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

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.

Notes

- Qwen 2.5 hat zweiseitige GuV von IBB entdeckt und zur Anpassung der Ground Truth
- Google gemma war mit alter Klassifikation erfolgreich (anderer Prompt, mehr Seiten)

implementation nach methods

Goals and Learnings

Achieved:

- thesis with bookdown
- docker image creation
- cluster orchestration

- llm usage
- guided decoding

Missed:

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

Contents

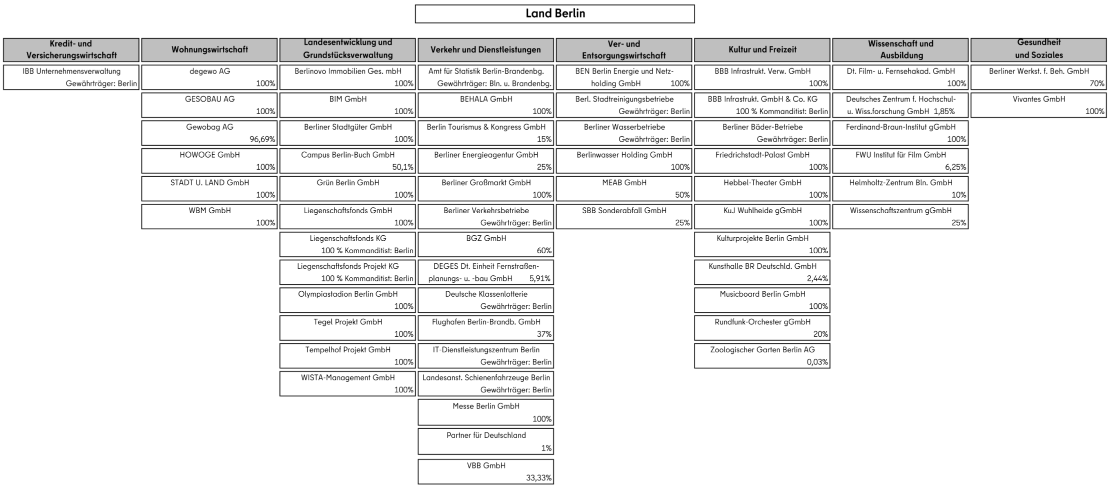
Chapter 1

Introduction

1.1 Motivation

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

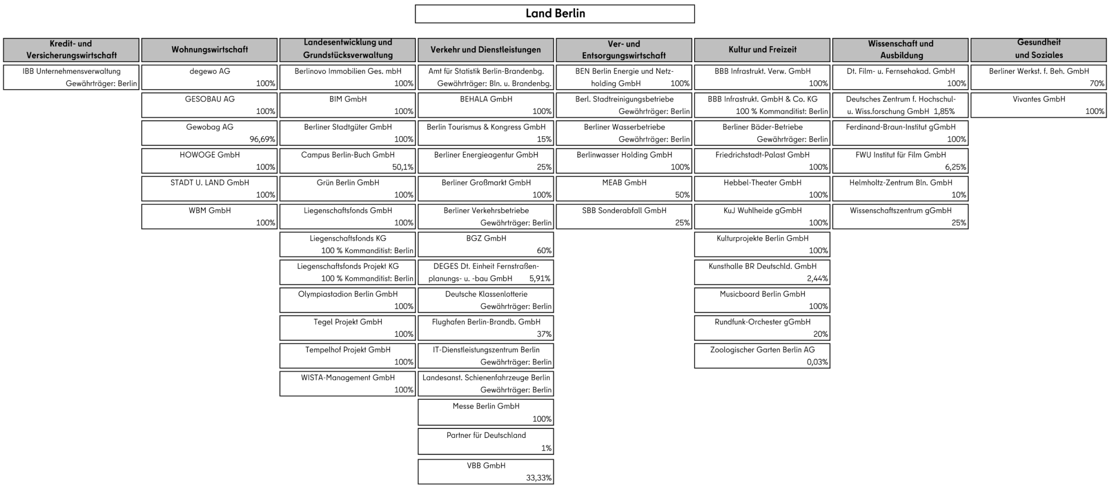
missing law to access digital data and no law to choose the format of the data extensible Business Reporting Language as a standard changing from HGB to IFSR



Land Berlin

Kredit- und Versicherungswirtschaft	Wohnungswirtschaft	Landesentwicklung und Grundstücksverwaltung	Verkehr und Dienstleistungen	Ver- und Entsorgungswirtschaft	Kultur und Freizeit	Wissenschaft und Ausbildung	Gesundheit und Soziales
IBB Unternehmensverwaltung Gewährträger: Berlin	degewo AG 100%	Berlinovo Immobilien Ges. mbH 100%	Amt für Statistik Berlin-Brandenbg. Gewährträger: Bln. u. Brandenbg.	BEN Berlin Energie und Netz-holding GmbH 100%	BBB Infrastrukt. Verw. GmbH 100%	Dt. Film- u. Fernsehakad. GmbH 100%	Berliner Werkst. f. Beh. GmbH 70%
	GESOBAU AG 100%	BIM GmbH 100%	BEHALA GmbH 100%	Berl. Stadtreinigungsbetriebe Gewährträger: Berlin	BBB Infrastrukt. GmbH & Co. KG 100 % Kommanditist: Berlin	Deutsches Zentrum f. Hochschul- u. Wiss.forschung GmbH 1,85%	Vivantes GmbH 100%
	Gewobag AG 96,69%	Berliner Stadtgüter GmbH 100%	Berlin Tourismus & Kongress GmbH 15%	Berliner Wasserbetriebe Gewährträger: Berlin	Berliner Bäder-Betriebe Gewährträger: Berlin	Ferdinand-Braun-Institut gGmbH 100%	
	HOWOGE GmbH 100%	Campus Berlin-Buch GmbH 50,1%	Berliner Energieagentur GmbH 25%	Berlinwasser Holding GmbH 100%	Friedrichstadt-Palast GmbH 100%	FWU Institut für Film GmbH 6,25%	
	STADT U. LAND GmbH 100%	Grün Berlin GmbH 100%	Berliner Großmarkt GmbH 100%	MEAB GmbH 50%	Hebbel-Theater GmbH 100%	Helmholtz-Zentrum Bln. GmbH 10%	
	WBM GmbH 100%	Liegenschaftsfonds GmbH 100%	Berliner Verkehrsbetriebe Gewährträger: Berlin	SBB Sonderabfall GmbH 25%	KuJ Wuhlheide gGmbH 100%	Wissenschaftszentrum gGmbH 25%	
		Liegenschaftsfonds KG 100 % Kommanditist: Berlin	BGZ GmbH 60%		Kulturprojekte Berlin GmbH 100%		
		Liegenschaftsfonds Projekt KG 100 % Kommanditist: Berlin	DEGES Dt. Einheit Fernstraßen-planungs- u. -bau GmbH 5,91%		Kunsthalle BR Deutschld. GmbH 2,44%		
		Olympiastadion Berlin GmbH 100%	Deutsche Klassenlotterie Gewährträger: Berlin		Musicboard Berlin GmbH 100%		
		Tegel Projekt GmbH 100%	Flughafen Berlin-Brandb. GmbH 37%		Rundfunk-Orchester gGmbH 20%		
		Tempelhof Projekt GmbH 100%	IT-Dienstleistungszentrum Berlin Gewährträger: Berlin		Zoologischer Garten Berlin AG 0,03%		
		WISTA-Management GmbH 100%	Landesanst. Schienenfahrzeuge Berlin Gewährträger: Berlin				
			Messe Berlin GmbH 100%				
			Partner für Deutschland 1%				
			VBB GmbH 33,33%				

