with four predictors.

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

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

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 5.14 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
- payment and insurance to pay for a human (e.g . student coworker).
- the training time and energy consumption for training either
  - a LLM (probably done by the LLM provider).
  - a human (growing up, getting educated).
- the energy consumed 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 task to solve is: extract the data from the document.

Which tasks have there been? Which models have been used for which ttask? What data has been used?

[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 CENTS p
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	

5.2.1 Baseline: Regex

The baseline for the table extraction task is set by an approach using regular expressions on the text extract. The approach performs much better[24] on the synthetic dataset compared to the real dataset (see Figure 5.27). Even though, it does not perform perfectly and its performance is more consistent on the text extracted with pymupdf compared to pdfium. Some possible explanations are:

- a duplicated row name[25]
- numeric columns extracted separated from row names by extraction libraries
- sums in the same row as the single values[26]
- with pdfium: missing white space[27]
- with pdfium: random line breaks[28]

You can find some examples for incorrect extracted texts in section A.9.

On the real dataset the approach shows a wider spread for the percentage of correct extracted numeric values as well as a considerable number of annual reports where the extraction did not work at all. Interestingly, the used text extraction library has no noticeable influence on the real dataset.

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). 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 0.7.

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.

Real dataset There are multiple hypotheses that don't get supported by the visual results (see Figure A.2). The pretty surprising results are:

1. The visual separation of columns or rows has an effect on the text processing.
2. It seems to have a positive effect on F1 and numeric correctness rate if the Passiva table is on the same page, even though it has no influence on the single predictions.

[24] A comparison of the numeric values over all methods can be found in section 5.2.3.
[25] 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.
[26] In this case the regex takes the sum as the value for the previous year.
[27] This can form unexpected numeric patterns or prevent the row names to be recognized.
[28] The approach takes care of line breaks between words, but not within. This leads to unrecognized row names as well.

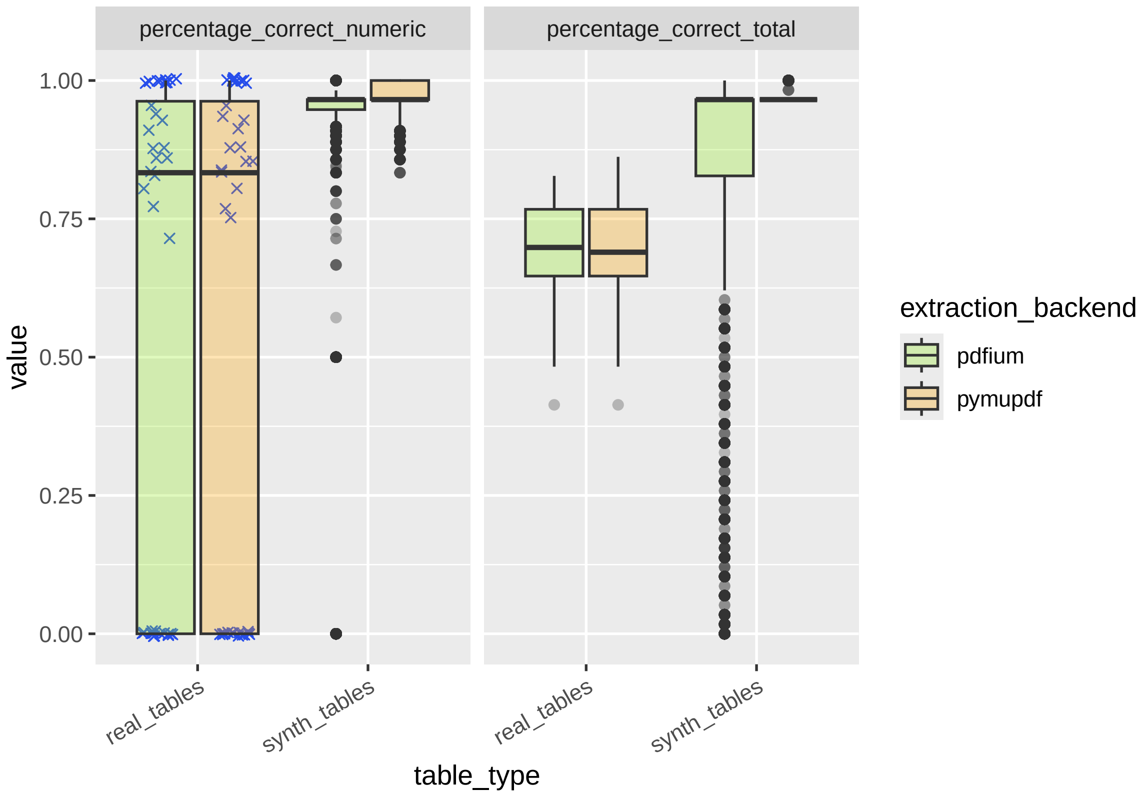


Figure 5.27: Performance overall and on numeric value extraction with regular expressions. Showing single scores for *percentage correct numeric* on real tables to explain wide boxes.

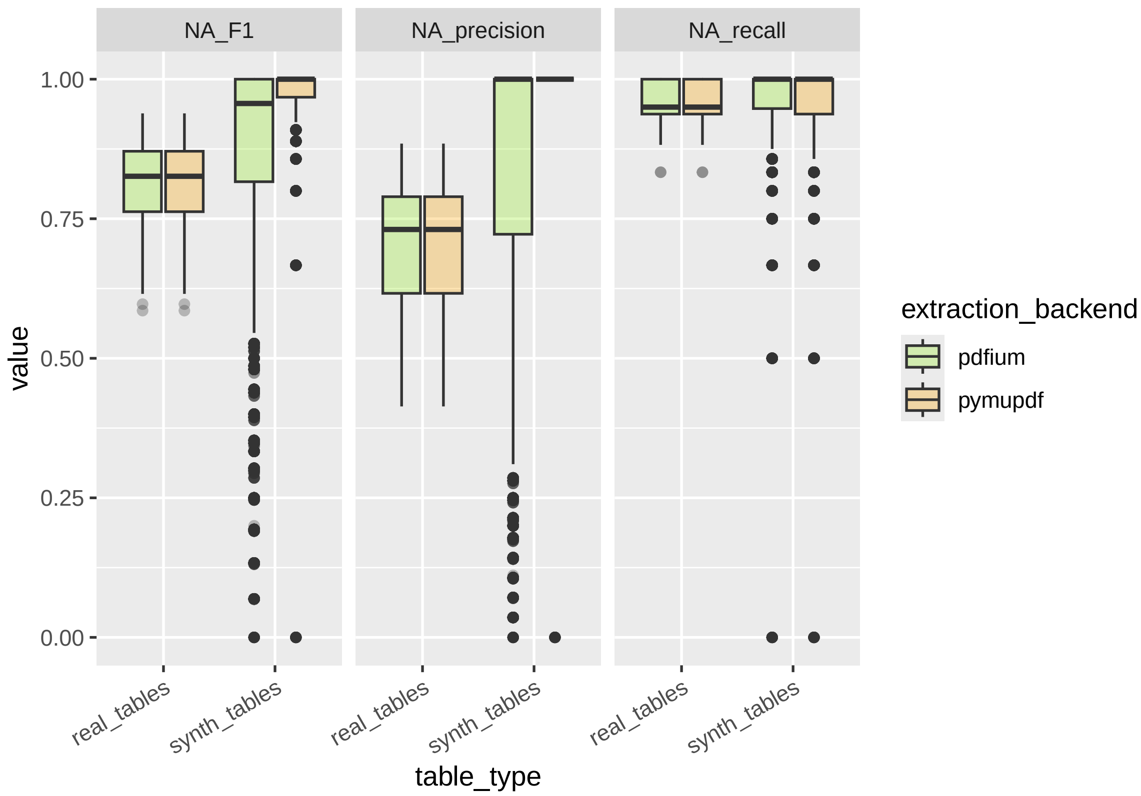


Figure 5.28: Performance on classification for missing values with regular expressions

But one has to keep in mind that the number of data points on the aggregated values for the test set of the real dataset is only 18. So these findings are not strongly supporting any interpretation at all. Furthermore, the found effects are not very large - most below 5 %. Only the hypothesis for a positive influence of a missing of a value for the binomial prediction gets solid support with a mean absolute SHAP value of over 20 %. To get reliable results more tables have to be included which would require additional manual encoding.

Synthetic dataset Interpreting the visual results for the SHAP analysis on the synthetic dataset brought some interesting insides into the question under which condition the two PDF extraction libraries perform differently. These results can be treated as reliable since the model has been trained with 50_000 rows and the SHAP values have been calculated on 2_000 rows each.

Very interestingly the number of columns is having an opposite effect for the two libraries (see Figure 5.29 A). Besides that often only pdfium struggled with some of the table characteristics while pymupdf is not influenced by them (for an example with header_span see Figure 5.29 B).

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.

Since there has no synthetic data created where also the Passiva table is present the result found with the real dataset can't be investigated further. Also the question if visual separation is having an effect was not studied, even though, creating such additional synthetic data would be very easy with the current generation process and could be done in future work. It would be interesting if the visual separation is cause for the maleous text extractions of pdfium as well.

X1	X2	X3	X4	X5	X6	X7
predictor	F1	F1	% numeric correct	% numeric correct	binomial	binomial
predictor	Hypothesis	Result	Hypothesis	Result	Hypothesis	Result
extraction_backend	neutral	pymupdf better	neutral	pymupdf better	neutral	pymupdf b
n_columns	4 is worse	positive	neutral	positive	neutral	positive
sum_same_line	neutral	neutral	negative	negative*	negative	neutral
header_span	neutral	negative*	neutral	negative*	neutral	negative*
thin	negative	NA	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	negative	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

5.2.2 Extraction with LLMs

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

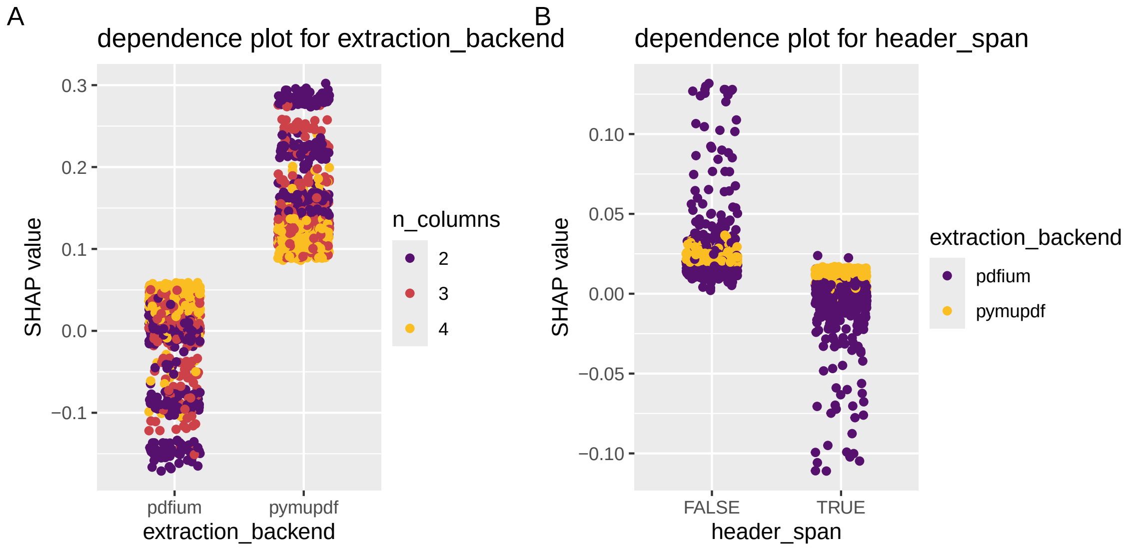


Figure 5.29: Showing the influence of the extraxtion library on the numeric text extraction task with synthetic data

5.2.2.1 Real tables only

For the table extraction task 31 open source models have been benchmarked[29]. The results are presented in Figure A.14, A.15 and A.16).

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. Only 5 models perform better without any guidance[30] (see Table 5.16). 10 models achieved an performance better as 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. Just 4 don't achieve a better value even with three or five realistic examples (see Table 5.17). 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.

With one and three examples the performance within one model family is positive correlated with the number of parameters the models have. Once the 4B parameters are passed the improvements get less and less getting closer to a perfect performance but never reaching it on all documents. Table 5.18 shows the mean performance for the best model-method approach for each model family. Most of the top performing model-method combinations rely on the maximum number of examples provided. Only the Llama-3 and Falcon3 model perform best with three examples[31].

Based on a small sample of 8 documents by the *Amt für Statistik Berlin-Brandenburg* it seems that there is support for the hypothesis, that providing Aktiva tables from the same company in in-context learning, is improving the results. This is especially noticeable for models with very few parameters and when providing only a single example. This seems intuitive, since there the potential for possibilities is much bigger. Figure A.17 shows that on this limited sample

[29] 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.

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

[31] Phi4 also perfroms best with three examples. But this is the maximum it can process due to a limited context length.

Table 5.16: Comparing table extraction performance with real 'Aktiva' dataset for models that perform well without or with little context learning

model	mean_total_zero_shot	mean_total_static_example
Llama4Maverick17B128EInstructFP8	0.816	0.844
Qwen2.532BInstruct	0.76	0.875
Qwen3235BA22BInstruct2507	{0.848}	{0.88}
Qwen3235BA22BInstruct2507FP8	0.825	0.873
phi4	0.807	0.75
Llama3.170BInstruct	NA	0.773
MistralSmall3.124BInstruct2503	NA	0.855
Qwen2.572BInstruct	NA	0.838
Qwen330BA3BInstruct2507	NA	0.812
gemma327bit	NA	0.785

Table 5.17: Comparing table extraction performance with real 'Aktiva' dataset for models that worse than the regex baselin with 3 or 5 examples for incontext learning

model	method	mean_total
Llama4Maverick17B128EInstructFP8	5_random_examples	0.041
Qwen2.50.5BInstruct	3_random_examples	0.585
Qwen30.6B	3_random_examples	0.608
gemma34bit	top_5_rag_examples_out_of_sample	{0.682}

Table 5.18: Comparing best mean table extraction performance with real 'Aktiva' dataset for each model family

model_family	model	method_family	n_examples	mean_total
Qwen 3	Qwen3235BA22BInstruct2507FP8	top_n_rag_examples	5	0.961
Llama-4	Llama4Scout17B16EInstruct	top_n_rag_examples	5	0.931
mistralai	MistralLargeInstruct2411	top_n_rag_examples	5	0.929
Llama-3	Llama3.170BInstruct	n_random_examples	3	0.911
Qwen 2.5	Qwen2.514BInstruct	n_random_examples	5	0.908
microsoft	phi4	n_random_examples	3	0.893
tiiuae	Falcon310BInstruct	top_n_rag_examples	3	0.862
google	gemma327bit	n_random_examples	5	0.821

Table 5.19: Comparing best mean table extraction performance with real 'Aktiva' dataset for each model family for models with less than 17B parameters

model_family	model	method_family	n_examples	mean_total
Qwen 3	Qwen314B	n_random_examples	3	0.912
Qwen 2.5	Qwen2.514BInstruct	n_random_examples	5	0.908
mistralai	Ministral8BInstruct2410	n_random_examples	5	0.896
microsoft	phi4	n_random_examples	3	0.893
tiiuae	Falcon310BInstruct	top_n_rag_examples	3	0.862
Llama-3	Llama3.18BInstruct	n_random_examples	5	0.849
google	gemma312bit	n_random_examples	5	0.797

- the improvement is bigger for Qwen 3 than for Qwen 2.5
- Googles gemma 27b and GPT 4.1 mini could overcome an unnoticed issue with the extraction with just one example.
- the effect of being overwhelmed by a too rich context with LLamas Maverick model could get reduced a bit.

To examine the question, if the reported confidence score of the responses can be used, to flag the predicted values as potentially wrong. Again, Figure 5.30 shows, that Qwen 3 reports very high confidence values no matter if the results are correct or not. With the Mistral model we find a wider range of confidences given and for wrong results lower confidence is reported.

Figure 5.31 shows, that the chance to make an mistake by believing the prediction is rising with lower confidence. The chance to make a mistake is higher for predictions of numeric values than for believing a value is not present in the table. The chance to make such a mistake is higher using the confidence reported by Qwen 3.

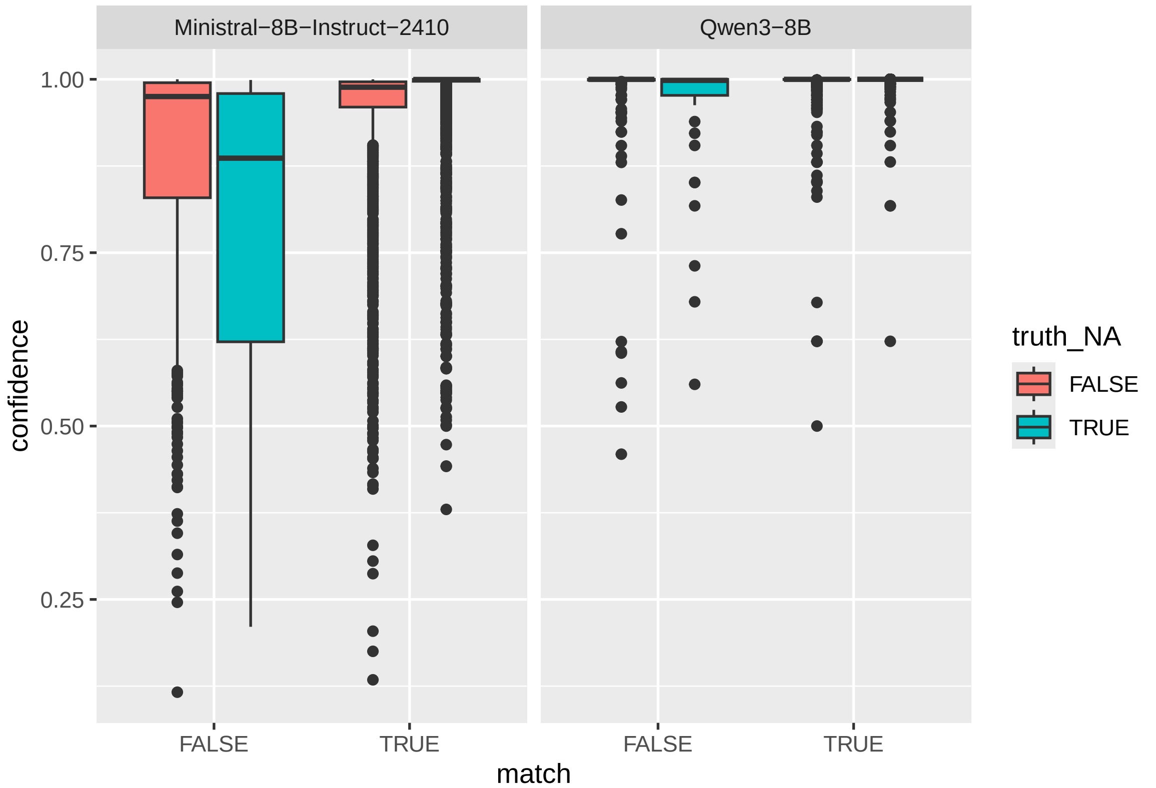


Figure 5.30: Comparing the reported confidence scores for the table extraction task on real dataset for the Mistral and Qwen 3 with 8B parameters.

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.