Figure 1.1: Companies Berlin has holds share at



Land Berlin

Kredit- und Versicherungswirtschaft	Wohnungswirtschaft	Landesentwicklung und Grundstücksverwaltung	Verkehr und Dienstleistungen	Ver- und Entsorgungswirtschaft	Kultur und Freizeit	Wissenschaft und Ausbildung	Gesundheit und Soziales
IBB Unternehmensverwaltung Gewährträger: Berlin	degewo AG 100%	Berlinovo Immobilien Ges. mbH 100%	Amt für Statistik Berlin-Brandenbg. Gewährträger: Bln. u. Brandenbg.	BEN Berlin Energie und Netz-holding GmbH 100%	BBB Infrastrukt. Verw. GmbH 100%	Dt. Film- u. Fernsehakad. GmbH 100%	Berliner Werkst. f. Beh. GmbH 70%
	GESOBAU AG 100%	BIM GmbH 100%	BEHALA GmbH 100%	Berl. Stadtreinigungsbetriebe Gewährträger: Berlin	BBB Infrastrukt. GmbH & Co. KG 100 % Kommanditist: Berlin	Deutsches Zentrum f. Hochschul- u. Wiss.forschung GmbH 1,85%	Vivantes GmbH 100%
	Gewobag AG 96,69%	Berliner Stadtgüter GmbH 100%	Berlin Tourismus & Kongress GmbH 15%	Berliner Wasserbetriebe Gewährträger: Berlin	Berliner Bäder-Betriebe Gewährträger: Berlin	Ferdinand-Braun-Institut gGmbH 100%	
	HOWOGE GmbH 100%	Campus Berlin-Buch GmbH 50,1%	Berliner Energieagentur GmbH 25%	Berlinwasser Holding GmbH 100%	Friedrichstadt-Palast GmbH 100%	FWU Institut für Film GmbH 6,25%	
	STADT U. LAND GmbH 100%	Grün Berlin GmbH 100%	Berliner Großmarkt GmbH 100%	MEAB GmbH 50%	Hebbel-Theater GmbH 100%	Helmholtz-Zentrum Bln. GmbH 10%	
	WBM GmbH 100%	Liegenschaftsfonds GmbH 100%	Berliner Verkehrsbetriebe Gewährträger: Berlin	SBB Sonderabfall GmbH 25%	KuJ Wuhlheide gGmbH 100%	Wissenschaftszentrum gGmbH 25%	
		Liegenschaftsfonds KG 100 % Kommanditist: Berlin	BGZ GmbH 60%		Kulturprojekte Berlin GmbH 100%		
		Liegenschaftsfonds Projekt KG 100 % Kommanditist: Berlin	DEGES Dt. Einheit Fernstraßen-planungs- u. -bau GmbH 5,91%		Kunsthalle BR Deutschld. GmbH 2,44%		
		Olympiastadion Berlin GmbH 100%	Deutsche Klassenlotterie Gewährträger: Berlin		Musicboard Berlin GmbH 100%		
		Tegel Projekt GmbH 100%	Flughafen Berlin-Brandb. GmbH 37%		Rundfunk-Orchester gGmbH 20%		
		Tempelhof Projekt GmbH 100%	IT-Dienstleistungszentrum Berlin Gewährträger: Berlin		Zoologischer Garten Berlin AG 0,03%		
		WISTA-Management GmbH 100%	Landesanst. Schienenfahrzeuge Berlin Gewährträger: Berlin				
			Messe Berlin GmbH 100%				
			Partner für Deutschland 1%				
			VBB GmbH 33,33%				

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 LLMs 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 abovehttp://127.0.0.1:29003/rmd_output/4/images/beteili

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 Basic terms

2.2 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?

2.2.1 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)

2.3 optinal more topics like previous

2.4 Summary (0.5 p)

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

2.5 To place in chapters above

2.6 Table extraction tasks

2.6.1 Difficulties

- Beispielbilder

2.7 Document Extrtaction Process

2.7.1 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

2.7.1.1 Vision Grid Transformer

2.7.2

2.8 Tools

2.8.1 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

The first task to solve, for a fully autonomous solution, is to identify the pages where the tables of interest are located. For benchmarking 74 annual reports from 7 companies have been used. For this benchmark we limit the tables of interest to those that show **Aktiva**, **Passiva** and **Gewinn- und Verlustrechnung**.

In those documents there are 252 pages of interest holding 265 relevant tables. On 13 pages there have been two tables (**Aktiva** and **Passiva**) on a single page. 21 tables are spread over two pages. In 8 documents there have been multiple tables per type of interest, distributed among the three types of tables as following:

type	count
Aktiva	7
GuV	8
Passiva	7

As a baseline a simple regex approach was used.

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

3.2.1.1.1 LLM based

3.2.1.2 Term frequency based

3.2.1.2.1 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

3.3.3.0.1 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

5.1 Page identification

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

5.1.1 Baseline: Regex

Building a sound regular expression often is an iterative process. In a first approach a very simple one was implemented.

Comparing the differences in the metrics based on the different text extraction libraries it can be said that the extracted text is very similar but not identical. Since the resukts are not depending on the used text extraction library the *exhaustive regex restricted* has only been run with the fast text extraction library *pdfium*. The results of the regex based page identification are presented in the following tables.

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

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.

The regular expressions can be found in the appendix (see 5.1.1).

General bad precision. Increasing recall degrades precision even further. number of pages positive identified total; used as subset for table identification task

Table 5.1: Comparing page identification metrics for different regular expressions for classification task 'Aktiva'

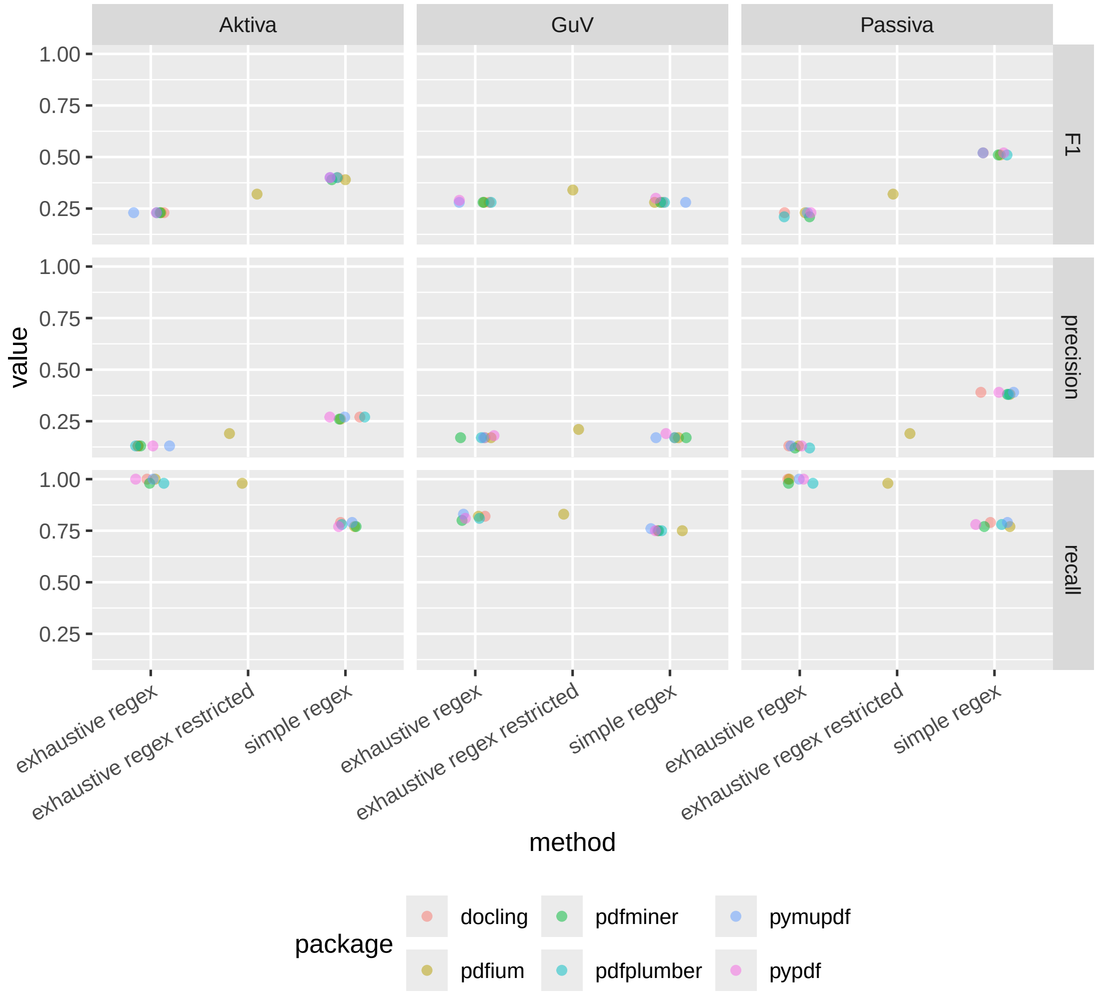
method	stat	precision	recall	F1
simple regex	mean	{0.267}	0.778	{0.397}
simple regex	sd	0.005	0.01	0.005
exhaustive regex restricted	mean	0.19	0.98	0.32
exhaustive regex restricted	sd	NA	NA	NA
exhaustive regex	mean	0.13	{0.993}	0.23
exhaustive regex	sd	0	0.01	0