Even though the samples size of Aktiva tables did not increase, the available training, test and SHAP sample size is much larger, because the experiment has been repeated with different models and methods. Thus, the interpretations based on the visual evaluation (see Figure A.9)) are more reliable for model and method specific predictors. Since there is one Aktiva example for every company files were found for they might even be generalizable for this population. But one has to keep in mind that there have been more Aktiva tables for *Amt Stat BBB* which might nudge the results a bit.

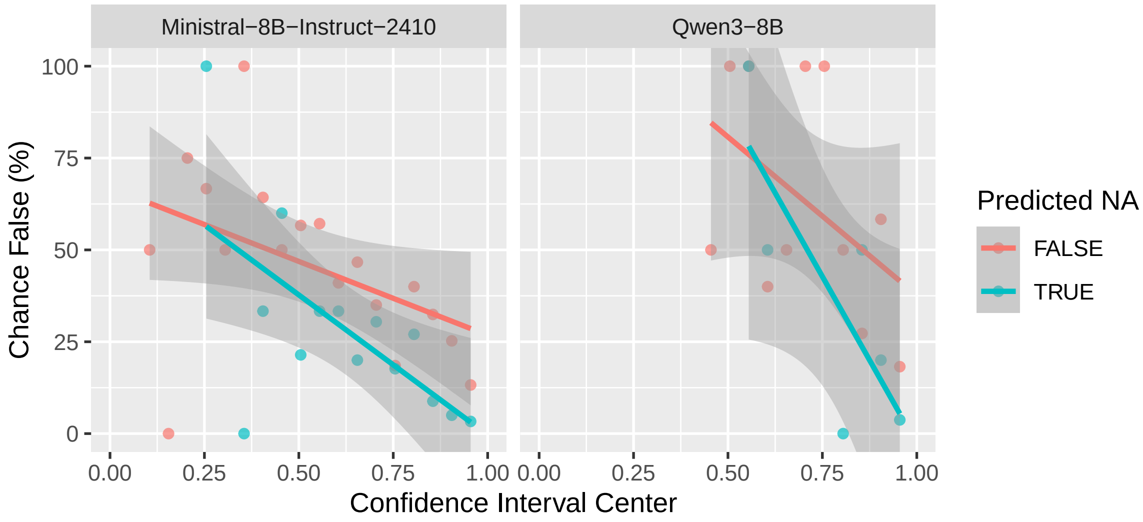


Figure 5.31: Estimating the relative frequency to find a wrong extraction result over different confidence intervals

The results assign much more influence on model and method specific attributes than on the table specific attributes. The importance of the table attributes are as low as found with the regular expresion approach. Only for the binomial prediction we find the predictor *missing* to get assigned more importance than to all model and method specific attributes. Same is true for the *label* that is having the highest influence on the reported confidence. Nevertheless, in the case of the binomial prediction there is half of the predictors *missing* and *label* importance shifted to model and method specific predictors.

Again, multiple hypotheses don't get supported by the visual results. The surprising results are:

1. In general more examples are helpful except for Llamas Maverick model that performs poorly with five examples. But this effect is only noticeable with the aggregated metrics nor for the case wise binomial evaluation.
2. The number of columns has a negative effect on the performance but no effect on the reported confidence.
3. There was no negative effect found if the *Passiva* table is on the same page as the Aktiva table.
4. Larger models start to report less confidence again. This is not unexpected for the Mistral model but was surprising for the largest Qwen 3 model. (Discussion: New Generation? Aktive paramaters count? Irrelevant because not well distinguishing?)
5. It not only influences the the performance to extract the correct numeric value from a row where there are additional sums present but also the F1 score.

Two interesting details found while inspecting the dependence plots for the metric *percentage_numeric_correct* are (see Figure 5.32 A) that the bad performance of LLamas Maverick with five examples is easily spottable and that the negative effect of *T_in_year* might be caused by an interaction with *vis_separated_rows* completely (see Figure 5.32 B). To investigate the second finding one would need tables where the uni is present in the year column and having no visual separation of the rows at the same time. Synthetic data potentiall could help to answer such questions.

GPT Even though a lot of documents to process at RHvB (Rechnungshof von Berlin) will not be public and thus must not be processed on public cloud infrastructure, the performance of models like OpenAI's GPT or Google's Gemini are interesting benchmark references within this thesis and for comparing these findings with other papers results. Therefore for this thesis the public available versions of annual reports have been

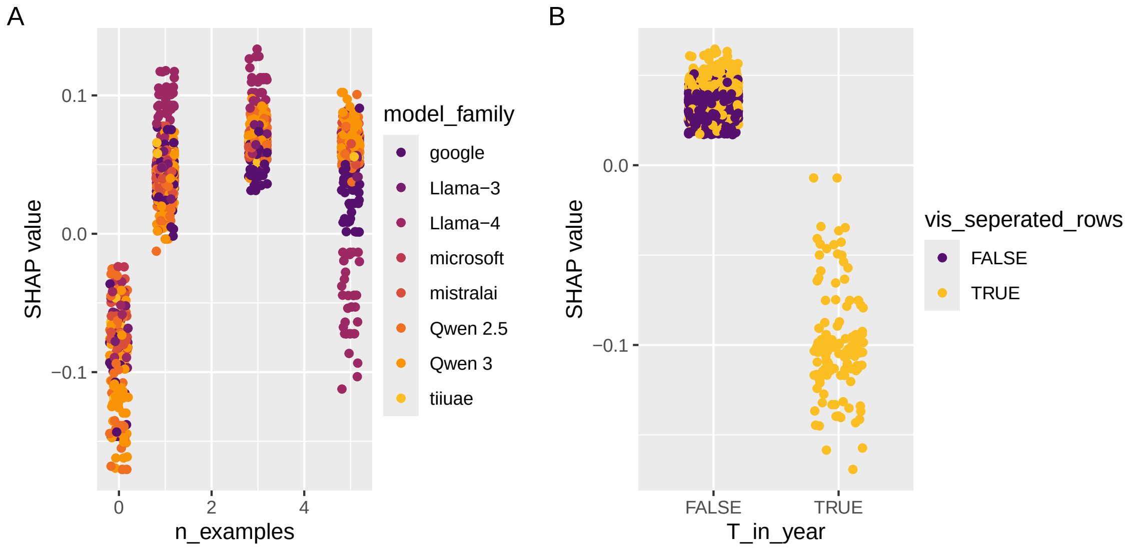


Figure 5.32: Showing the influence of many examples on Llama 4 Maverick (A) and interaction between *T in year* and *vis separated rows* (B)

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.

As a reference to compare the performance of OpenAI's models with the results of four Qwen 3 models are shown as well (see Figure 5.33, 5.34 and 5.35). Surprisingly gpt-5-mini is almost performing as good as the top Qwen 3 model and gpt-4.1. But besides gpt-4.1-nano and Qwen3-0.6B all models perform pretty well with the random or top-rag example methods. The ranking for the best model-method combination can be found in Table @ref(tab.table-extraction-llm-performance-total-gpt-ranking). Since gpt-4.1 costs five times more than gpt-4.1-mini (see Table @ref{tab:costs-azure}) it seems reasonable to prefer the smaller model for this specific task.

Costs for gpt-5-mini not shown in Azure yet. :(

The author was not able to get OpenAI's models to stick to the provided json schema strictly. Passing the ebnf (extended Backus–Naur form) grammar did not work at all. This means that with gpt-4.1-nano there have been 12 predictions that have been completely empty. Overall there have been 22.4 % of the responses of OpenAI's models that were compatible with the schema but had a wrong number of rows predicted (see Figure @red(fig:table-extraction-llm-prediction-count-gpt)).

Using gpt-5-nano and gpt-5-chat for the table extraction task was not working. With gpt-5-nano the answers were not respecting the provided grammar. Running gpt-5-chat resulted in the error informing that a *json_schema* can't be used with this model. With gpt-5-mini the very approach worked flawless. Running gpt-oss-20b with the vllm offline inference framework was possible and the new harmony output format could be processed after minor code changes for most approaches[32]. With a gpt-oss-120b instance hosted on Azure the guided decoding worked flawless.

[32]With the static example approach there have been 24 files where the resonse could not get parsed into valid json. With the other approaches there are one to four unparsable responses.

Table 5.20: Comparing best mean table extraction performance with synthetic 'Aktiva' dataset for each model family

model_family	model	method_family	n_examples	mean_total
Qwen 3	Qwen3235BA22BInstruct2507	n_random_examples	5	{0.99}
Qwen 2.5	Qwen2.572BInstruct	top_n_rag_examples	5	{0.986}
mistralai	MistralLargeInstruct2411	top_n_rag_examples	1	{0.985}
Llama-4	Llama4Maverick17B128EInstructFP8	top_n_rag_examples	1	{0.975}
google	gemma327bit	top_n_rag_examples	5	0.918
Llama-3	Llama3.18BInstruct	top_n_rag_examples	3	0.907

model	method	mean correct total
Qwen3-235B-A22B-Instruct-2507	3_random_examples	0.95
gpt-4.1	3_random_examples	0.94
gpt-5-mini	top_3_rag_examples_out_of_sample	0.93
Qwen3-30B-A3B-Instruct-2507	top_3_rag_examples_out_of_sample	0.90
gpt-4.1-mini	top_3_rag_examples_out_of_sample	0.90
Qwen3-8B	3_random_examples	0.89
gpt-oss-120b	top_3_rag_examples_out_of_sample	0.88
gpt-oss-20b	1_random_examples	0.85
Qwen3-0.6B	top_3_rag_examples_out_of_sample	0.66
gpt-4.1-nano	3_random_examples	0.26

Model	Cost	Cost_all_tasks	Currency
gpt 4.1 Inp glbl Tokens	3.53	7.02	EUR
gpt 4.1 Outp glbl Tokens	2.71	3.44	EUR
gpt 4.1 mini Inp glbl Tokens	1.23	1.23	EUR
gpt 4.1 mini Outp glbl Tokens	0.71	0.71	EUR
gpt 4.1 nano Inp glbl Tokens	0.31	0.31	EUR
gpt 4.1 nano Outp glbl Tokens	0.15	0.15	EUR
gpt-oss-120B Outp glbl Tokens	0.85	0.85	EUR
gpt-oss-120B Inp glbl Tokens	0.42	0.42	EUR

5.2.2.2 Synthetic tables only

Table 5.20 shows that for 4 from 2 model families there is at least one model-method combination that performed better than the regex baseline. For the synthetic table extraction task the baseline is 0.9691401.

Only 25 from 2 model-method combinations performed better than this baseline. There has been no model that performed better than this baseline with the zero or static example approach.

span argument was not implemented correct in html tables and md :/

already just using 10 % of documents generated; and then 10 % of that sum of all experiment results (factor 14) with random forest?

A.19

Hypotheses HTML and Markdown better but expected interaction effects mostly not found - except: - columns help pdf - thinning least bad for pdf - pdf worst with numbers that have currency units (short numbers, maybe no 1000er delimiter) - enumeration positive for pdf (and interaction with log10 mult)

line breaks are no problem

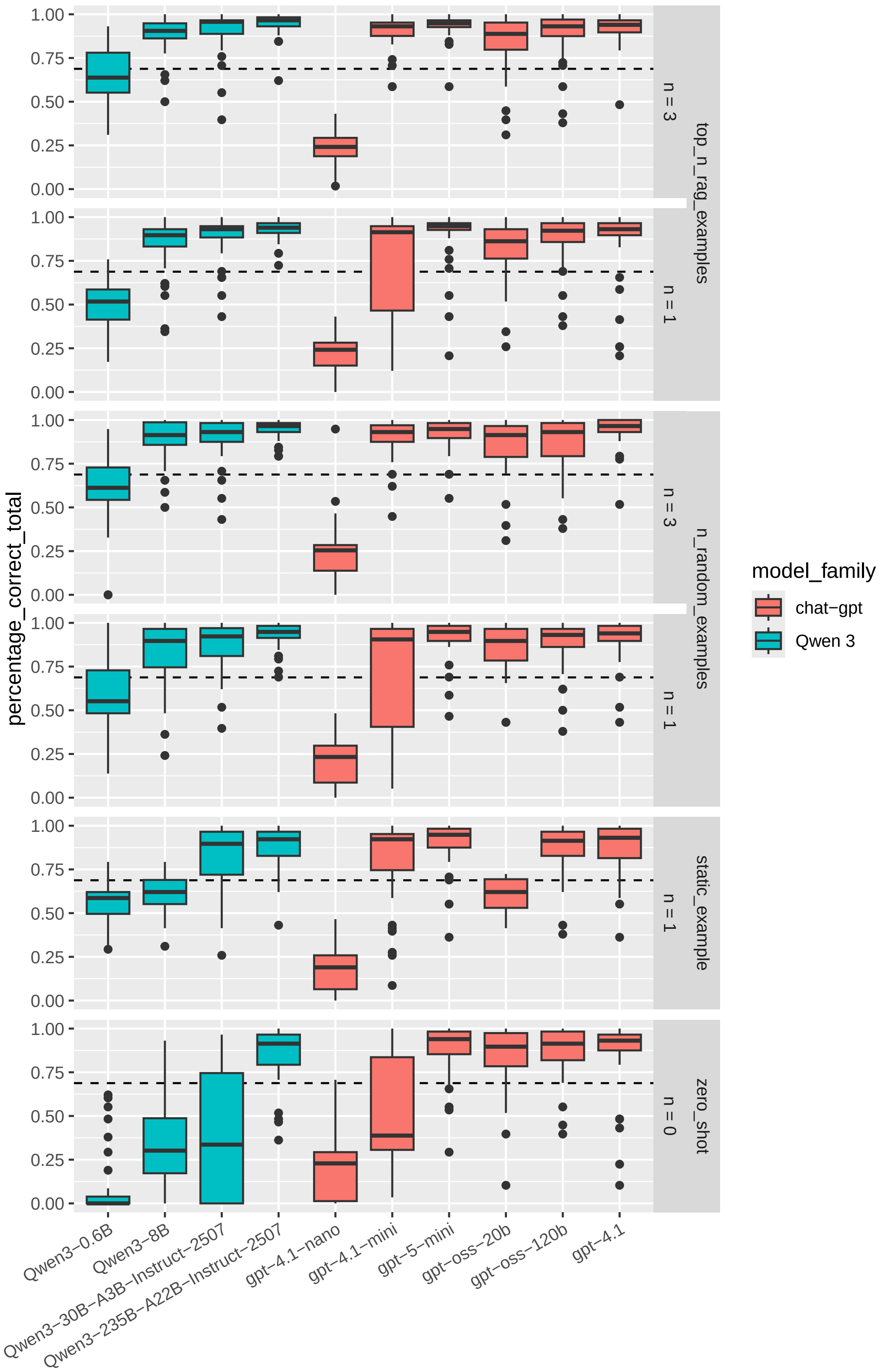


Figure 5.33: Comparing the percentage of correct predictions overall for OpenAi's LLMs with some Qwen 3 models

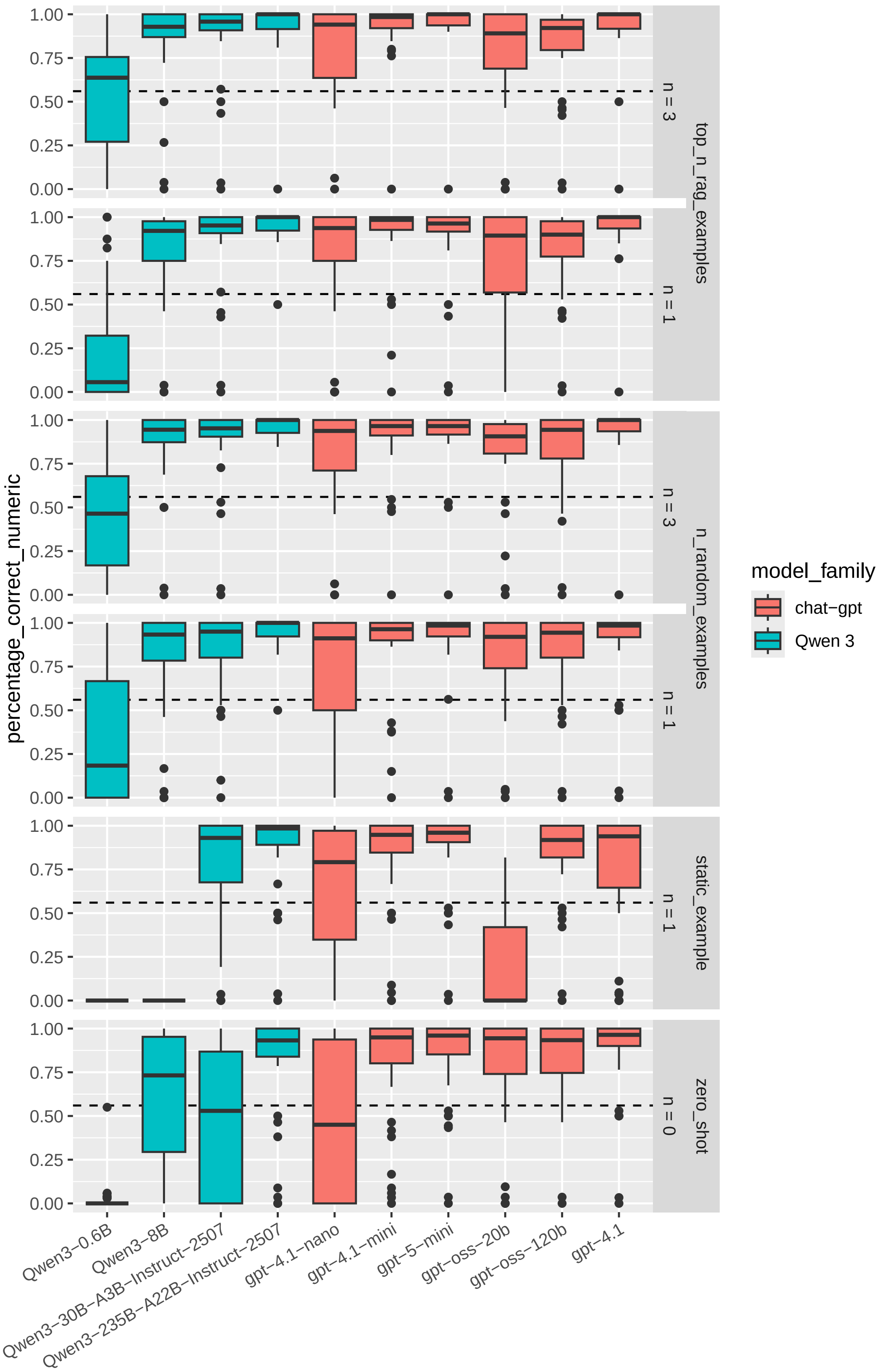


Figure 5.34: Comparing the percentage of correct numeric predictions for OpenAi's LLMs with some Qwen 3 models