Table 5.2: Comparing page identification metrics for different regular expressions for classification task 'Passiva'

method	stat	precision	recall	F1
simple regex	mean	{0.385}	0.78	{0.515}
simple regex	sd	0.005	0.009	0.005
exhaustive regex restricted	mean	0.19	0.98	0.32
exhaustive regex restricted	sd	NA	NA	NA
exhaustive regex	mean	0.127	{0.993}	0.223
exhaustive regex	sd	0.005	0.01	0.01

Table 5.3: Comparing page identification metrics for different regular expressions for classification task 'Gewinn und Verlustrechnung'

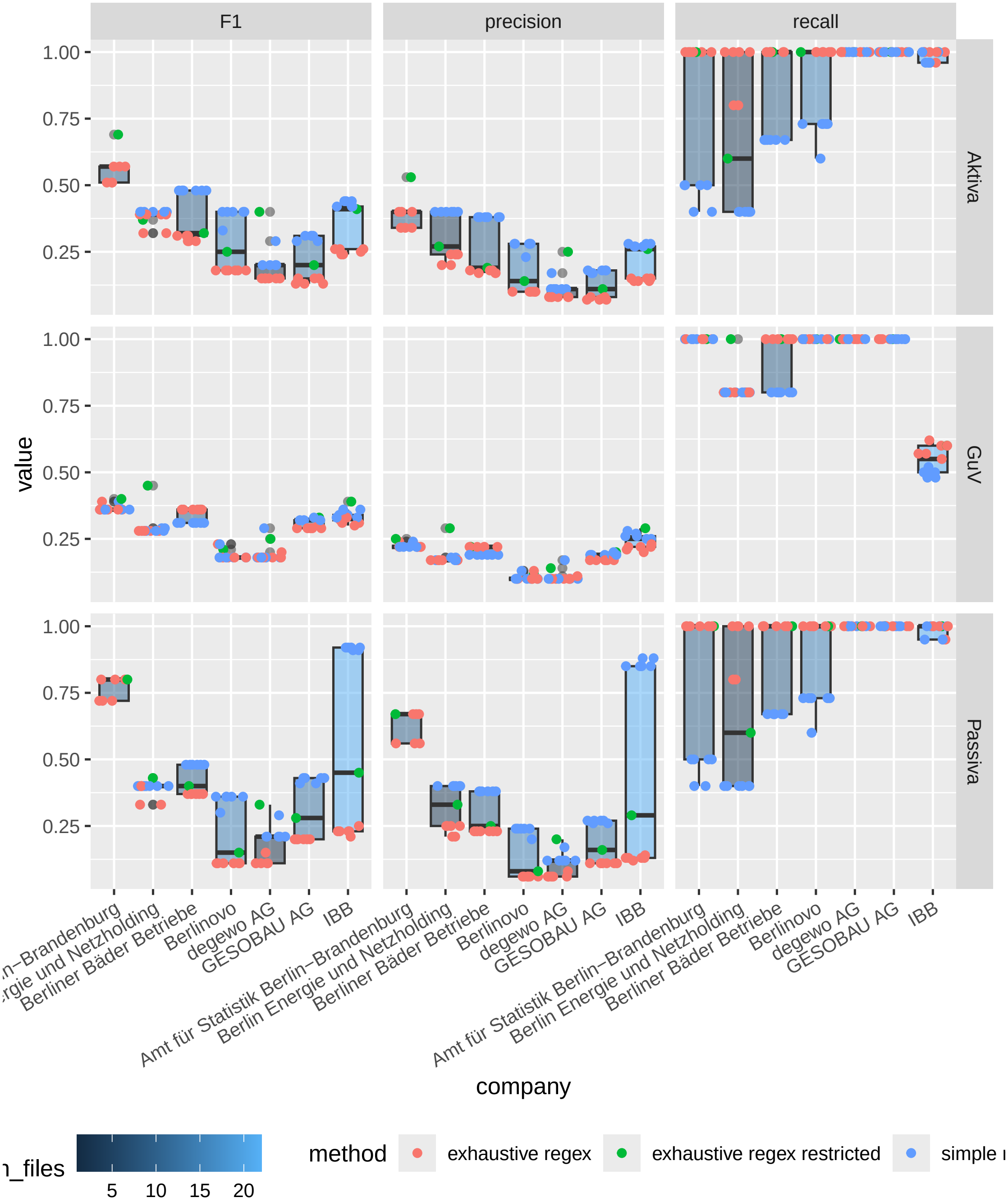
method	stat	precision	recall	F1
simple regex	mean	0.173	0.752	0.283
simple regex	sd	0.008	0.004	0.008
exhaustive regex restricted	mean	{0.21}	{0.83}	{0.34}
exhaustive regex restricted	sd	NA	NA	NA
exhaustive regex	mean	0.172	0.815	0.282
exhaustive regex	sd	0.004	0.01	0.004



Results by company?

```
## Warning: Removed 24 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

```
## Warning: Removed 24 rows containing missing values or values outside the
## scale range (‘geom_point()‘).
```



5.1.2 Table of Contents understanding

```
## Joining with ‘by = join_by(filepath)‘
## ‘summarise()‘ has grouped output by ’type’, ’benchmark_type’. You
## can override using the '.groups' argument.
```

An optional step for larger documents in Li et al. (2023) framework is to identify the pages of interest based on the table of contents (TOC). This would be more efficent than processing the whole document with an LLM. The TOC in a PDF can be given explicit and machine readable or it can be presented in form of text on any page. Of course it can be missing completely as well.

- For a lot of short annual reports one can find the tables of interest within the first eight pages as well.
- calculate and add Qwen, Gemini or LLama results? <– No time!

5.1.2.1 Text based

Li et al. (2023) used the table of contents 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 that "ACFRs typically spans no more than the initial 165 lines of the converted document" (p. 20), they use the first 200 lines of text.

My expectation was to find the TOC within the first five pages. Often we find way less than 200 lines of text on the five first pages (see Figure 5.1). Some files are not machine readable without OCR and thus show zero lines in the first five pages as well.

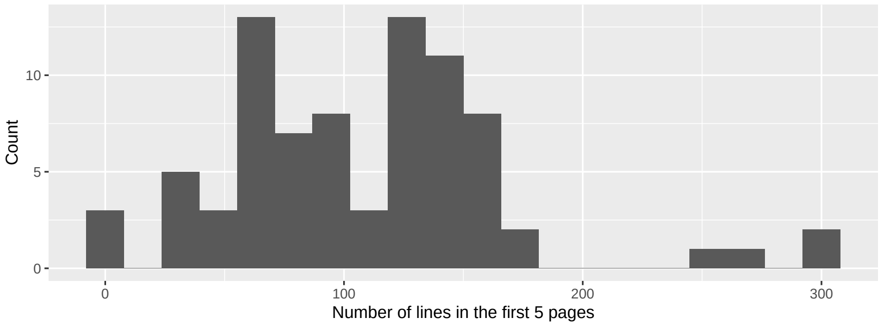


Figure 5.1: Histogram of the number of lines in the first 5 pages of the annual reports

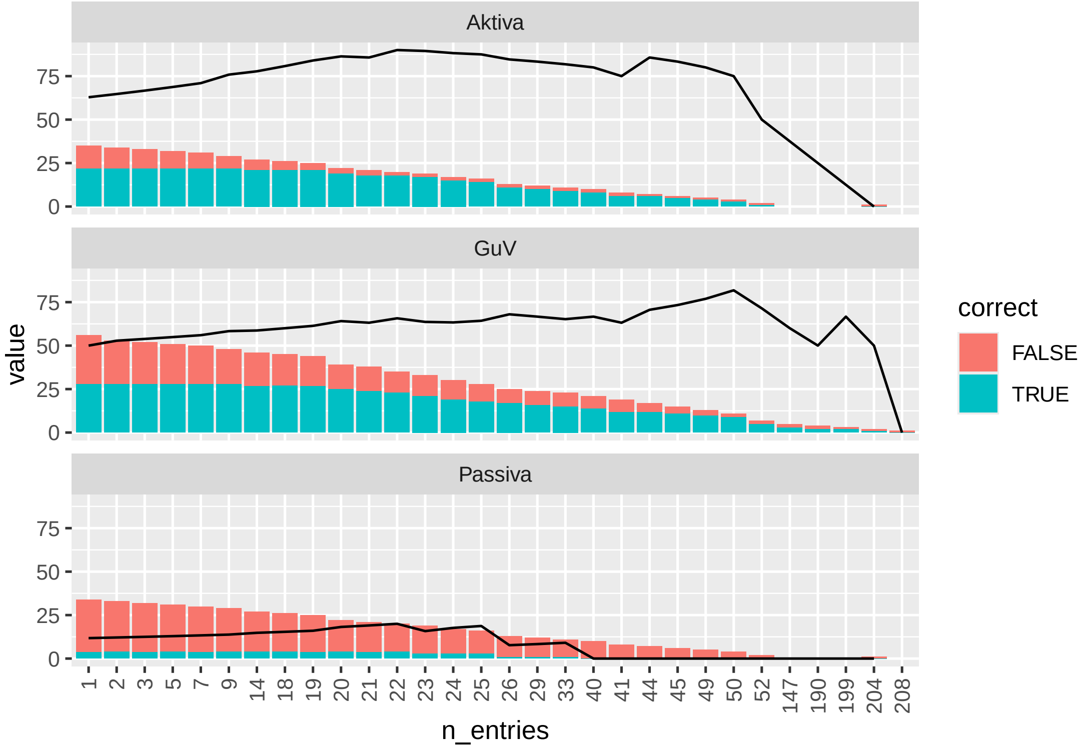
5.1.2.1.1 First five pages A request to Mistral results in 63 strings that should represent a table of contents among the first five pages [strings not checked in detail].

5.1.2.1.2 First 200 lines A request to Mistral results in 68 strings that should represent a table of contents among the first five pages [strings not checked in detail].

5.1.2.1.3 Machine readable TOC based To limit the text and hopefully increase the quality of the input data one can work with the TOC representation embedded within the PDF files. From 80 annual reports 43 files do have a machine readable separate table of contents and 37 do not have one.