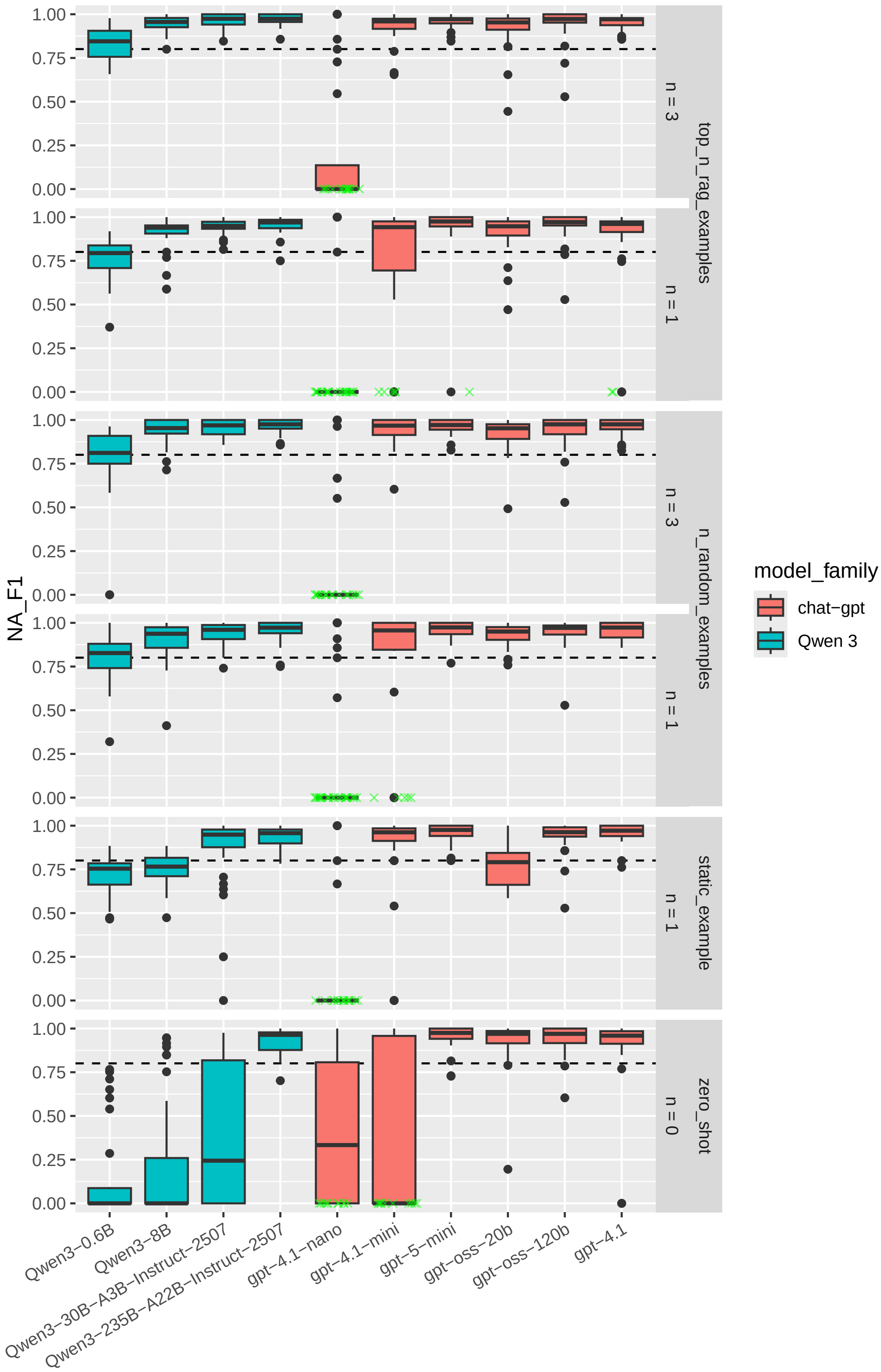


Figure 5.35: Comparing the F1 score for predicting the missingness of a value for OpenAi's LLMs with some Qwen 3 models. The green crosses indicate results where a model has predicted only numeric values even though there have been missing values.

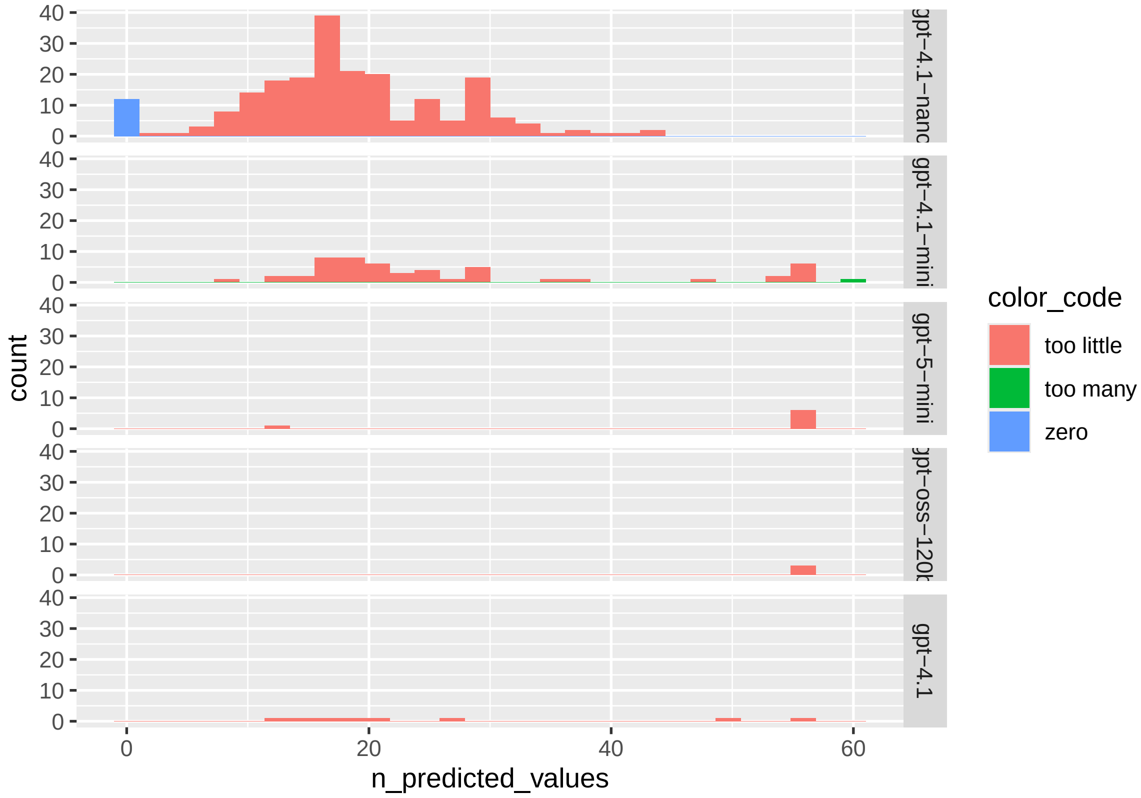


Figure 5.36: Showing the number of predictions OpenAI's models made.

Table 5.21: Comparing best mean 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
Qwen 3	Qwen38B	top_n_rag_examples	5	0.963
mistralai	Ministral8BInstruct2410	top_n_rag_examples	5	0.939
Qwen 2.5	Qwen2.57BInstruct	top_n_rag_examples	3	0.938
Llama-3	Llama3.18BInstruct	top_n_rag_examples	3	0.907
google	gemma312bit	top_n_rag_examples	5	0.859

Table 5.22: Comparing extraction performance for real Aktiva extraction task with synthetic and real examples for incontext learning with a zero shot approach

model	method	mean_synth	mean_real	mean_zero_shot
Llama4Scout17B16EInstruct	top_5_rag_examples_out_of_sample	{0.887}	{0.925}	0.387
MistralLargeInstruct2411	5_random_examples	0.873	0.901	{0.691}
Qwen38B	5_random_examples	0.797	0.898	0.359
Llama3.18BInstruct	top_3_rag_examples_out_of_sample	0.764	0.805	0.5
gemma327bit	5_random_examples	0.732	0.821	0.255
Ministral8BInstruct2410	3_random_examples	0.732	0.882	0.541
gemma312bit	top_1_rag_examples_out_of_sample	0.607	0.713	0.582

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 Extract from real tables with synthetic content

Table 5.22 shows that using real examples for in-context- learning is better than using the created synthetic data. Nevertheless, it is improving the overall performance for the table extraction task by almost 20 % for all models but Google's gemma-3-12b-it. However, the performance difference with and without in-context learning is the smallest for the gemma-3-12b-it model as well. The spread of the performance is bigger using synthetic in-context learning data for all models but Llama 3 8B Instruct.

Any pattern?

Table 5.23 shows, that synthetic data can be used for in-context learning in a task where the currency units given for a table or specific columns[33]. Except for Llama 3.1 all models achieved much better numeric extraction results for tables where currencies are given for all columns. The model also achieved better for cases where there was only a single column with units, even though there have been no examples with units only for one column in the synthetic data. On the other hand, being prompted to respect currency units decreased the performance for tables where no units are given for all models but Llama 4 Schout. This decrease was highest for Qwen 3 and also higher than 10 % for Mistral Large.

A.11, A.12 and A.13

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

Confidence with both tasks (respect or ignore units) the same.

```r
confidence_vs_truth %>% group_by(model, method, method_family, respect_units,
↪  predicted_NA) %>% reframe(mean_conf = mean(confidence)) %>% pivot_wider(names_from =
↪  respect_units, values_from = mean_conf, names_prefix = "mean_conf_units_") %>% #
↪  mutate(delta = units_FALSE - units_TRUE) %>% select(-c(units_FALSE, units_TRUE)) %>%
↪  pivot_wider(names_from = n_col_T_EUR, values_from = delta, names_prefix =
↪  "n_cols_T_EUR_") %>% group_by(model) %>% summarise_if(is.numeric, mean) %>%
  mutate(across(is.numeric, ~round(., 2))) %>%
```

[33] Synthetic data is used here because the characterization, which real *Aktiva* table has units in which column, was created too late.

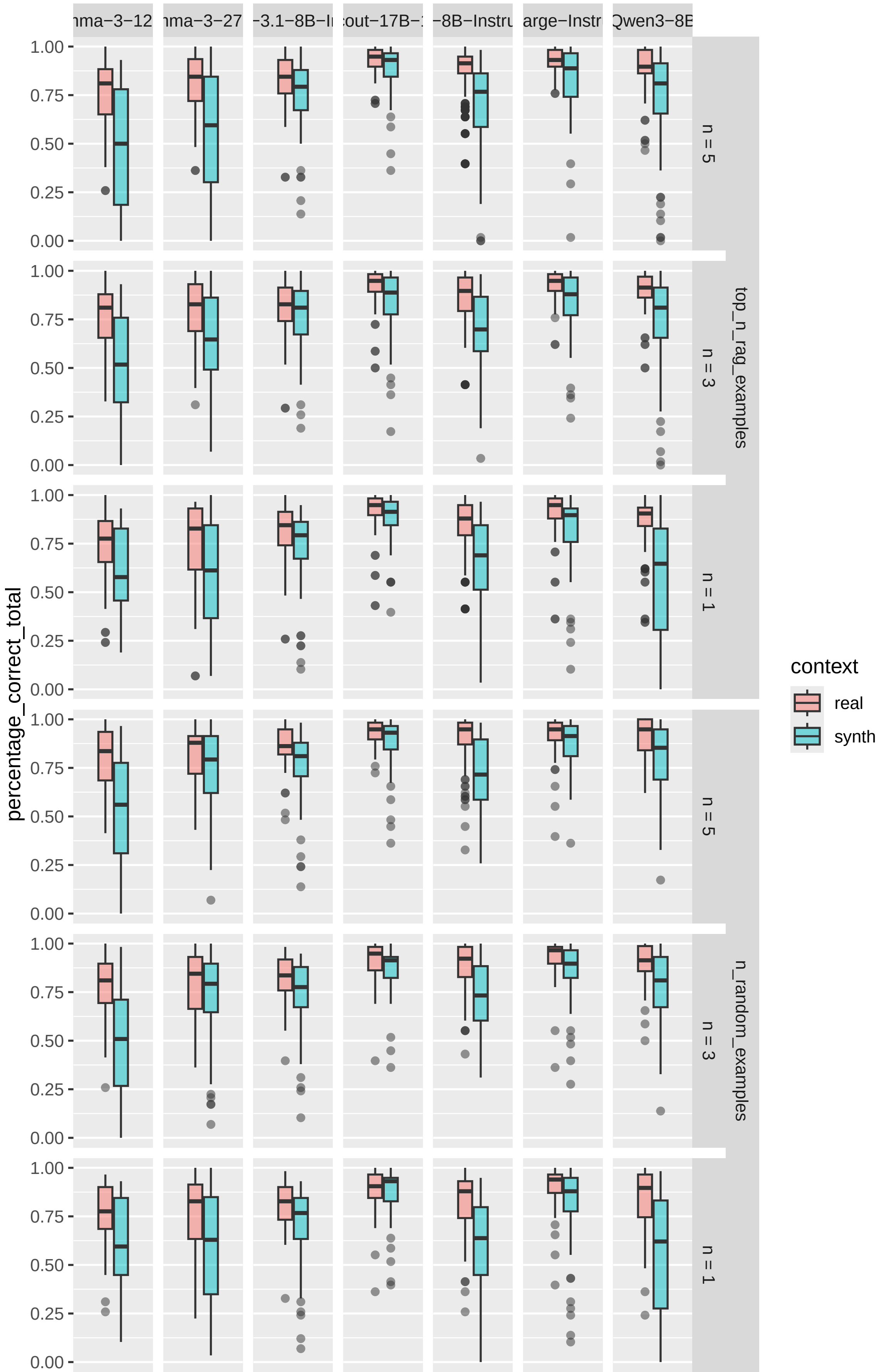


Figure 5.37: Comparing table extraction perfromance for real Aktiva extraction task with synthetic and real examples for in-contect learning

Table 5.23: 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	0.04	0.02	0.02
Llama4Scout17B16EInstruct	{0.01}	0.23	0.87
Ministral8BInstruct2410	0.06	0.19	0.65
MistralLargeInstruct2411	0.14	{0.4}	{0.9}
Qwen38B	0.31	0.1	0.51
gemma312bit	0.03	0.14	0.76
gemma327bit	0	0.3	0.36

Table 5.24: Comparing extraction confidence for real Aktiva extraction task dependent on the promt addition to respect currency units. No difference in confidence appearent.

model	method	method_family	predicted_NA	mean_conf_units_FALSE	mean_c
Ministral8BInstruct2410	3_random_examples	n_random_examples	FALSE	0.86	0.84
Ministral8BInstruct2410	3_random_examples	n_random_examples	TRUE	0.97	0.97
Qwen38B	5_random_examples	n_random_examples	FALSE	0.98	0.98
Qwen38B	5_random_examples	n_random_examples	TRUE	1	1

```
render_table(
  alignment = "lllrr",
  caption = "Comparing extraction confidence for real Aktiva extraction task dependent
  ↪  on the promt addition to respect currency units. No difference in confidence
  ↪  appearent.",
  ref = opts_current$get("label"), dom="t")
```

Risk for false NAs less with synth data for Mistral but greater for numeric values (both).

Hypotheses

5.2.3 Comparison

Most models performe better on synth tables once they have enough in-context examples. (Needing more für random examples thanwith top-n-rad approach). Especially Llama 3 models show wider performance spread even with three examples A.18

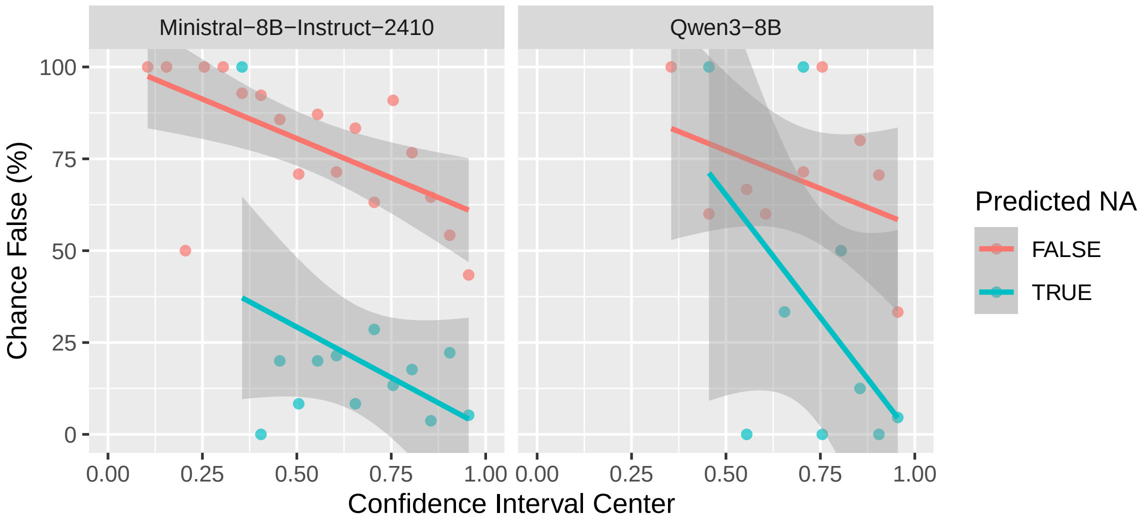


Figure 5.38: Estimating the relative frequency to find a wrong extraction result over different confidence intervals for predictions based on synthetic examples for in-context learning.

Chapter 6

Discussion

6.1 Limitations

6.1.1 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

6.1.1.1 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?

6.1.2 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.

6.2 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)?)

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

6.3 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)
-

6.3.1 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. 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*.