One can see that correct predictions for the page range are more probable when the TOC has a medium number of entries. It is possible to drop PDFs with less than 9 without loosing a single correct prediction. This means that for PDFs with TOC with less then 9 entieres the LLM was not able to make a correct prediction. This is not surprising since neither *Bilanz* nor *Gewinn- und Verlustrechnung* are mentioned there.

Almost no influence if TOC is passed formated as markdown or json. With the json formated TOC it found two more correct page ranges (single test run). It was testes because the relation *page_number* heading and value might have been clearer in json for a linear working LLM.



5.1.2.2 Comparison of the different approaches

- toc analysis
- cleaned measures

The LLM performed best on the machine readable TOC. It resulted in highest ratio of correct page ranges as well as in highest absolute numbers even though there were least available TOC.

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. Since the prompt for the LLM was not elaborated enough to cover cases, where there are multiple tables of interest for a single type that are not placed on concurrent pages, one could argue to drop those files from the analysis. This does not change the results significantly, since there are only few files with more than one table of interest per type.

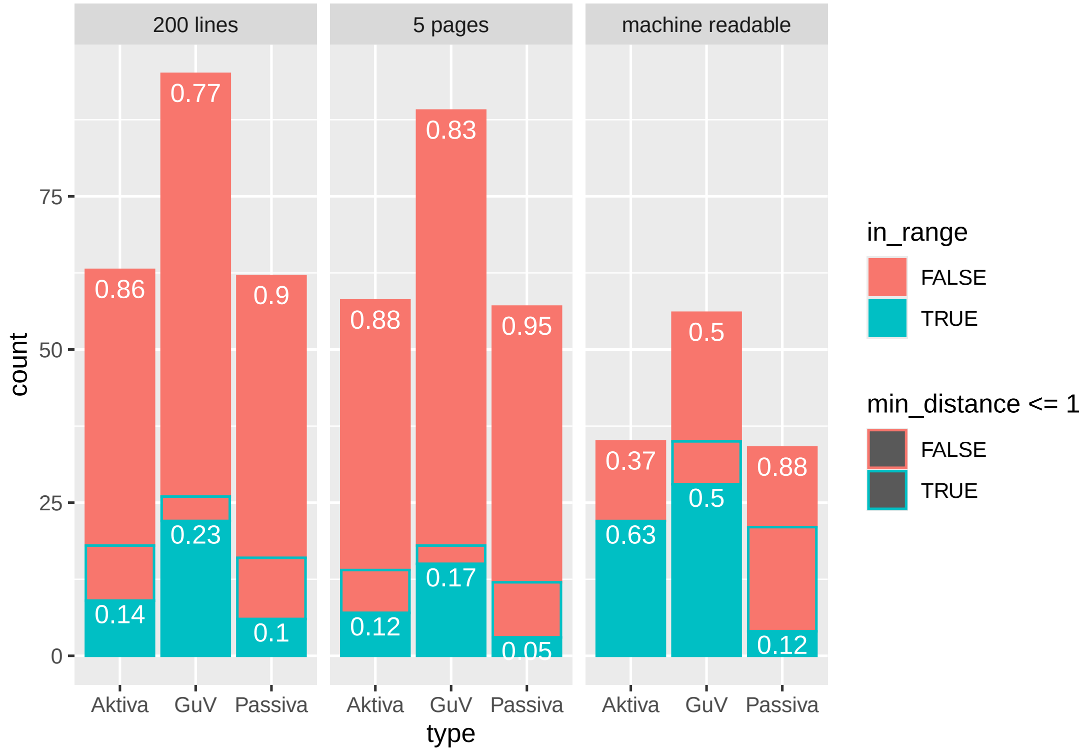
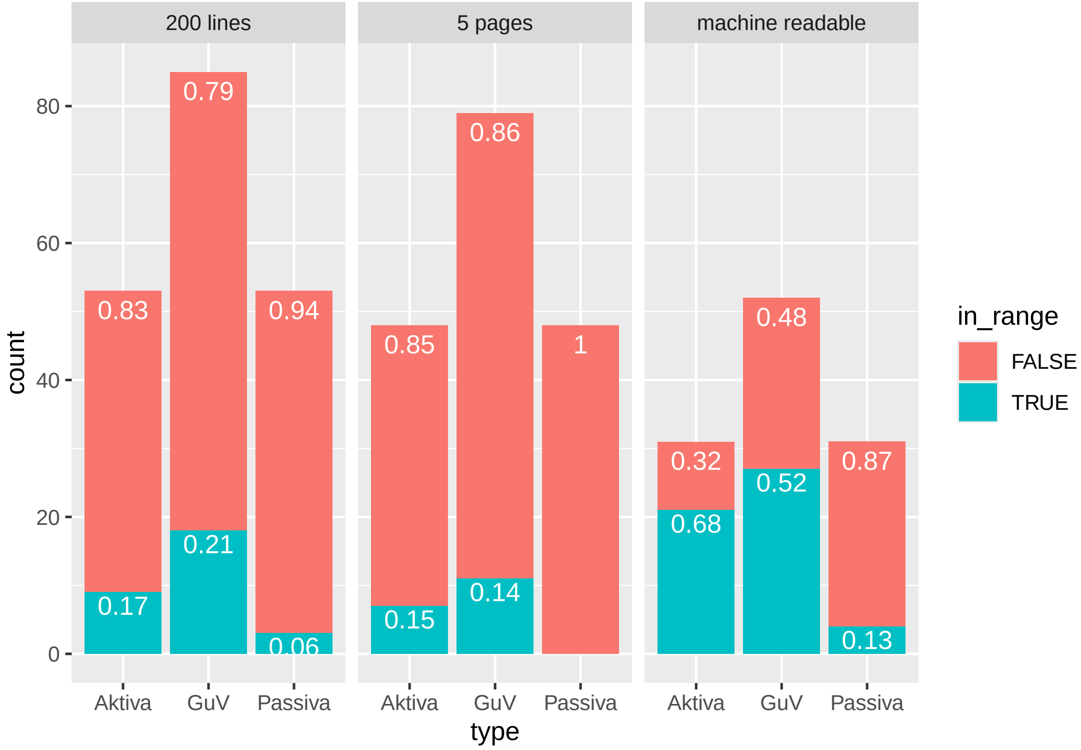


Figure 5.2: Comparing number of fount TOC and amount of correct and incorrect predicted page ranges

benchmark_type	mean range	SD range
5 pages	{2.35}	{1.79}
200 lines	2.78	2.33
machine readable	2.81	4.35



Besides a single group that was predicted far off for the machine readable TOC approach the LLM reported higher confidence for the correct page ranges and got the ranges less far off. But it did not predict the smallest ranges.

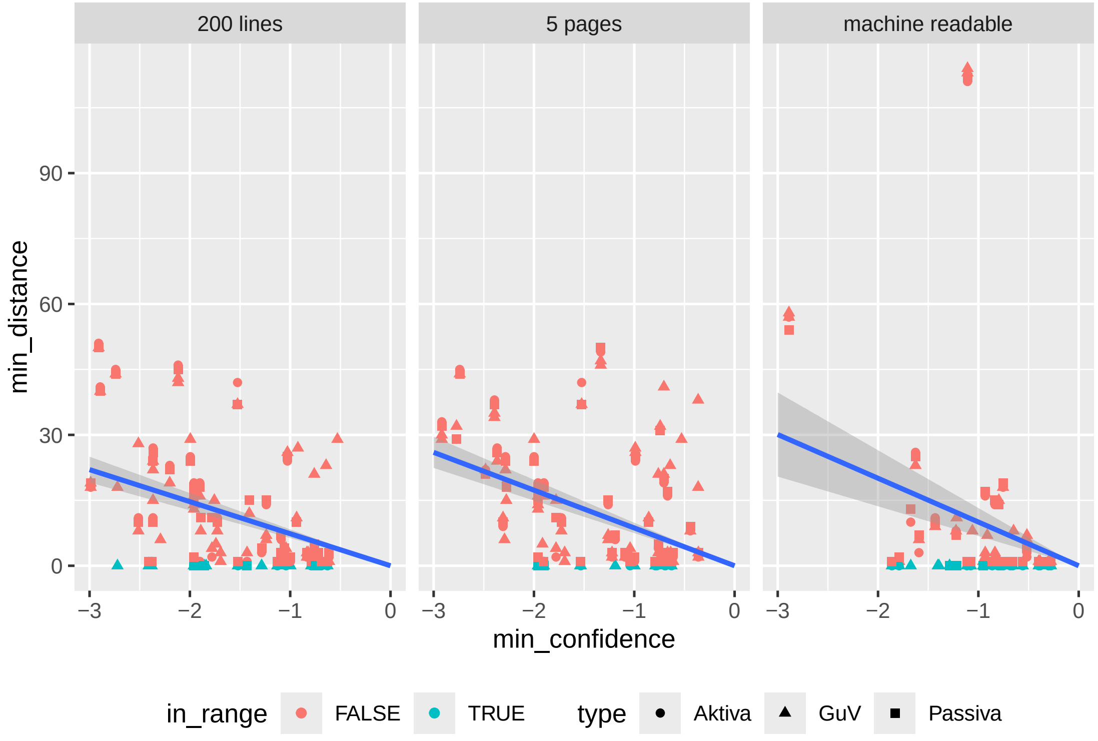
```
mean_ranges %>%
  mutate_if(
    is.numeric,
    ~ifelse(
      . == min(., na.rm = TRUE),
      paste0("**", ., "**"),
      .
    )
  ) %>% arrange(`mean range`) %>%
  render_table()
```

```
## Warning: Removed 78 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).
```

Table 5.4: Comparing GPU time for page range prediction and table of contents extraction

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

```
## Warning: Removed 78 rows containing missing values or values outside the
## scale range (‘geom_point()‘).
```



In general the LLM performed worst to identify the correct page range for *Passiva*. The median distance is one page bigger than for *Aktiva* and *Gewinn- und Verlustrechnung*. This makes sense for *Aktiva* because the *Passiva* is often on the next page but the predicted page range for *Aktiva* and *Passiva* are often identical. Furthermore the predicted page range for *Aktiva* is often only a single page wide. Thus the *Passiva* on the next page is not inside the predicted page range.

This problem was solved by explicitly mentioning that assets and liabilities are both part of the balance sheets for the five pages and 200 lines approach but not for the machine readable TOC one.

A pragmatic way would be to use the machine readable TOC approachs prediction for the *Aktiva* page range and add one to the end page to get the *Passiva* page range. Beside the problem to predict a correct page range for *Passiva* the machine readable TOC approach was very effective and is also pretty efficient if one counts in the effort the LLM driven TOC extraction takes.

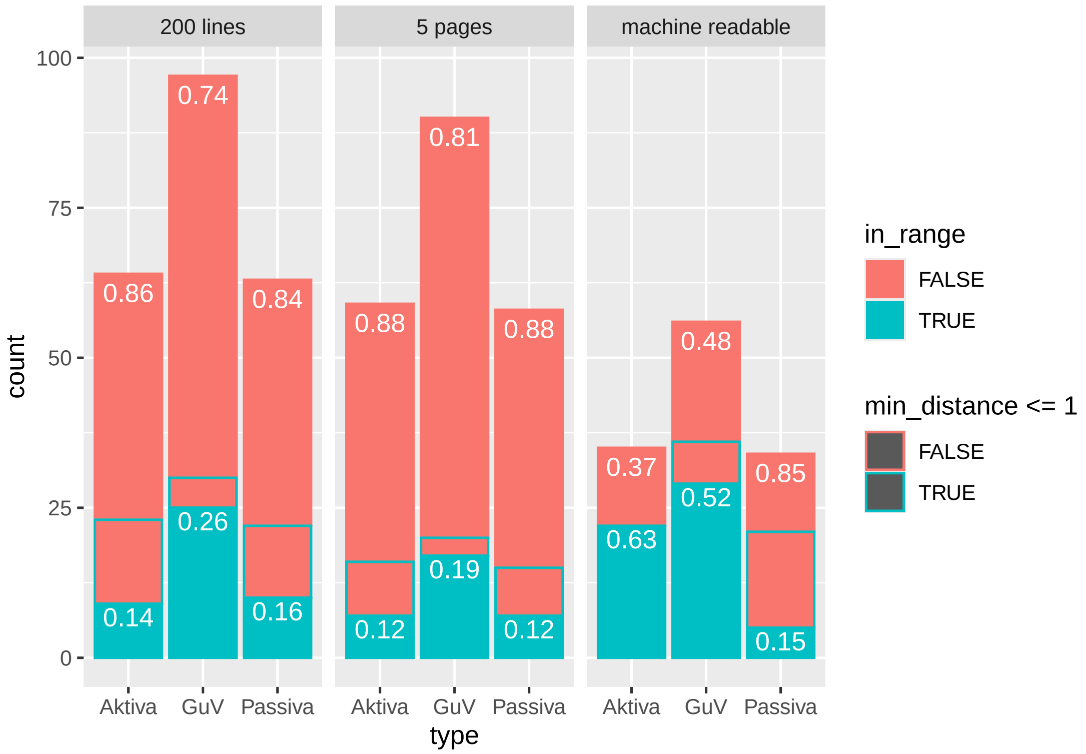


Figure 5.3: Comparing number of fount TOC and amount of correct and incorrect predicted page ranges

5.1.3 Classification with LLMs

structured outputs forcing to answer with a *yes* or *no* for binary task or with *Aktiva*, *Passiva*, *GuV* or *other* for multi classification task

top n accuracy

out of company vs in compnay rag

5.1.3.1 Binary classification

Could be more efficient to predict “is any of interest” and then which type, because dataset is highly imbalanced.

24 models from 6 haven been benchmarked among 5 methods

Most models have been used up to 3 examples for the context.

The best combination of model and method for each method family is presented in the following table. It is clear that the Google Gemma models are performing worst.[1] Surprisingly Mistral 2410 is the best performing model for all three prediction tasks even though it only has 8B parameters.