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

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

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

List of Figures

1.1 Overview of companies Berlin holds share at 1

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. 14

5.2 Comparing the performance among different companies. 16

5.3 Histogram of the number of lines in the first 5 pages of the annual reports 18

5.4 Comparing number of found TOC and amount of correct and incorrect predicted page ranges 19

5.5 Comparing number of fount TOC and amount of correct and incorrect predicted page ranges . 20

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. 22

5.7 Showing the minimal distance of the predicted page range to the actual page number overthe logprobs of the models response confidence. 22

5.8 Showing the amount of correct and incorrect predicted page ranges (bars) and the percentage of correct predictions (black line). 23

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. 24

5.10 Showing F1 score performance over normalized runtime for binary classification for Mistral-8B-Instruct-2410. 27

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. 28

5.12 Showing the confidence score for the Aktiva classification task grouped by table type and correctness for Mistral-8B-Instruct-2410. 30

5.13 Showing the confidence score for the Aktiva classification task grouped by table type and correctness for Qwen-2.5-32B-Instruct. 31

5.14 Showing the precsion-recall-curve for the best performing model. 32

5.15 Comparing the reported confidence scores for the page identification task for the Mistral and Qwen 3 with 8B parameters. 33

5.16 Estimating the relative frequency to find a wrong classification over different confidence intervals 34

5.17 Comparing F1 score micro averaged for the minority classes for two models over their normalized runtime. 36

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. 37

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. 38

5.20 Showing the precsion-recall-curve for Ministral-8B-Instruct-2410. 40

5.21 Comparing the reported confidence scores for the multi-class page identification task for the Mistral and Qwen 3 with 8B parameters. 41

5.22 Estimating the relative frequency to find a wrong classification over different confidence intervals for the multi-class classification task. 41

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. 42

5.24 Comparing the top n recall on training and test dataset among the random forest with two and four predictors. 43

5.25 Beeswarm plot of SHAP importance values for the four predictors of the second random forest classifier. 44

5.26 Showing the precsion-recall-curve for the random forest with four predictors. 45

5.27 Performance overall and on numeric value extraction with regular expressions. Showing single scores for *percentage correct numeric* on real tables to explain wide boxes. 49

5.28 Performance on classification for missing values with regular expressions 50

5.29 Showing the influence of the extraxtion library on the numeric text extraction task with synthetic data 52

5.30 Comparing the reported confidence scores for the table extraction task on real dataset for the Mistral and Qwen 3 with 8B parameters. 54

5.31 Estimating the relative frequency to find a wrong extraction result over different confidence intervals 55

5.32 Showing the influence of many examples on Llama 4 Maverick (A) and interaction between *T in year* and *vis separated rows* (B) 56

5.33 Comparing the percentage of correct predictions overall for OpenAi's LLMs with some Qwen 3 models 58

5.34 Comparing the percentage of correct numeric predictions for OpenAi's LLMs with some Qwen 3 models 59

5.35 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. 60

5.36 Showing the number of predictions OpenAI's models made. 61

5.37 Comparing table extraction perfromance for real Aktiva extraction task with synthetic and real examples for in-contect learning 63

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. 65

A.1 Comparing page identification metrics for different regular expressions for each classification task by type of the target table. 100

A.2 Mean absolute SHAP values and beeswarm plots for real table extraction with regular expression approach 101

A.3 Comparing number of fount TOC and amount of correct and incorrect predicted page ranges . 102

A.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. . . . 102

A.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. . . . 103

A.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. . 104

A.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. . . . 104

A.8 Showing the precsion-recall-curve for Llama-4-Scout-17B-16E-Instruct. . . . 105

A.9 Mean absolute SHAP values and beeswarm plots for real table extraction with LLMs 106

A.10 Mean absolute SHAP values and beeswarm plots for synth table extraction with regular expression approach 107

A.11 Comparing the effect on overall perfrormance if currency units should be respected on all predictions and specifically on predictions where all or just some columns have units. 108

A.12 Comparing the effect on numeric perfrormance if currency units should be respected on all predictions and specifically on predictions where all or just some columns have units. 109

A.13 Comparing the effect on NA F1 score if currency units should be respected on all predictions and specifically on predictions where all or just some columns have units. 110

A.14 Percentage of correct extracted or as missing categorized values for table extraction task on real Aktiva tables 111

A.15 Percentage of correct extracted numeric values for table extraction task on real Aktiva tables . 112

A.16 F1 score for the missing classification if a value is missing for table extraction task on real Aktiva tables 112

A.17 Comparing the overall extraction performance depending on the condition if examples from the same company can be used (only for Amt für Statistik Berlin-Brandenburg). 113

A.18 Comparing the table extraction performance among real and synthetic Aktiva tables 113

A.19 Percentage of correct extracted or as missing categorized values for table extraction task on synthetic Aktiva tables 114

A.20 Percentage of correct extracted numeric values for table extraction task on synthetic Aktiva tables 114

A.21 F1 score for the missing classification if a value is missing for table extraction task on synthetic Aktiva tables 115

A.22 Example balance sheet pagefom Californias Annual Comprehensive Financial Report 2023 . . 119

A.23 Flowchart of the extraction framework 120

A.24 Flowchart of the extraction framework of Li et al. (2023) 121

List of Tables

5.1 Showing the number of documents with multiple target tables per type and the number of target tables that span two pages. 15

5.2 Comparing page identification metrics for different regular expressions for each classification task by type of the target table. 15

5.3 Comparing the number and percentage of correct identified page ranges among the approaches. 18

5.4 Comparing the number and percentage end pages prediction for Aktiva and Passiva that are equal. 19

5.5 Comparing the number and percentage of correct identified page ranges among the approaches. 21

5.6 Comparing GPU time for page range prediction and table of contents extraction. Time in seconds per text processed. 21

5.7 Comparing the mean and median page range sizes. 21

5.8 Overview of benchmarked LLMs for the classification tasks. 25

5.9 Overview of benchmarked LLMs for the binary classification tasks. Limiting the number of examples provided for the few shot approach to 3. 26

5.10 Overview of benchmarked LLMs for the multiclass classification tasks. Limiting the number of examples provided for the few shot approach to 3. 34

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. 35

5.12 Comparing page identification perfromance among all four approaches. 46

5.13 Comparing the top k recall for the termfrequency and LLM approaches. 46

5.14 Comparing page identification efficiency among all four approaches. 48

5.16 Comparing table extraction performance with real 'Aktiva' dataset for models that perform well without or with little context learning 53

5.17 Comparing table extraction performance with real 'Aktiva' dataset for models that worse than the regex baselin with 3 or 5 examples for incontext learning 53

5.18 Comparing best mean table extraction performance with real 'Aktiva' dataset for each model family 53

5.19 Comparing best mean table extraction performance with real 'Aktiva' dataset for each model family for models with less than 17B parameters 53

5.20 Comparing best mean table extraction performance with synthetic 'Aktiva' dataset for each model family 57

5.21 Comparing best mean table extraction performance with synthetic 'Aktiva' dataset for each model family for models with less than 17B parameters 61

5.22 Comparing extraction performance for real Aktiva extraction task with synthetic and real examples for incontext learning with a zero shot approach 62

5.23 Comparing extraction performance for real Aktiva extraction task dependent on the promt addition to respect currency units 64

5.24 Comparing extraction confidence for real Aktiva extraction task dependent on the promt addition to respect currency units. No difference in confidence appearent. 64

A.1 Comparing extraction time (in seconds) for different Python package 82

A.2 Comparing time (in seconds) for processing ten asset tables using different libraries and approaches 86

A.3 Comparing the actual number of provided examples depending on the classification type, example selection strategy and chosen parameter n_examples. 99

Glossary

ACFR Annual Comprehensive Financial Report
AUC area under the curve
GuV Gewinn- und Verlustrechnung
HGB Handelsgesetzbuch
LLM large language model
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 less computationally demanding tasks below. It is a lightweight laptop device. Its performance cores support hyperthreading and have a clock range between 2.1 and 4.7 GHz. However, due to the flat design, there is little active cooling. Thus, thermal throttling starts rather quickly. It is therefore a reasonable assumption that most locally benchmarked tasks are running at 2.1 GHz. Despite this handicap, it has a sufficiently large RAM of 32 GB and 3 GB 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

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

A.2 Benchmarks

A.2.1 Text extraction

A basic requirement for all succeeding tasks is, that the text gets extracted from the PDF files. As written in doclings technical report (Auer et al., 2024) the available open source libraries differ in their speed and restrictiveness of licensing. Since there are no benchmark results this report multiple libraries have been tested here.

The benchmark ran on the local machine described in section A.1. There have been 5256 pages to extract the text from.

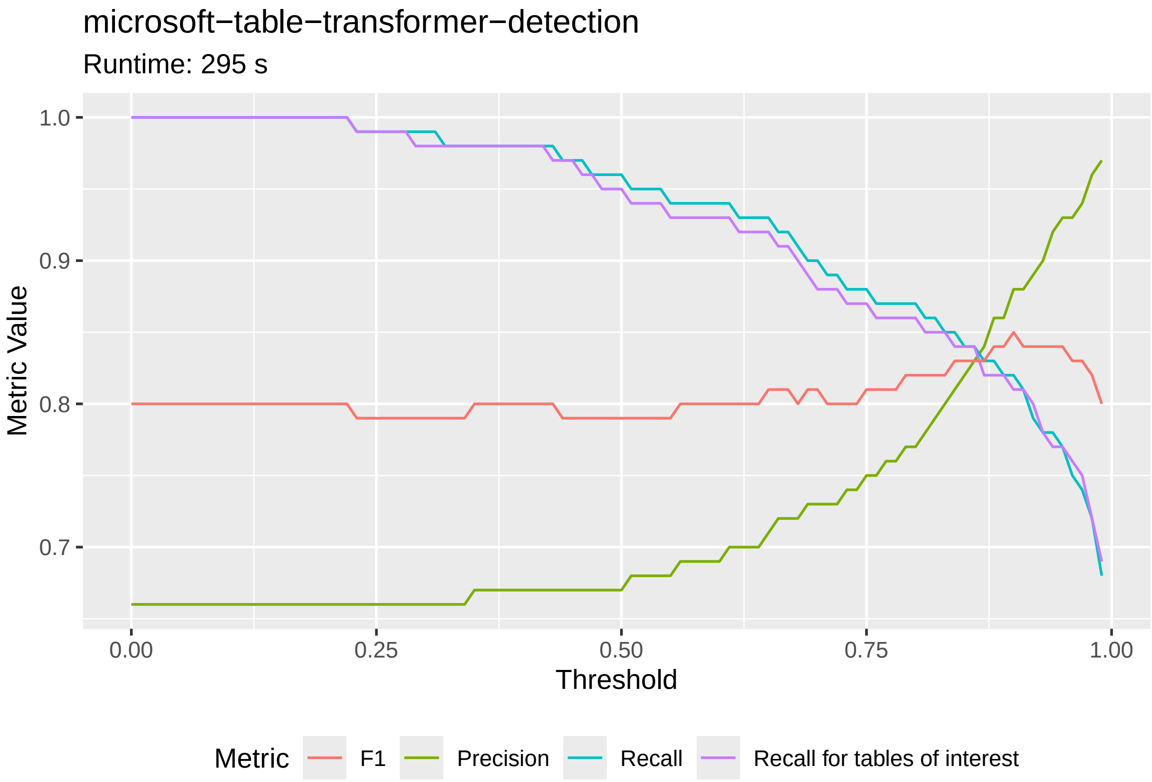
The result of docling-parse is not formated as markdown yet but also just plain text.

For implementation in a system where the text has to get extracted live or frequently the speed of the library might be paramount. But in special cases it can be important to invest more computational power into text extraction if this assures extraction according a more complicated document layout. E.g. some of the tables have been parsed by pdfium in such a manner that first all row descriptors have been extracted (first row) and thereafter all numeric columns (rowwise) ADD REFERENCE / EXAMPLE.

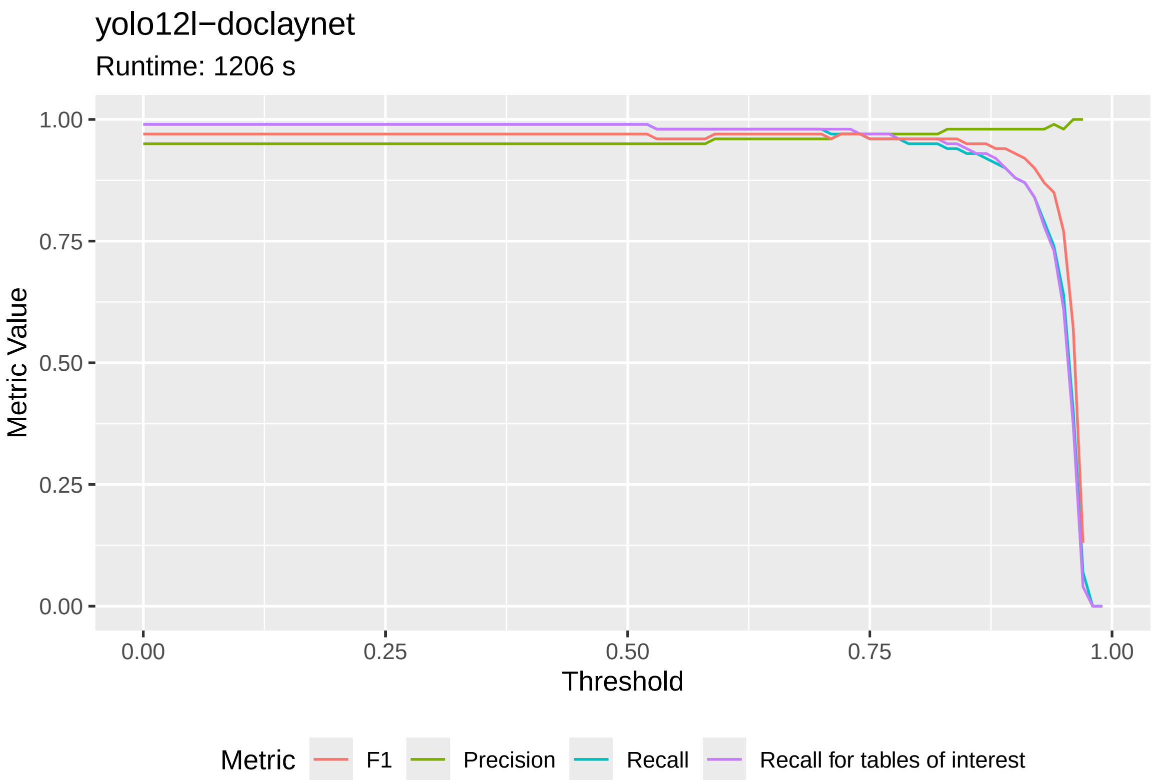
A.2.2 Table detection

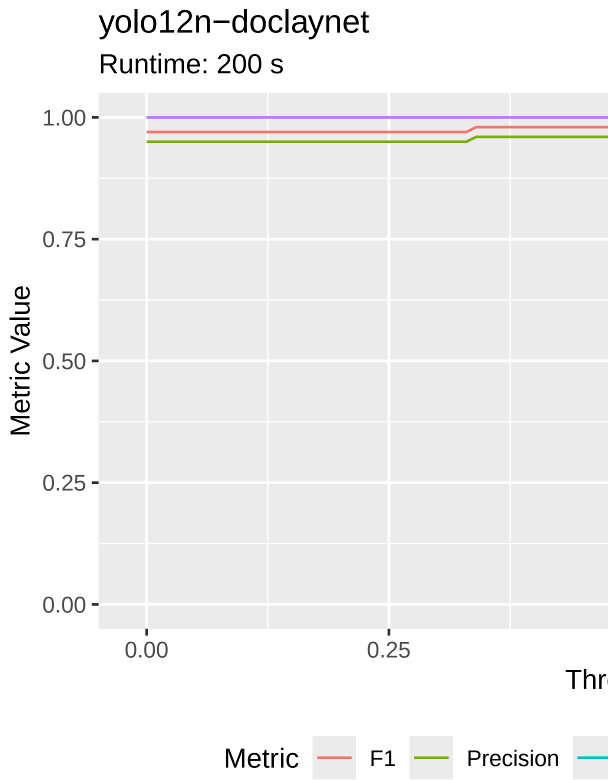
- yolo benchmark and table transformer
- skip classification with llm

not so important anymore

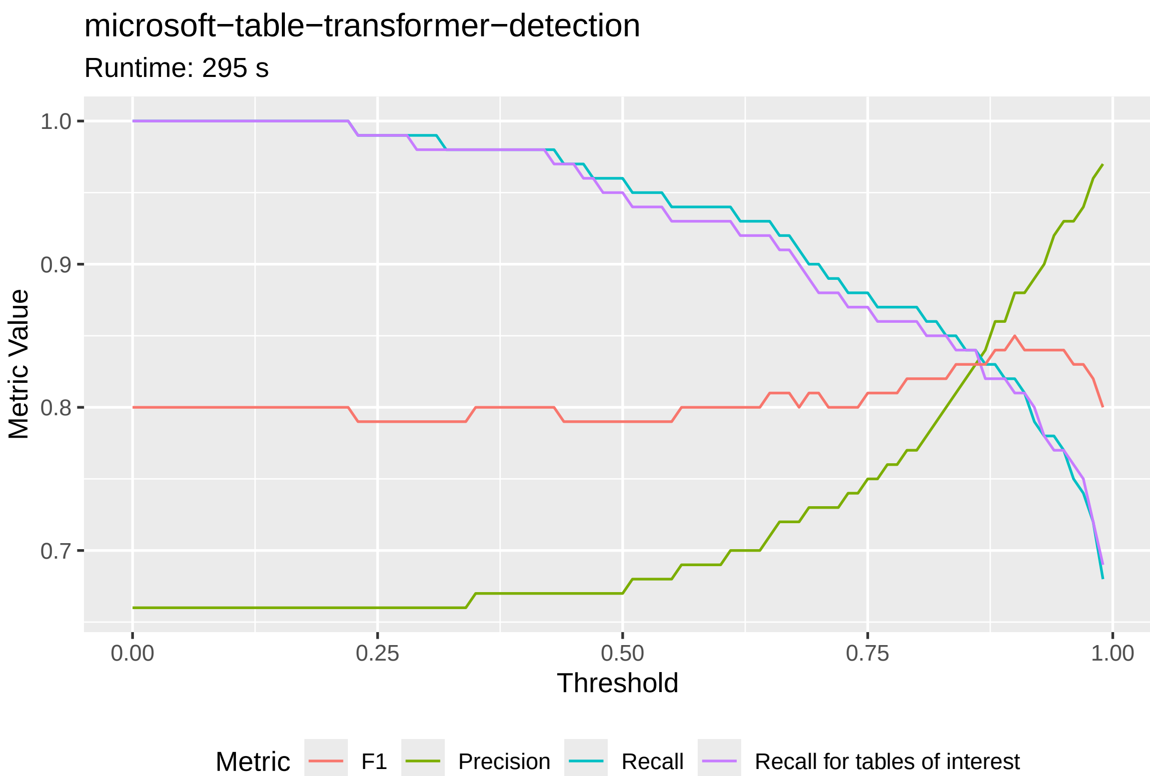


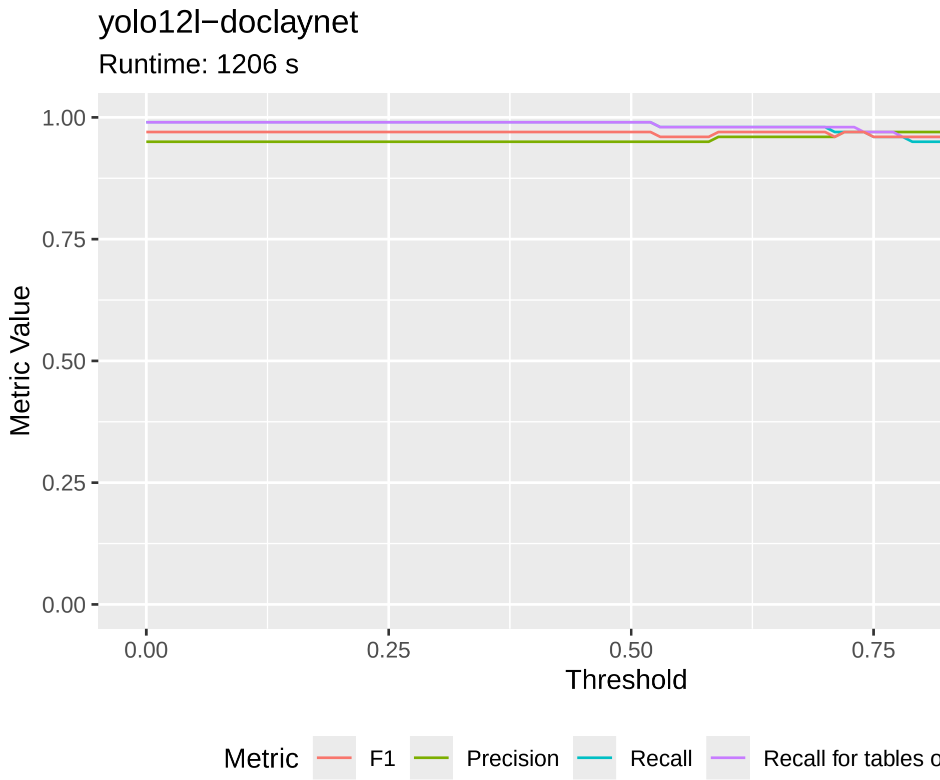
You see the plot for: microsoft-table-transformer-detection. (Click to stop automatic rotation.)