```
## ‘mutate_all()‘ ignored the following grouping variables:
## ‘mutate_all()‘ ignored the following grouping variables:
## * Columns ‘model_family‘, ‘classification_type‘
## i Use ‘mutate_at(df, vars(-group_cols()), myoperation)‘ to silence
##   the message.
```

It is interesting that the predictions do not get better by providing more and more examples. Especially for the n-rag-example approach we find a significant drop in the F1 score if the examples pages come from different companies annual reports. This is caused by a sever recall drop. But also for the n-ranom-example approach we see this for the prediction of class Passiva.

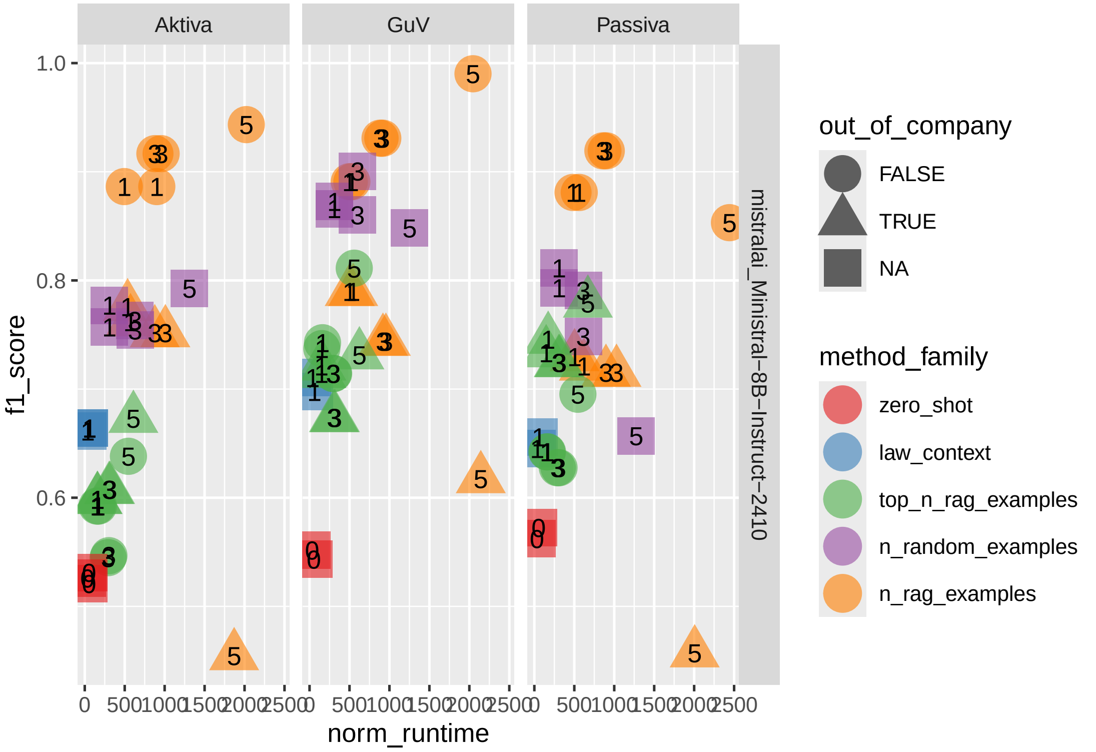
Recall better with examples from same company. Precision better without.

We can also see that the prediction performance is stable.[2]

[1] 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.

[2] 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.

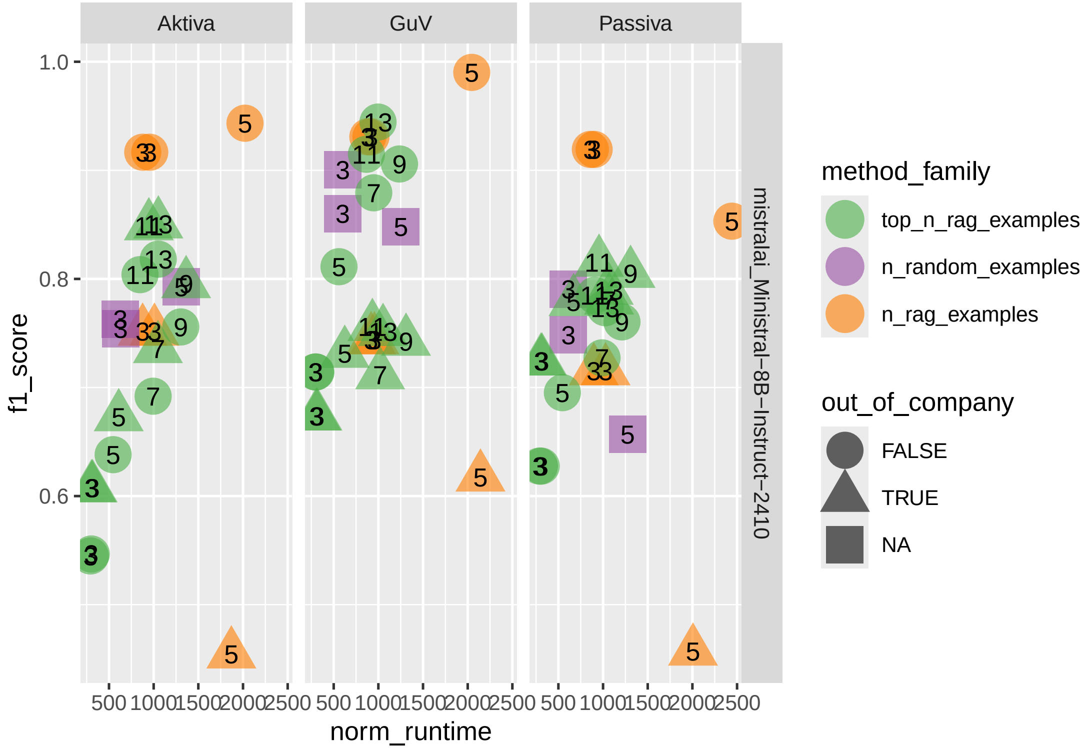
model_family	model	classification_type	method_family	n_examples
mistralai	mistralai_Ministral8BInstruct2410	GuV	n_rag_examples	3
meta-llama	metallama_Llama4Scout17B16EInstruct	GuV	n_rag_examples	3
mistralai	mistralai_Ministral8BInstruct2410	Passiva	n_rag_examples	3
mistralai	mistralai_Ministral8BInstruct2410	Aktiva	n_rag_examples	3
Qwen	Qwen_Qwen2.532BInstruct	GuV	n_rag_examples	1
meta-llama	metallama_Llama4Scout17B16EInstruct	Passiva	n_rag_examples	3
Qwen	Qwen_Qwen2.532BInstruct	Aktiva	n_rag_examples	1
Qwen	Qwen_Qwen3235BA22BInstruct2507	Aktiva	n_rag_examples	3
meta-llama	metallama_Llama4Scout17B16EInstruct	Aktiva	n_rag_examples	3
Qwen	Qwen_Qwen2.532BInstruct	Passiva	n_rag_examples	1
microsoft	microsoft_phi4	Aktiva	law_context	1
microsoft	microsoft_phi4	Passiva	law_context	1
google	google_gemma327bit091	Passiva	n_rag_examples	1
google	google_gemma327bit091	Aktiva	n_rag_examples	1
tiiuae	tiiuae_Falcon310BInstruct	Passiva	n_random_examples	1
google	google_gemma327bit091	GuV	n_rag_examples	1
tiiuae	tiiuae_Falcon310BInstruct	Aktiva	n_rag_examples	1
tiiuae	tiiuae_Falcon310BInstruct	GuV	top_n_rag_examples	1



- f1
- multiple models

- best model detail (different methods / settings)

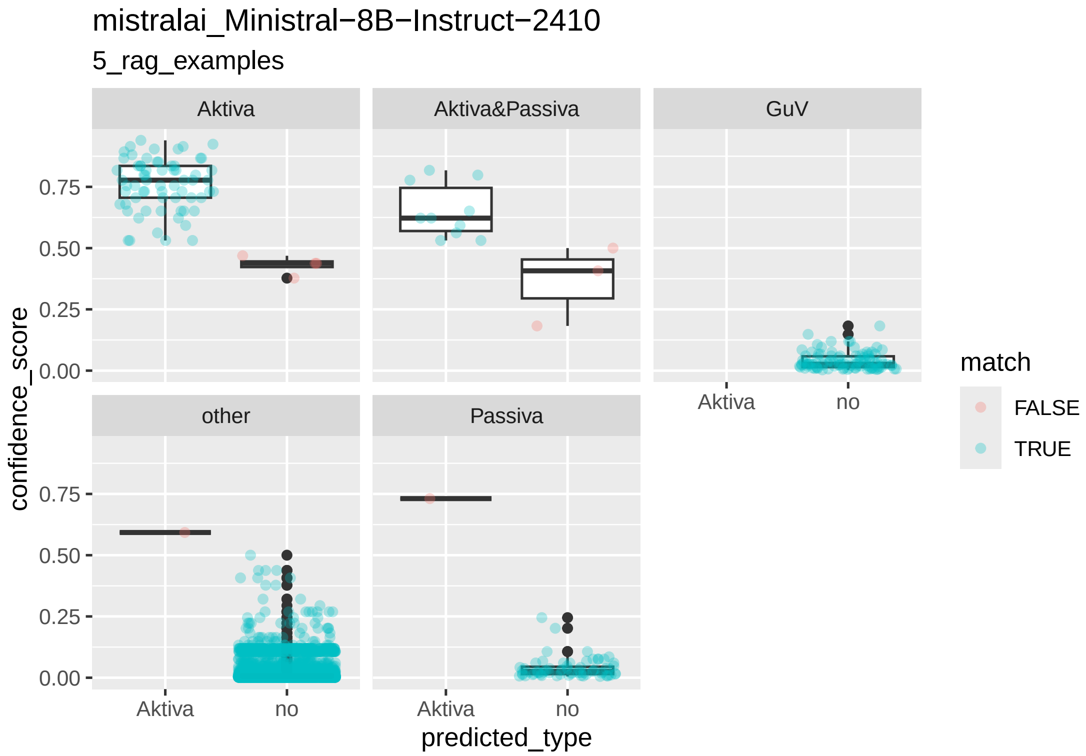
The experiments for the best performing model, Ministral-8B-Instruct-2410, have been extended by methods with even more examples. Especially for the top-n-rag-example approach to get a better comparable picture based on the real number of examples / context length.



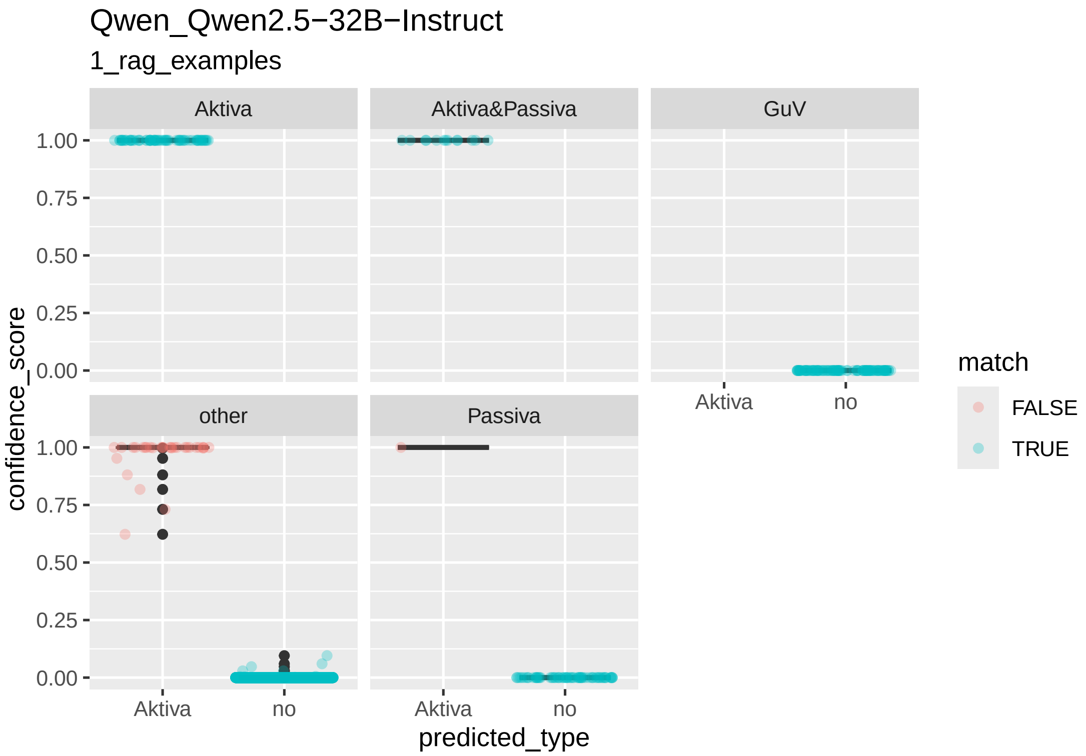
```
## Rows: 26 Columns: 6
## -- Column specification --------------------------------------------
## Delimiter: ","
## chr (2): approach, classification
## dbl (4): n_example, target, other, sum
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

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

Predictions very accurate. Confidence not always 1. Wrong predictions with often with medium confidence. If Aktiva and Passiva on same page more often Aktiva predicted. Confidence for no displayed as 1-confidence to represent confidence for yes (binary classification).