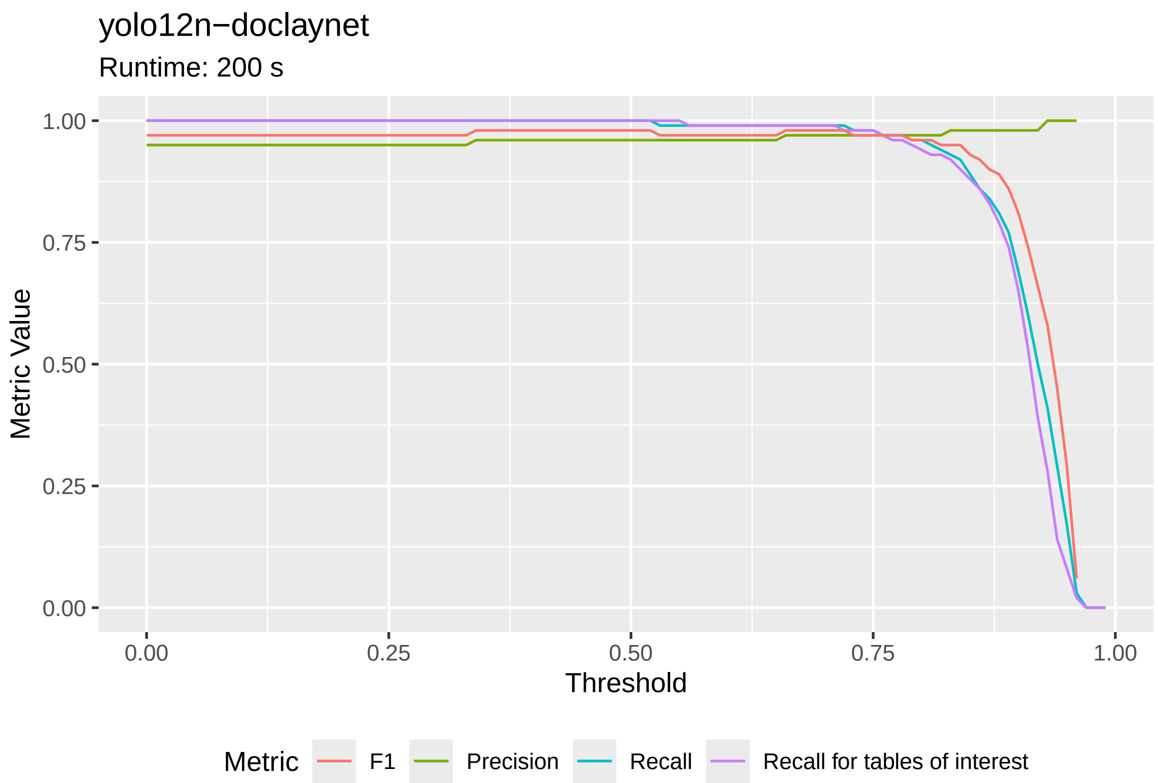


You see the plot for: yolo12l-doclaynet. (Click to stop automatic rotation.)
You see the plot for: yolo12n-doclaynet. (Click to stop automatic rotation.)





microsoft-table-transformer-detection



yolo12l-doclaynet
yolo12n-doclaynet

Table A.2: Comparing time (in seconds) for processing ten asset 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

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 have been conducted using the transformers library. When the problems of building a working vllm based docker image for the experiments it was 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 was Qwen 2.5 Instruct. The task was to extract the assets table for ten real example pages.

Table A.2 shows that the experiments with vllm library run are around four to five times faster. Processing the messages in a batched mode again is six to seven times faster.

The change of the experimental setup from transformers loop-based to vllm batched mode made is possible run the benchmark on whole PDF documents giving a sound estimate of the false positive rate in the page identification task (see section 5.1.3). Previous experiments have only been using a subset of pages that have been selected with the baseline 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 Tables

A.5.1 Classification

A.6 Figures

A.6.1 Page identification

A.6.1.1 Regex baseline

A.6.1.2 TOC understanding

A.6.1.3 Classification

Binary Binary classification F1 score over runtime limited to 60 minutes

Binary classification F1 score over runtime unlimited

Table A.3: 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

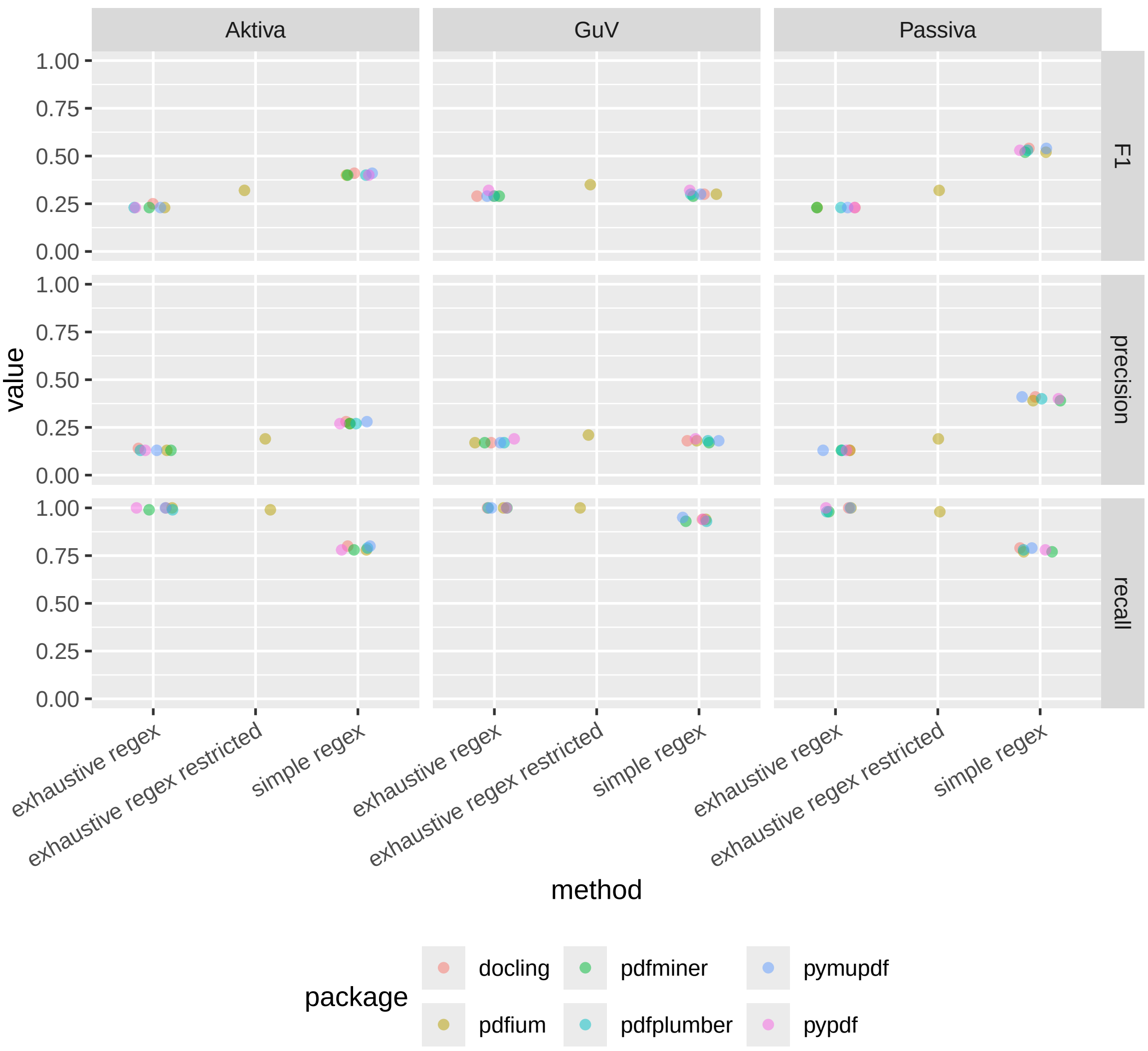


Figure A.1: Comparing page identification metrics for different regular expressions for each classification task by type of the target table.

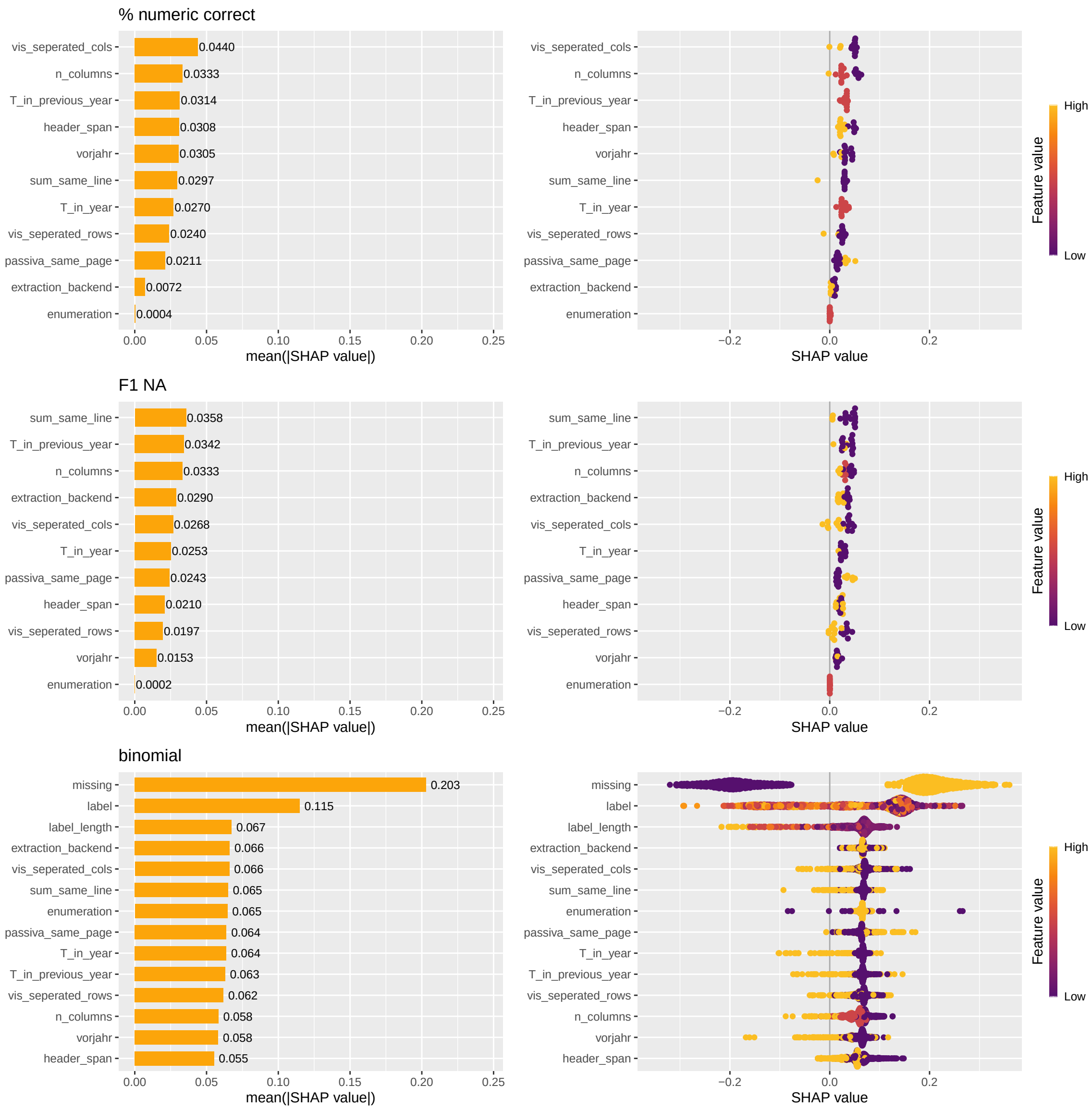


Figure A.2: Mean absolute SHAP values and beeswarm plots for real table extraction with regular expression approach

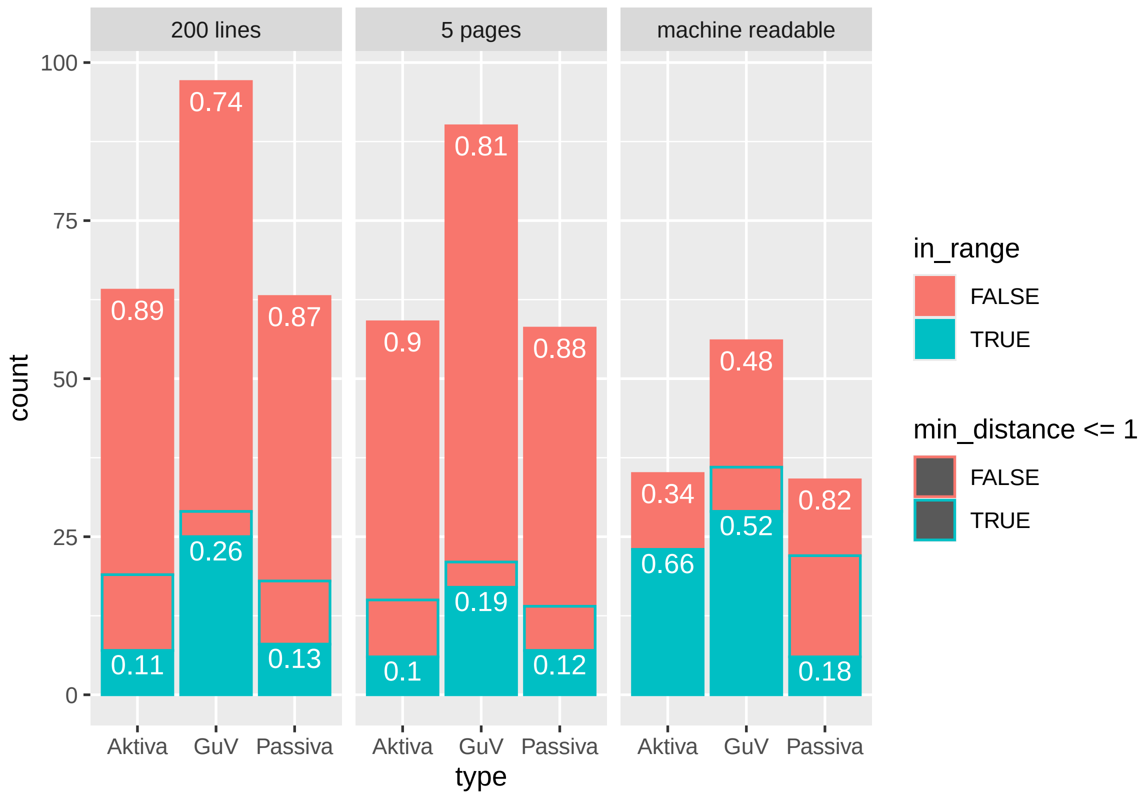


Figure A.3: Comparing number of fount TOC and amount of correct and incorrect predicted page ranges

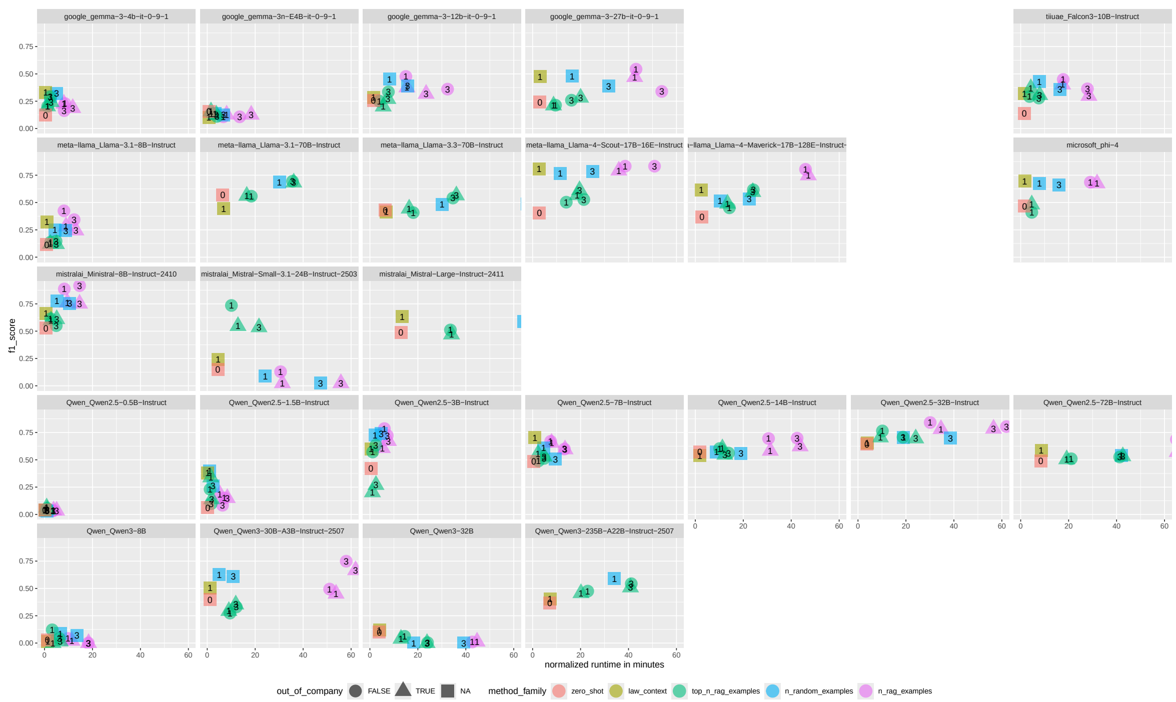


Figure A.4: Comparing F1 score over normalized runtime for binary classification task. The normalized runtime is given in minutes of processing on a single B200. The time to load the model into the VRAM is excluded. Focussing on small models showing only 60 minutes of runtime.

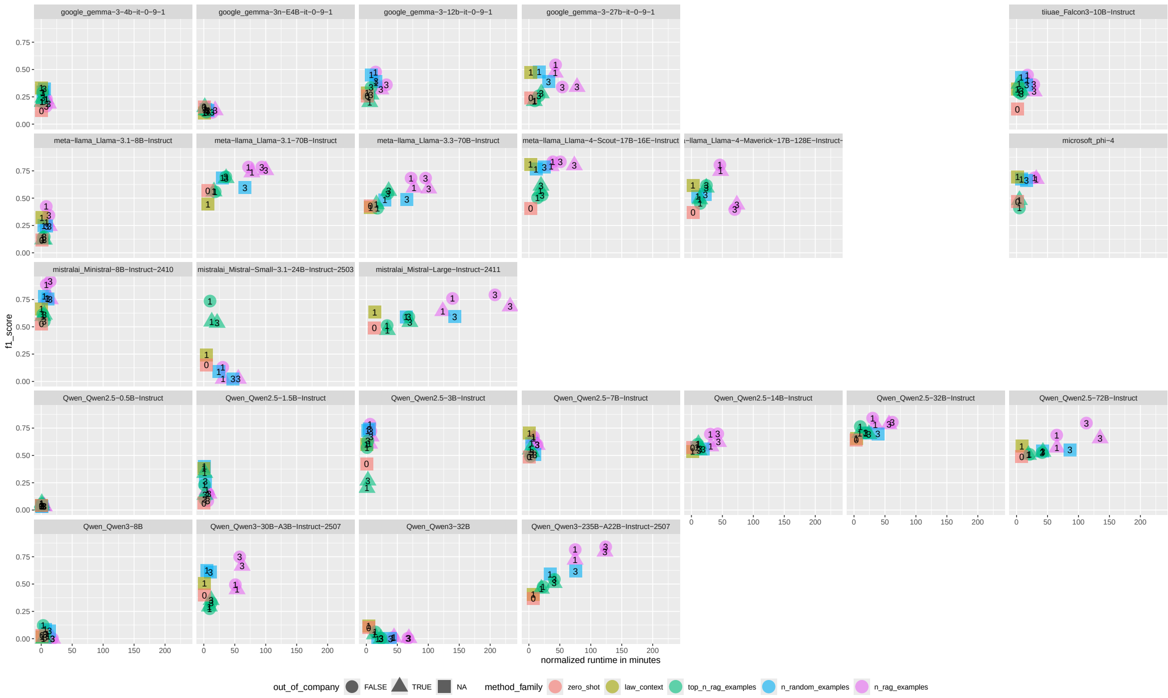


Figure A.5: Comparing F1 score over normalized runtime for binary classification task. The normalized runtime is given in minutes of processing on a single B200. The time to load the model into the VRAM is excluded.

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

A.6.2 Table extraction

Figure A.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.

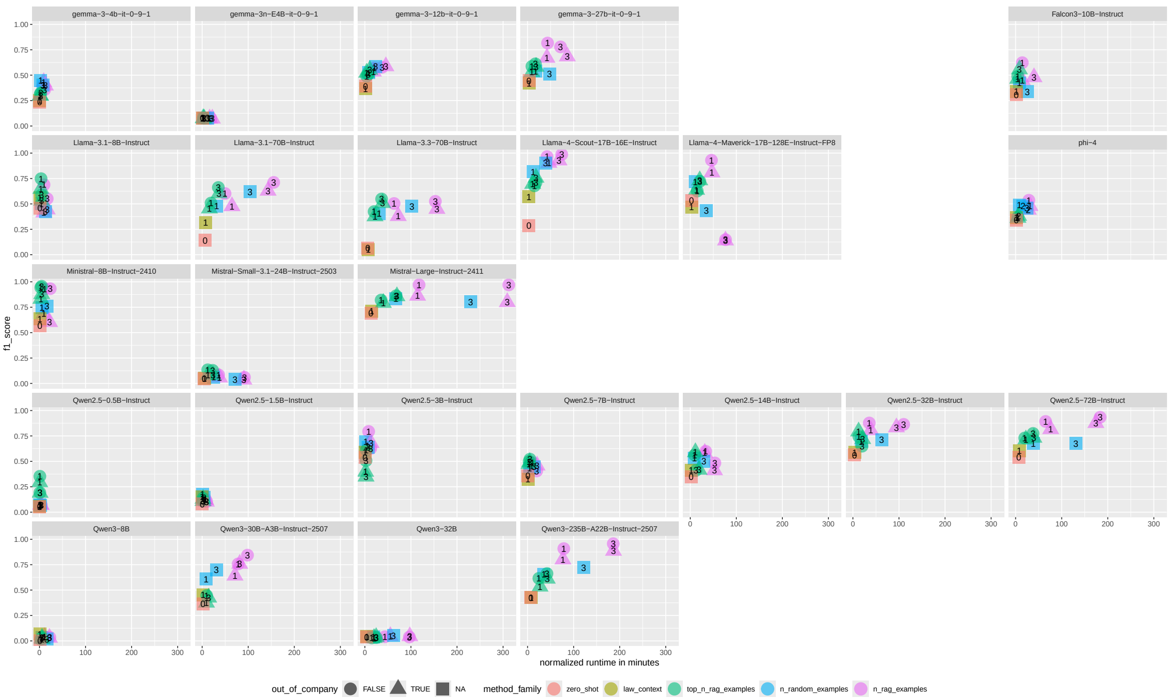


Figure A.7: Comparing F1 score over normalized runtime for multi-class classification task. The normalized runtime is given in minutes of processing on a single B200. The time to load the model into the VRAM is excluded.

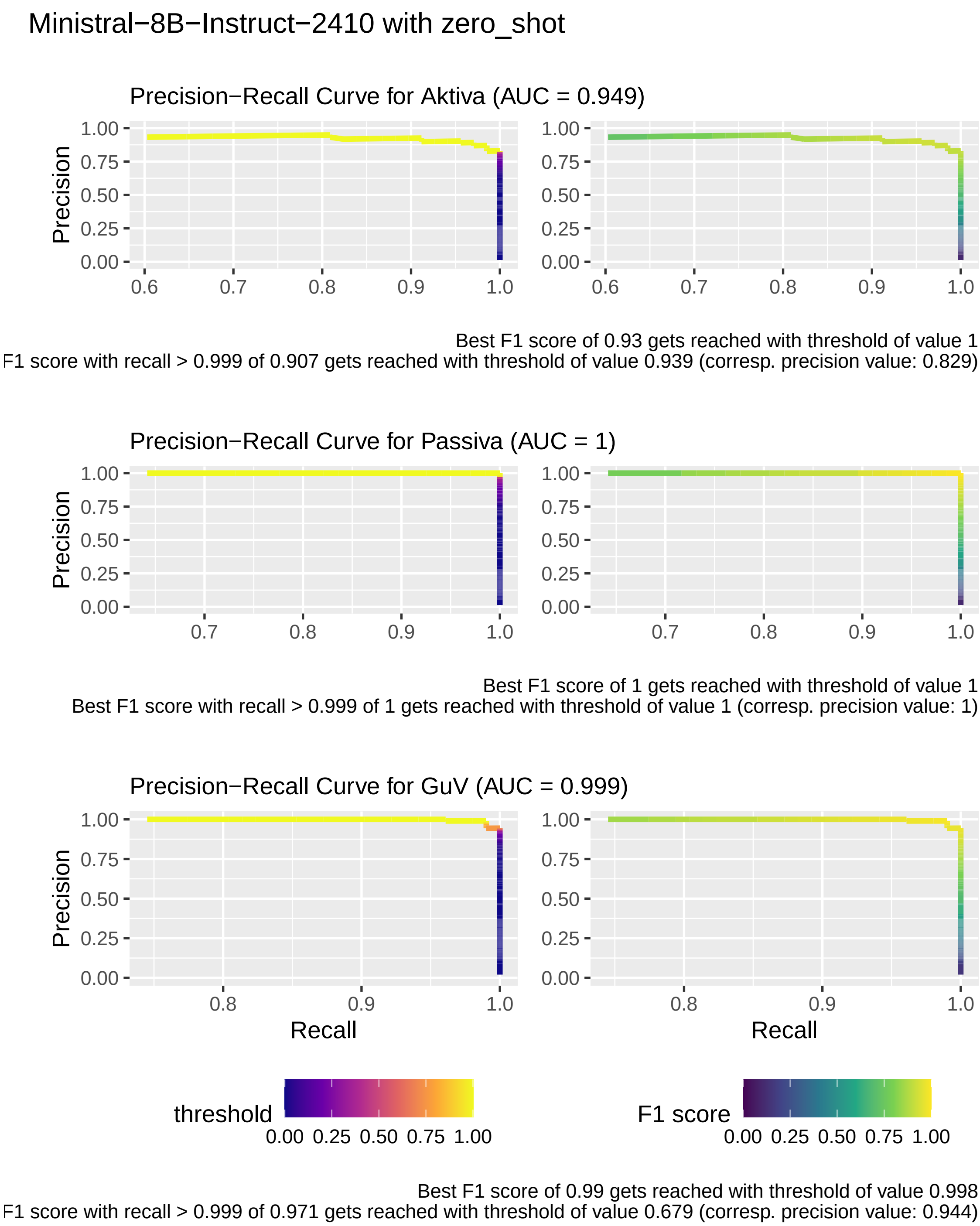


Figure A.8: Showing the precsion-recall-curve for Llama-4-Scout-17B-16E-Instruct.

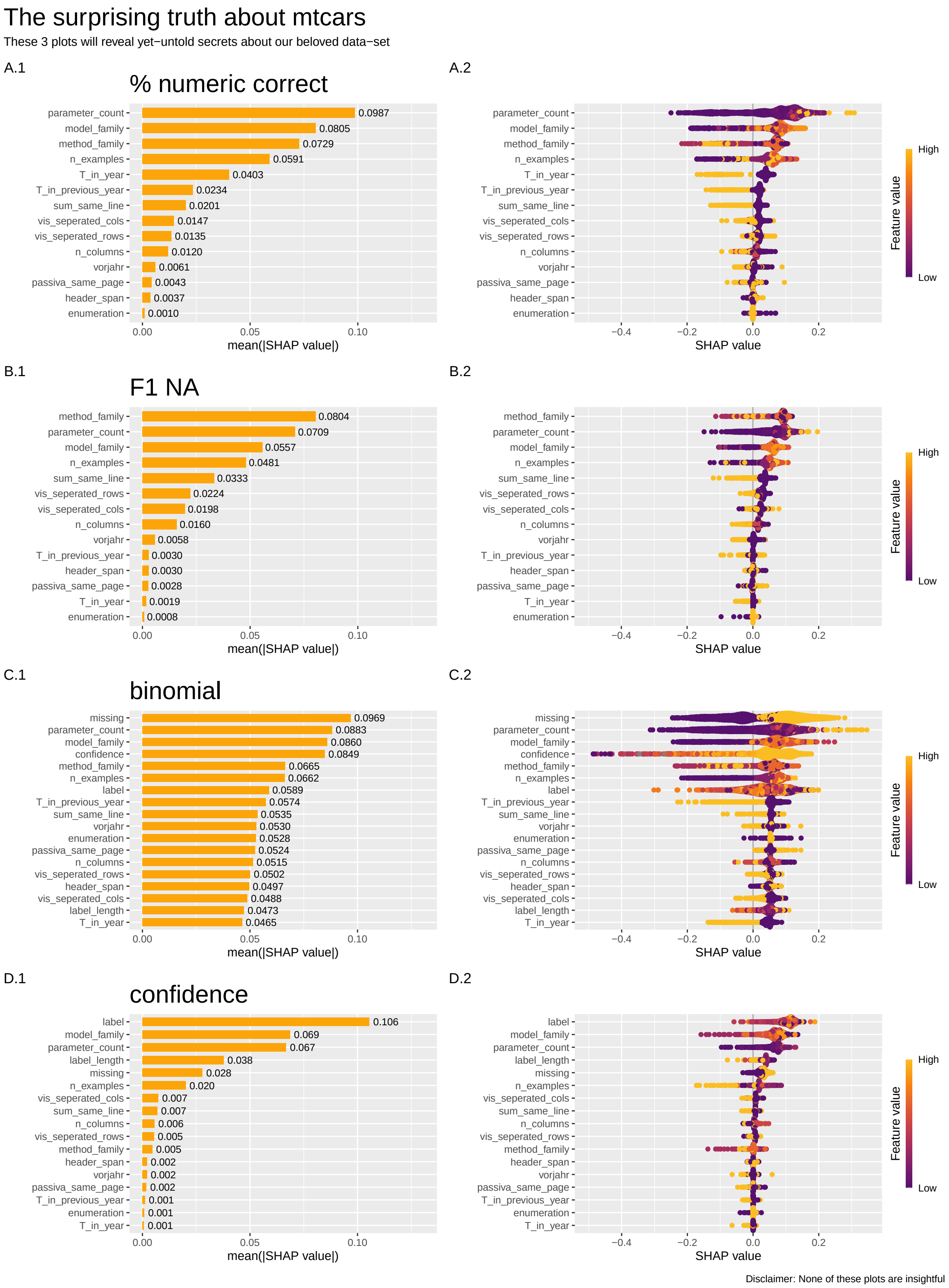


Figure A.9: Mean absolute SHAP values and beeswarm plots for real table extraction with LLMs

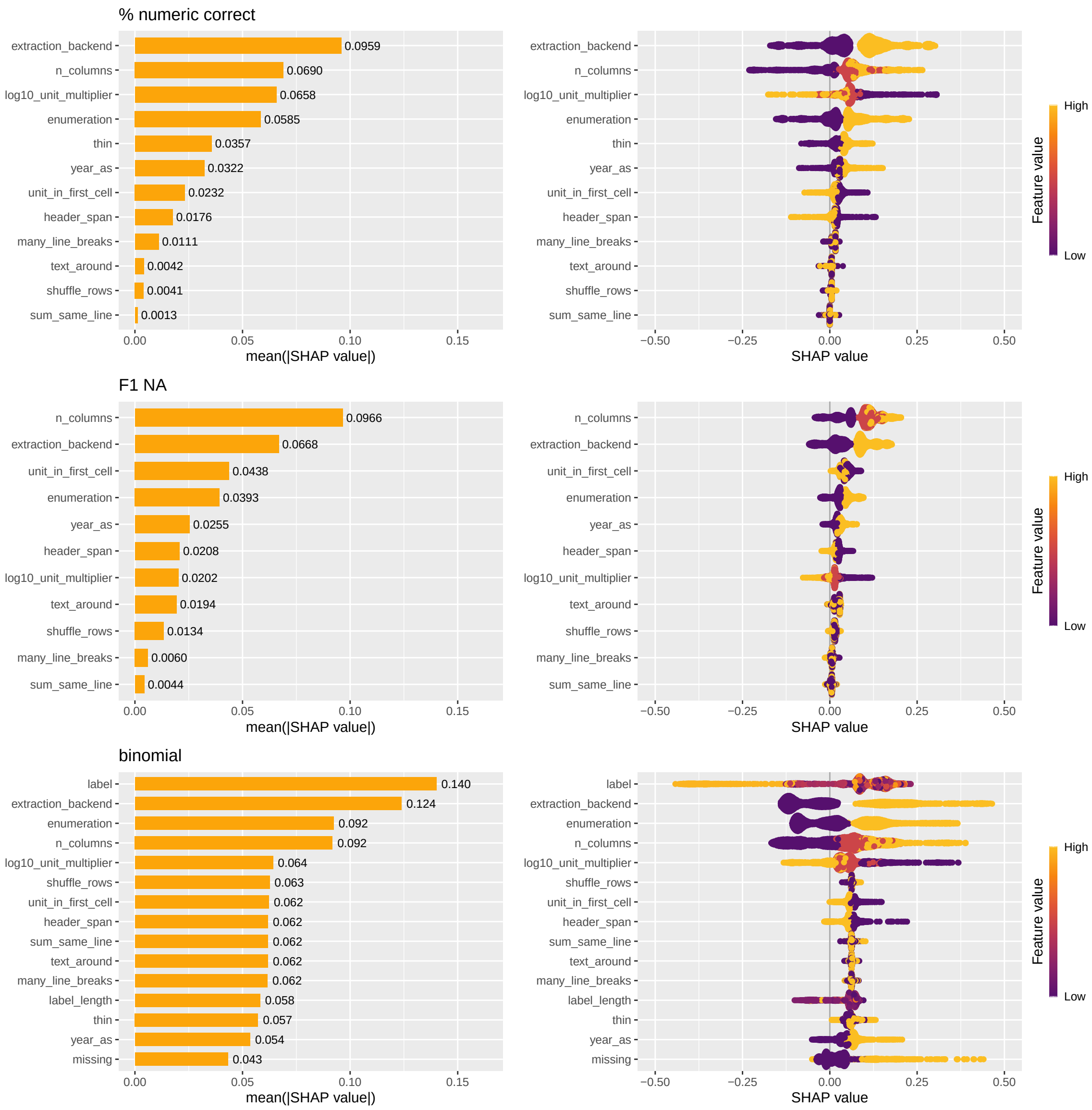
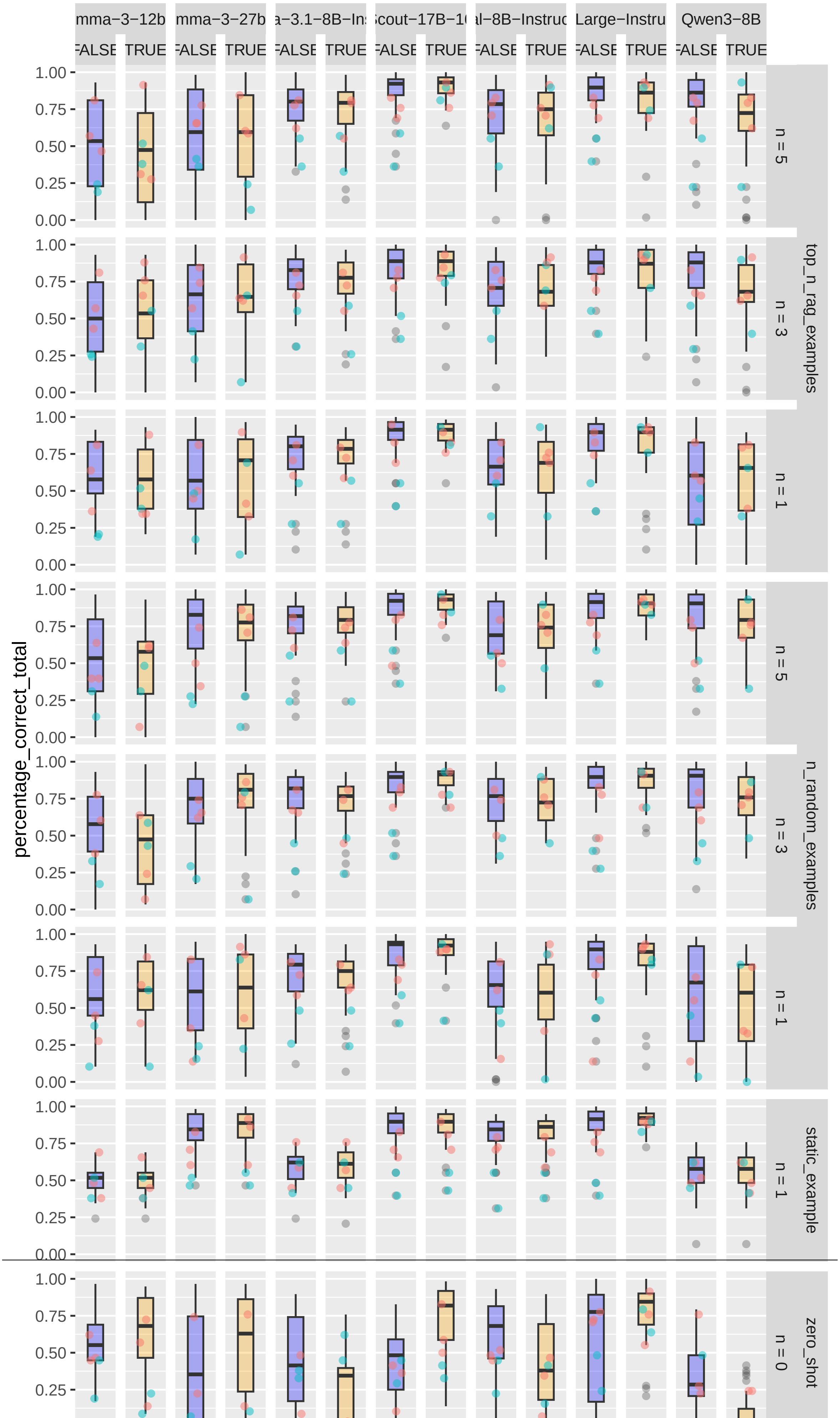
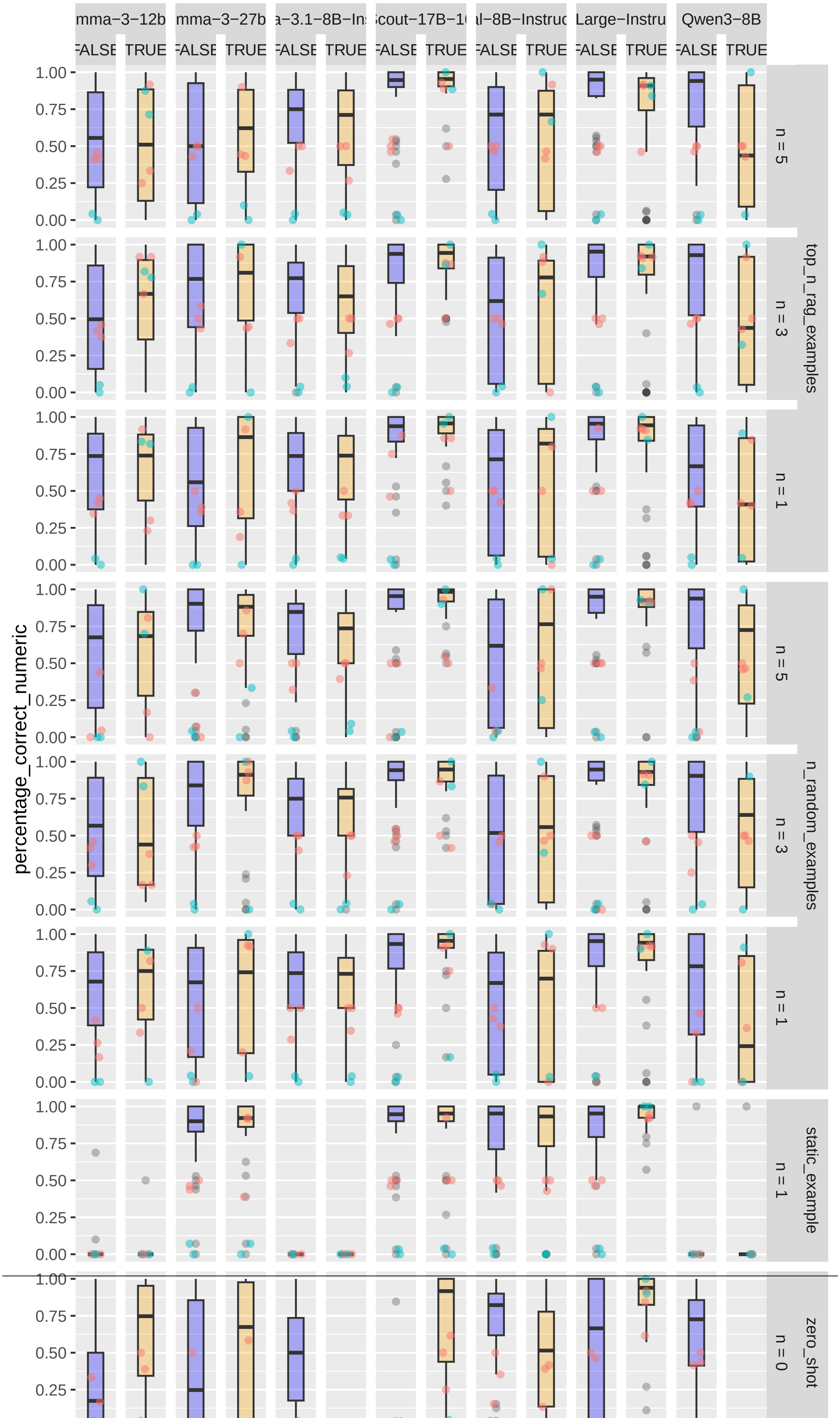
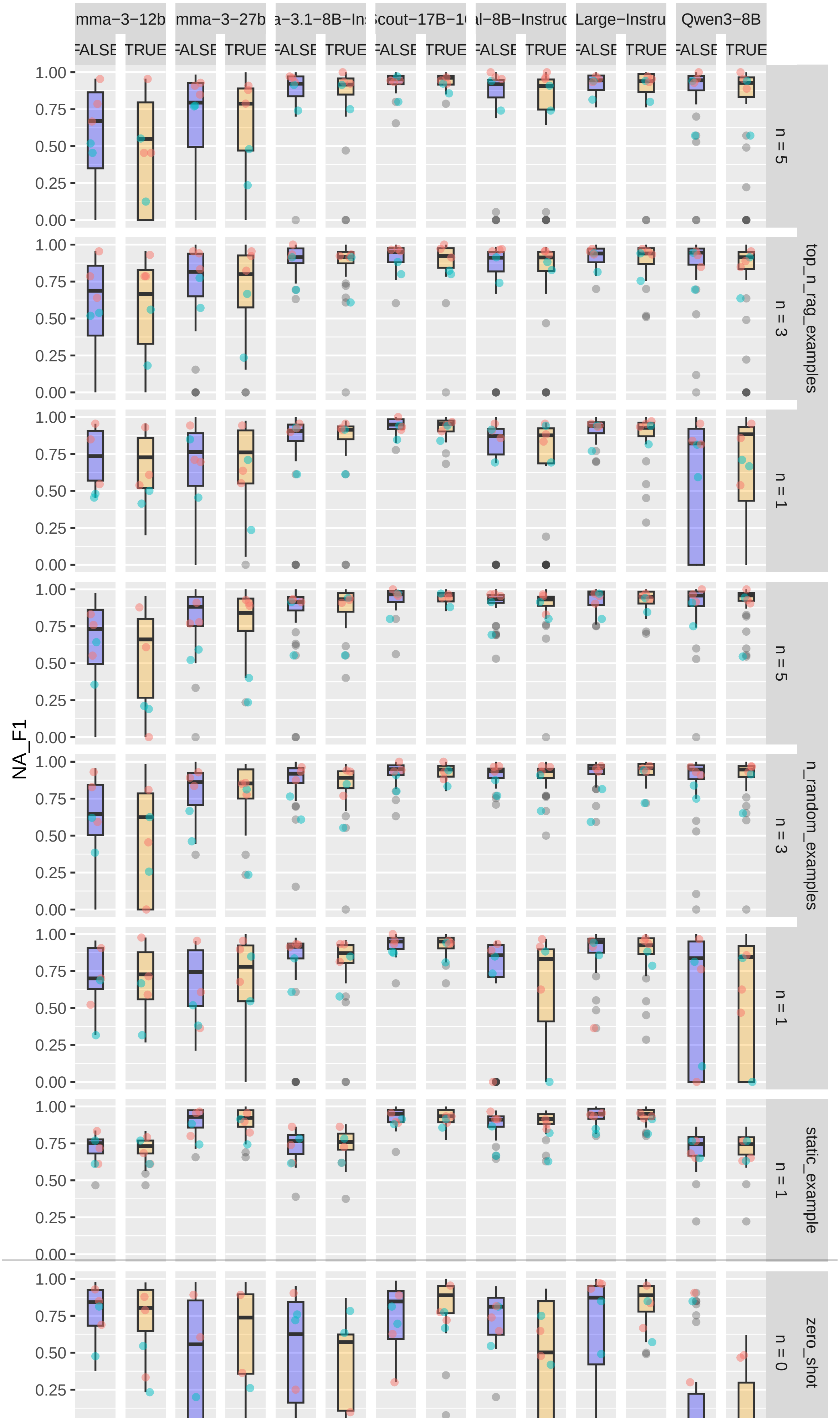


Figure A.10: Mean absolute SHAP values and beeswarm plots for synth table extraction with regular expression approach


mma−3−12b
mma−3−27b
a−3.1−8B−In:
scout−17B−1(
al−8B−Instruc
Large−Instru
Qwen3−8B
FALSE
TRUE
percentage_correct_total
1.00
0.75
0.50
0.25
0.00
n = 5
n = 3
n = 1
n = 0
top_n_rag_examples
n_random_examples
static_example
zero_shot


mma−3−12b
mma−3−27b
a−3.1−8B−In:
cout−17B−1
l−8B−Instruc
Large−Instru
Qwen3−8B
FALSE
TRUE
percentage_correct_numeric
1.00
0.75
0.50
0.25
0.00
n = 5
n = 3
n = 1
n = 0
top_n_rag_examples
n_random_examples
static_example
zero_shot


mma−3−12b
mma−3−27b
a−3.1−8B−In
cout−17B−1
l−8B−Instru
Large−Instru
Qwen3−8B
FALSE
TRUE
1.00
0.75
0.50
0.25
0.00
NA_F1
n = 5
n = 3
n = 1
n = 0
top_n_rag_examples
n_random_examples
static_example
zero_shot

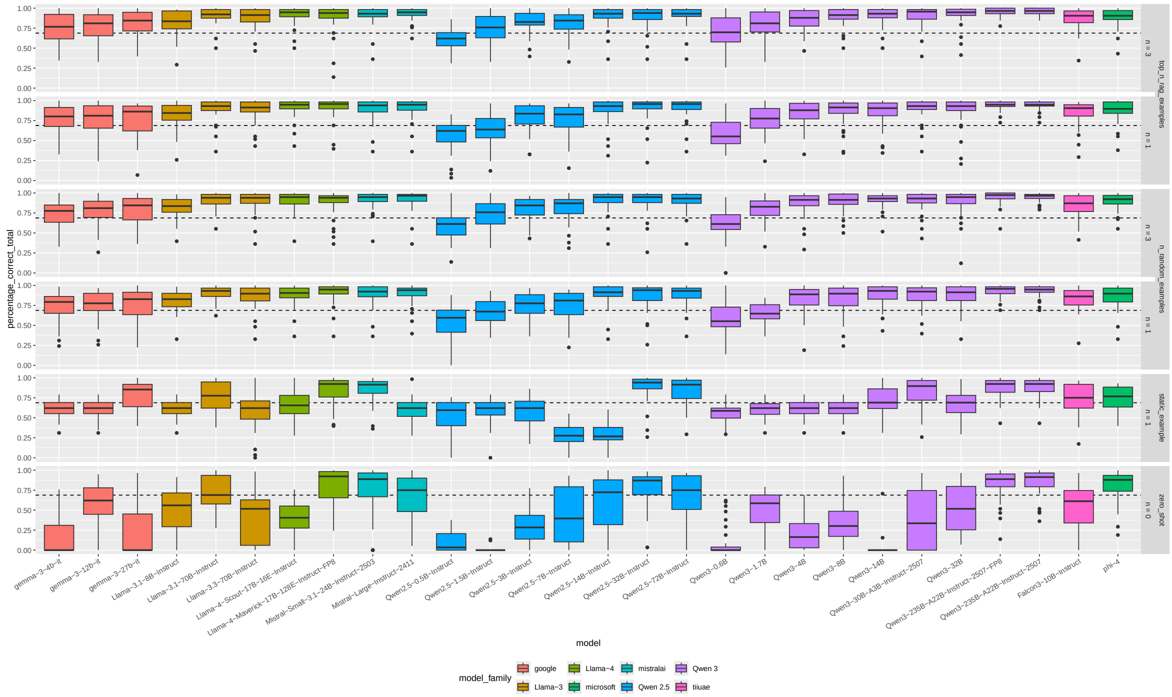


Figure A.14: Percentage of correct extracted or as missing categorized values for table extraction task on real Aktiva tables

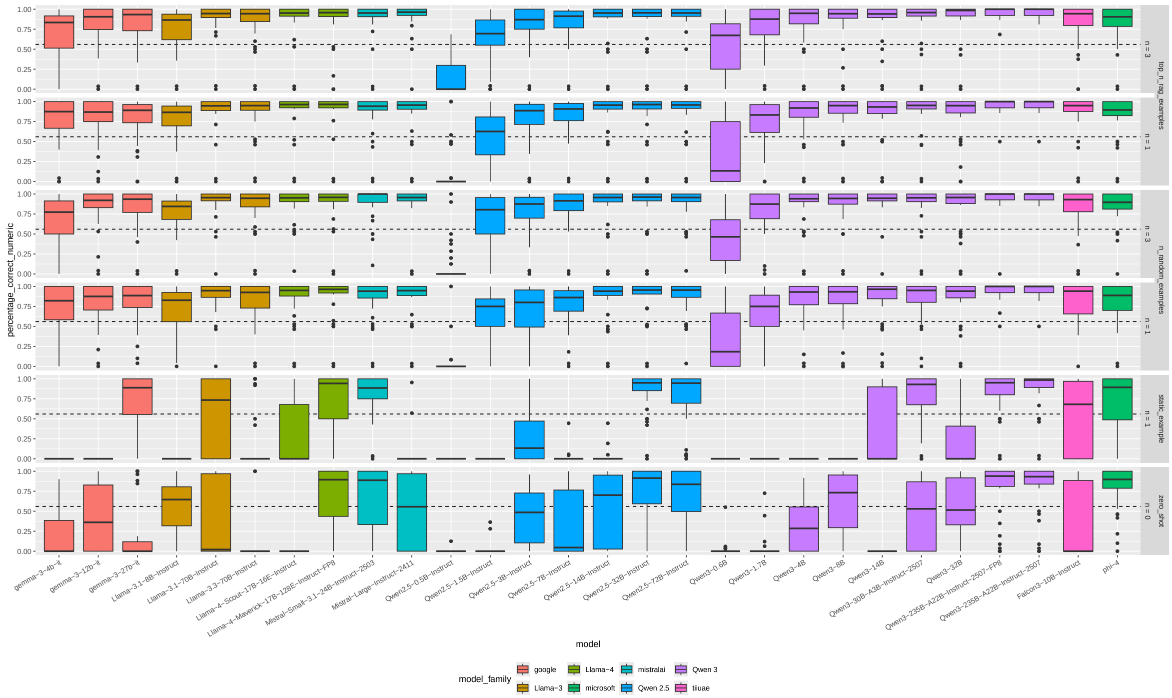


Figure A.15: Percentage of correct extracted numeric values for table extraction task on real Aktiva tables

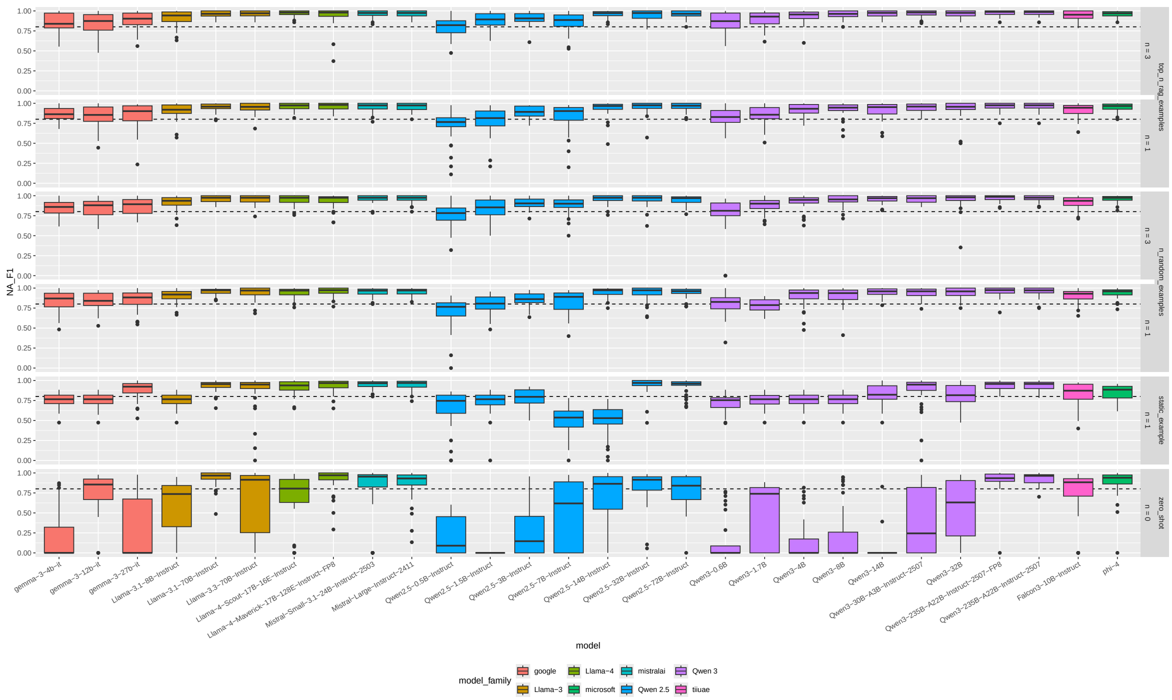


Figure A.16: F1 score for the missing classification if a value is missing for table extraction task on real Aktiva tables

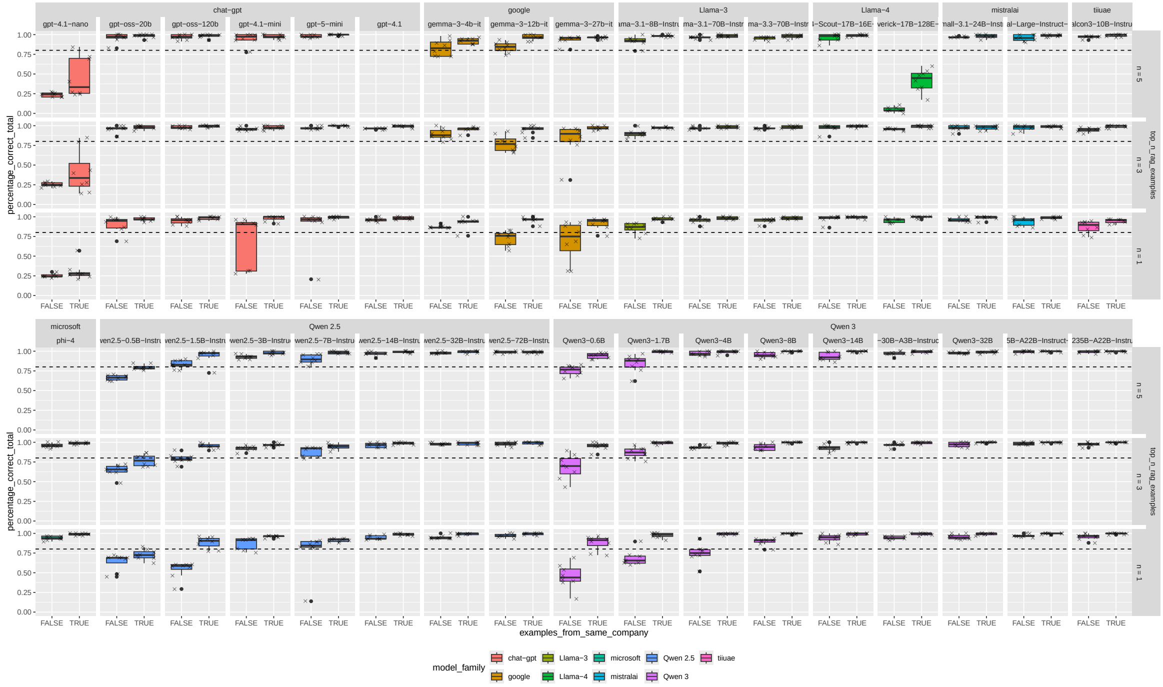


Figure A.17: Comparing the overall extraction performance depending on the condition if examples from the same company can be used (only for Amt für Statistik Berlin-Brandenburg).

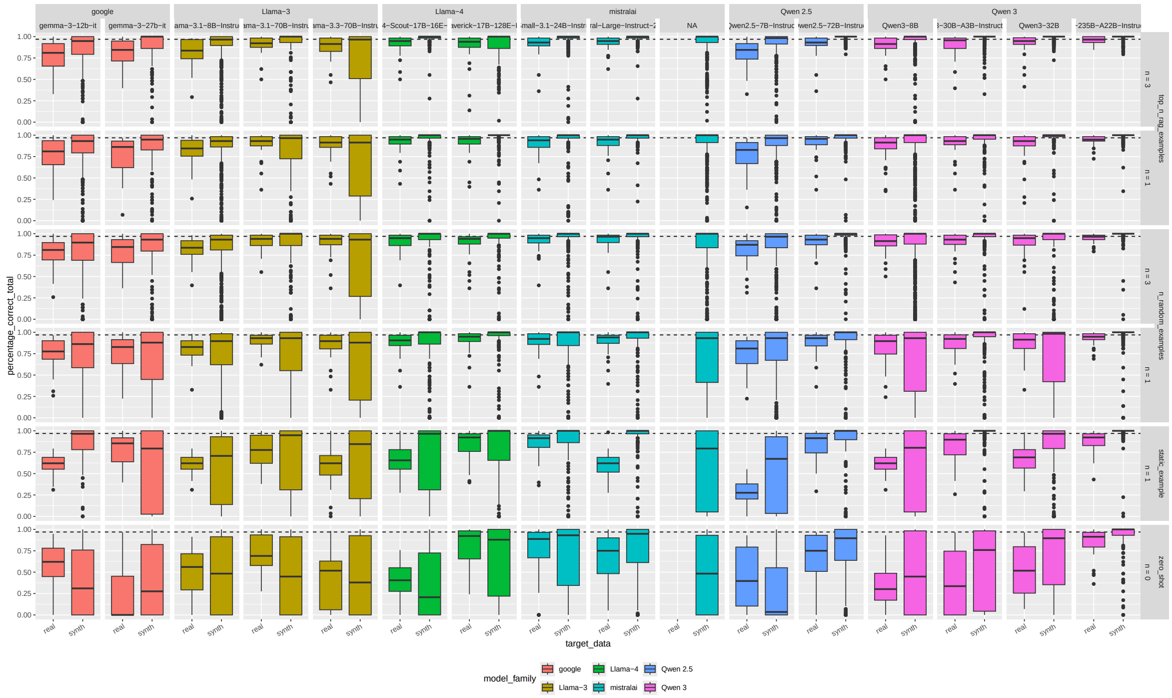


Figure A.18: Comparing the table extraction performance among real and synthetic Aktiva tables

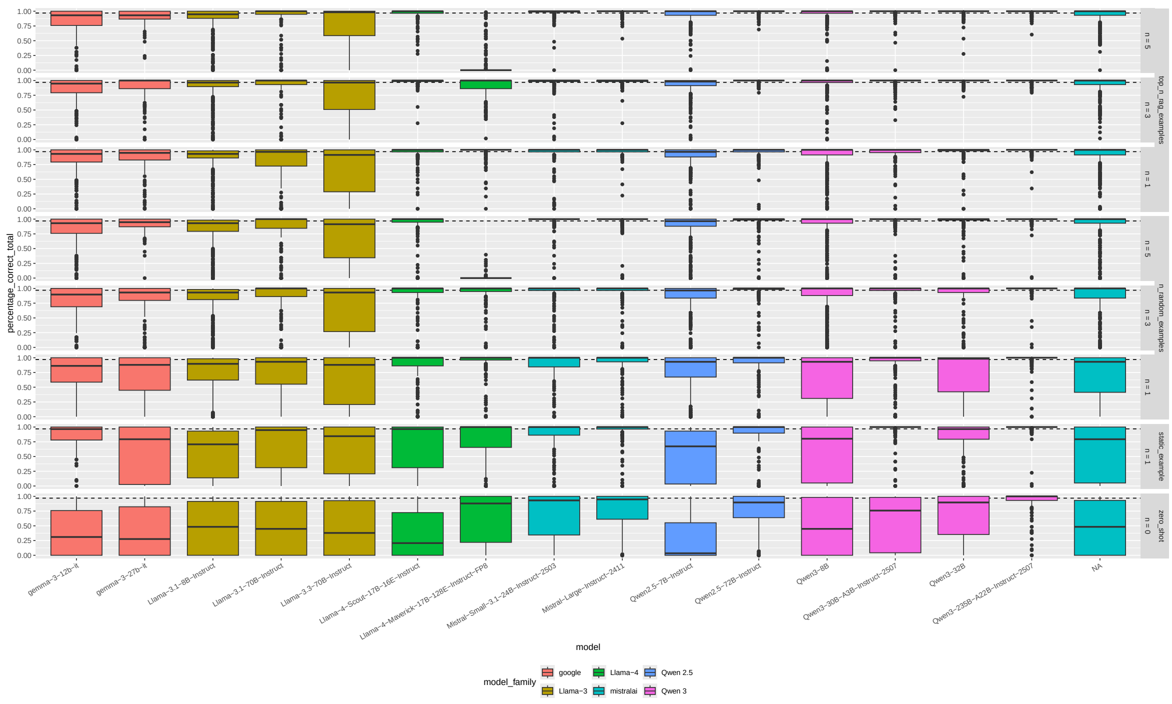


Figure A.19: Percentage of correct extracted or as missing categorized values for table extraction task on synthetic Aktiva tables

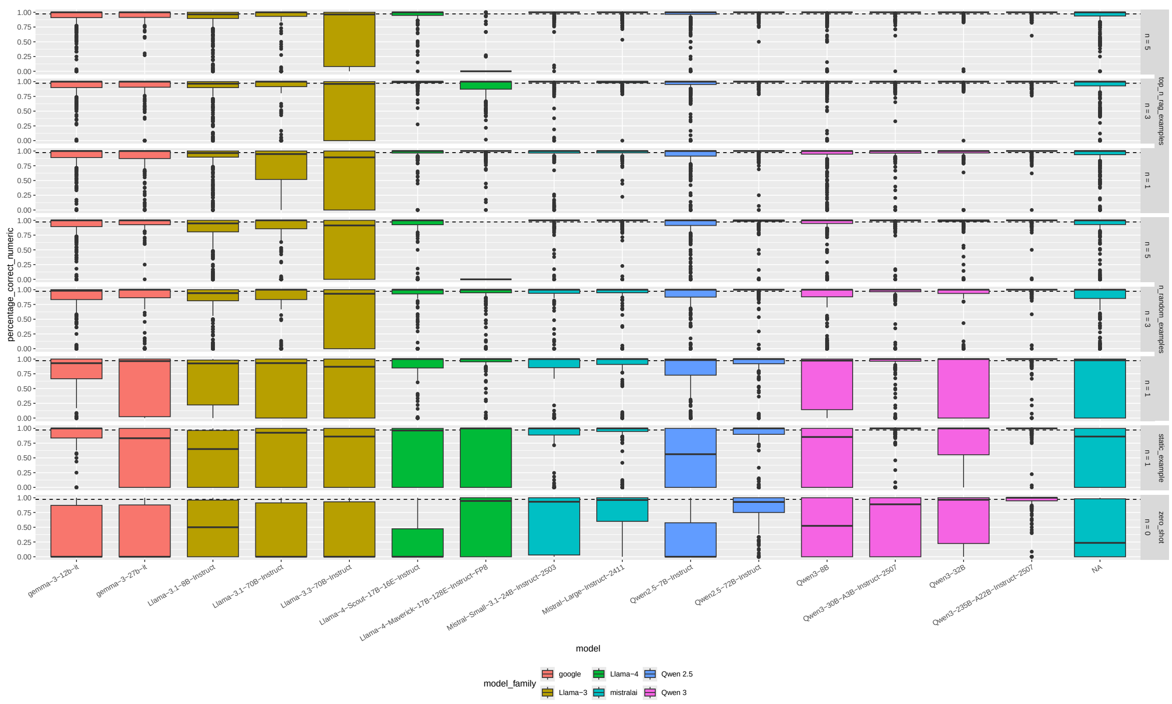


Figure A.20: Percentage of correct extracted numeric values for table extraction task on synthetic Aktiva tables

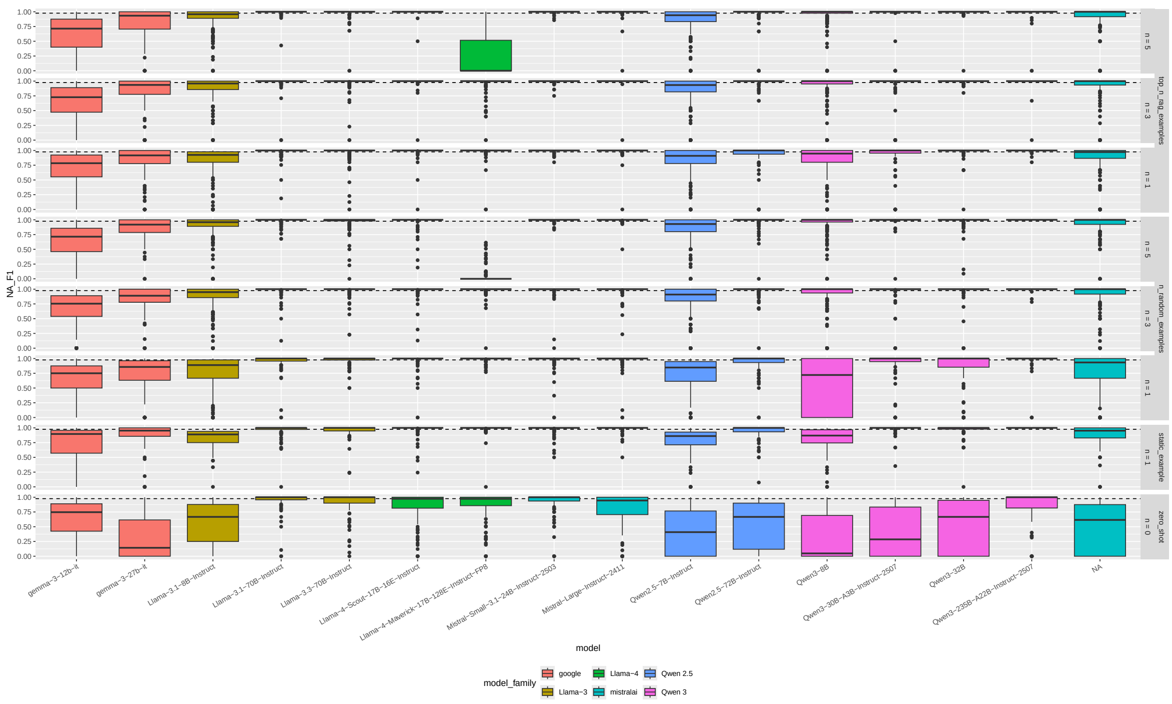


Figure A.21: F1 score for the missing classification if a value is missing for table extraction task on synthetic Aktiva tables

12sdgsdgj

Landscape

A.7 Annual Comprehensive Financial Report Balance Sheet

A.8 Extraction framework flow chart

A.9 Table extraction with regular expressions

Extract by pdfium for '../../benchmark_truth/synthetic_tables/separate_files/final/aktiva_table__3_columns__span_False__thin_Fa
€__enumeration_False__shuffle_True__text_around_True__max_length_50__sum_in_same_row_False__0.pdf':

```
A

ktiva(inMio. €)GeschäftsjahrVorjahr

Anlagevermögen Immaterielle Verm

ögensgegenstände

SelbstgeschaffenegewerblicheSchutzrechteund

ähnlicheRechteundWerte

0,184,77

Geschäfts-oderFirmenwert4,426,78

geleisteteAnzahlungen1,780,65

entgeltlicherworbeneKonzessionen, gewerbliche

SchutzrechteundähnlicheRechteundWertesowie

LizenzenansolchenRechtenundWerten

4,646,71

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
```

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

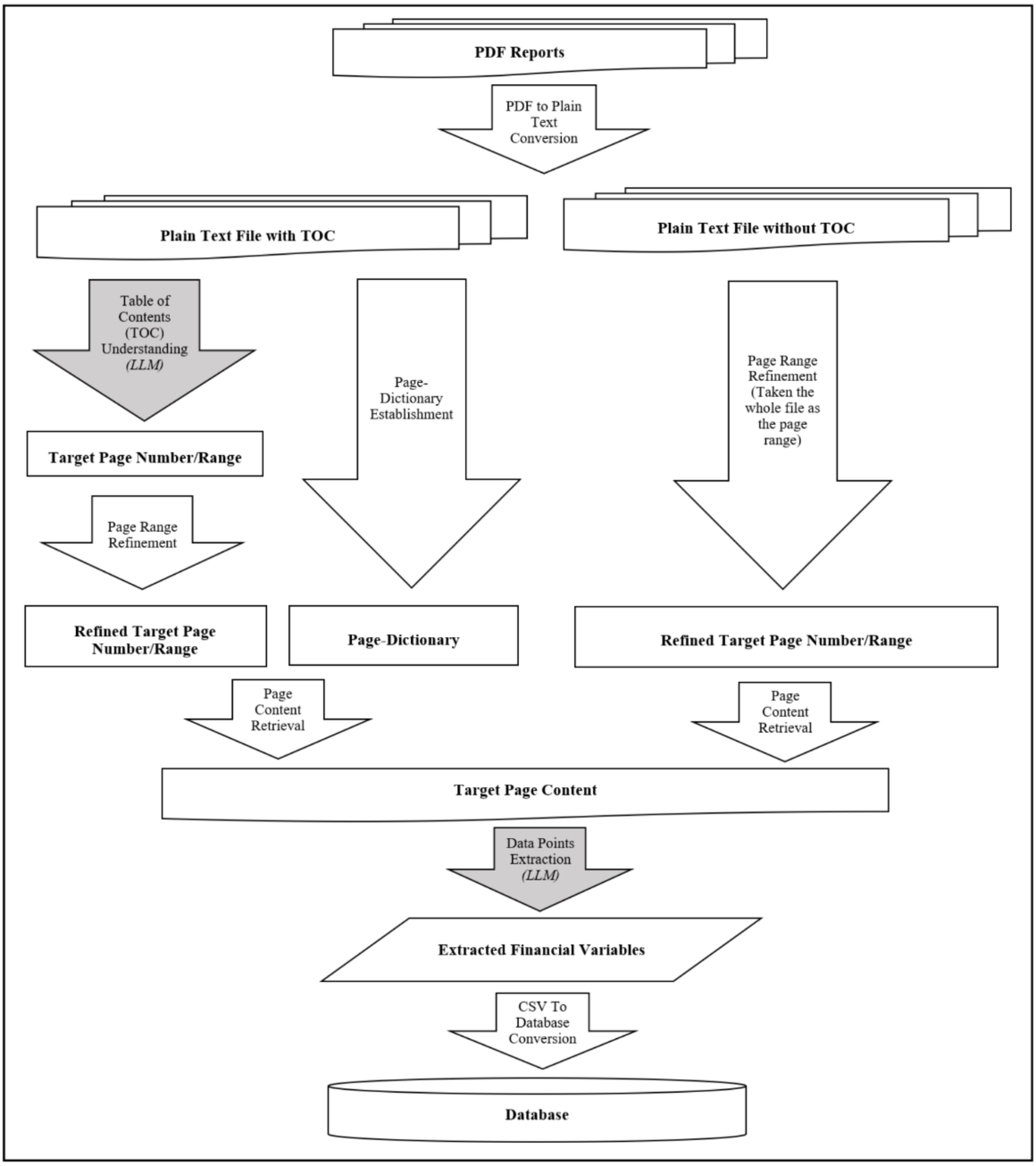


Figure A.23: Flowchart of the extraction framework

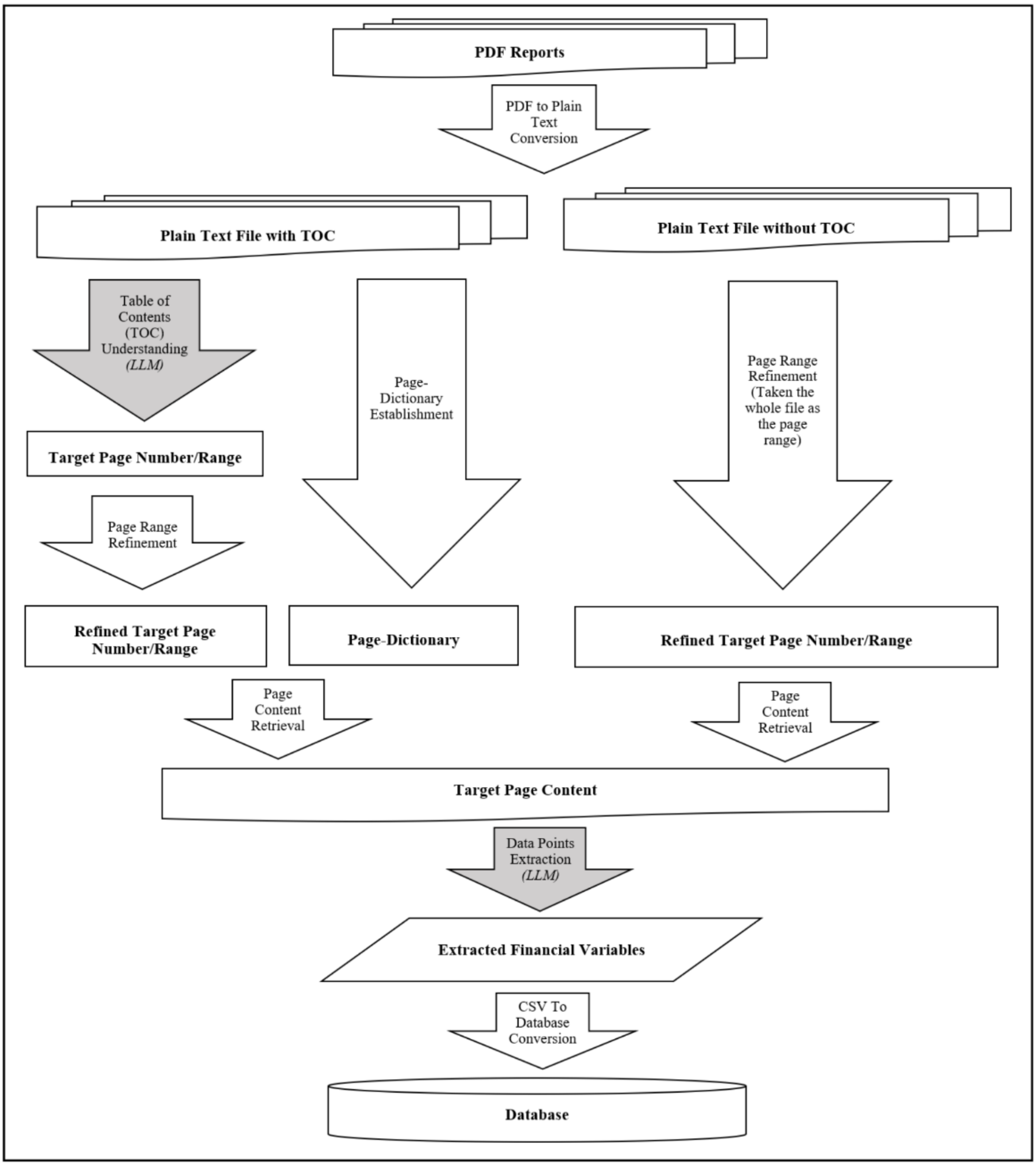


Figure A.24: Flowchart of the extraction framework of Li et al. (2023)

```
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
```