Qwen returns always high confidence even if it is wrong.

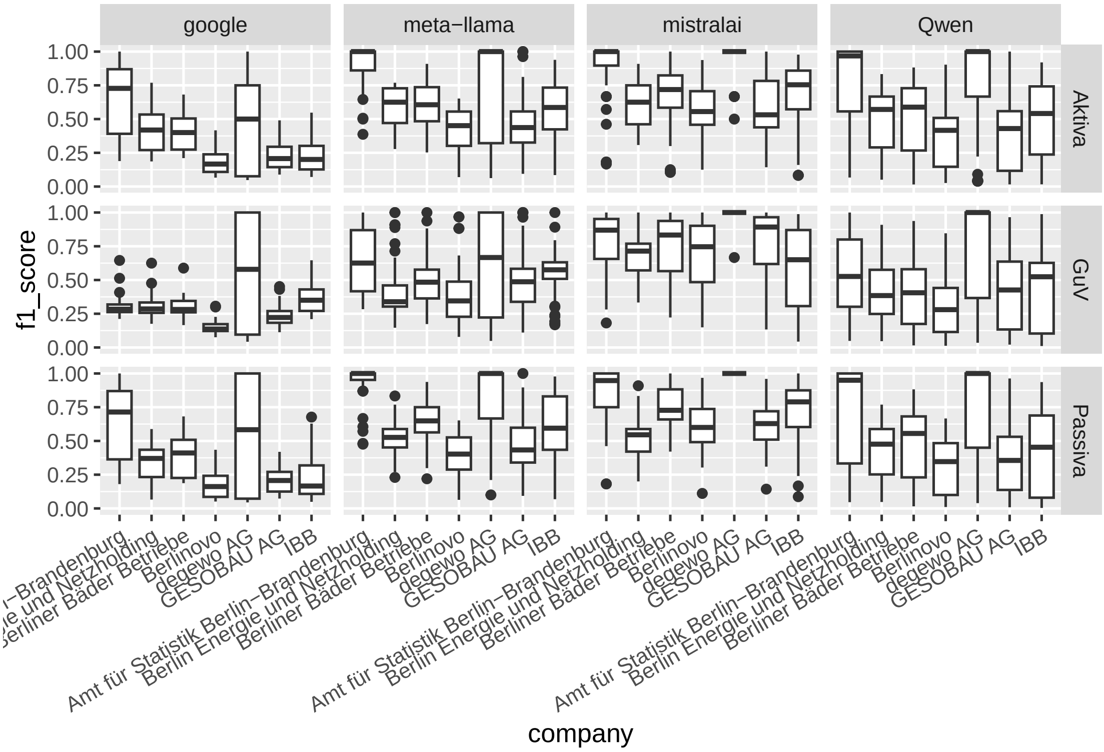


- IBB other law
- degewo only one where no ocr is needed

mistral: recall IBB and Netzholding big range meta & mistral: very high precision for Amt für Statistik BBB <- lowest average pagecount (29.3) but IBB has more pages than berlinovo but better precision. No information about which company / report the page is from

```
## Rows: 252 Columns: 3
## -- Column specification --------------------------------------------
## Delimiter: ","
## chr (2): filepath, type
## dbl (1): page
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
## Warning: Removed 246547 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



```
n_reports_by_company_no_ocr <- df_temp %>% select(company, filepath) %>% unique() %>% group_by(company) %
n_reports_by_company <- df_filtered %>% unnest(predictions) %>% select(company, filepath) %>% unique() %>%

n_reports_by_company_no_ocr %>% kbl()
```

company	n
Amt für Statistik Berlin-Brandenburg	10
Berlin Energie und Netzholding	3
Berliner Bäder Betriebe	10
Berlinovo	15
GESOBAU AG	13
IBB	22
degewo AG	1

```
n_reports_by_company %>% kbl()
```

company	n
Amt für Statistik Berlin-Brandenburg	10
Berlin Energie und Netzholding	3
Berliner Bäder Betriebe	10
Berlinovo	15
GESOBAU AG	13
IBB	22
degewo AG	7

- Performance makes a jump at a critical parameter number (3B) then slow increase (compare Qwen 2.5)
- Changes unsystematic with new models (see Mistral, Qwen 3 old vs llama 4)

PR curves for all classes look very alike- showing micro average curve

```
## Loading required package: rlang
```

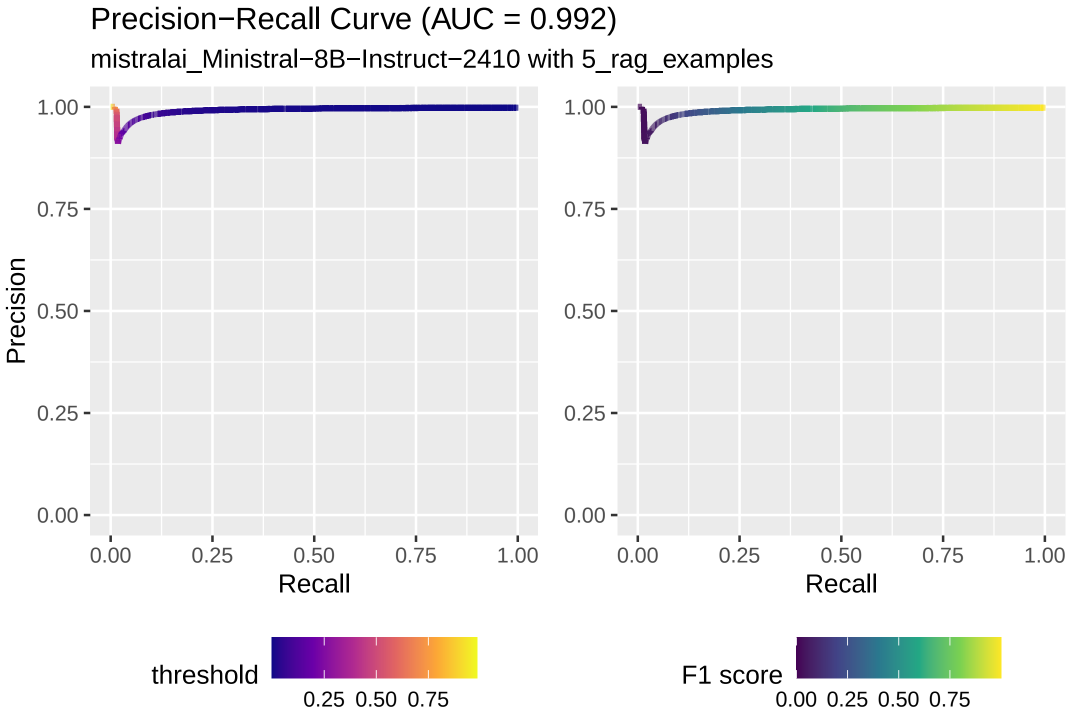
```
##
## Attaching package: 'rlang'
```

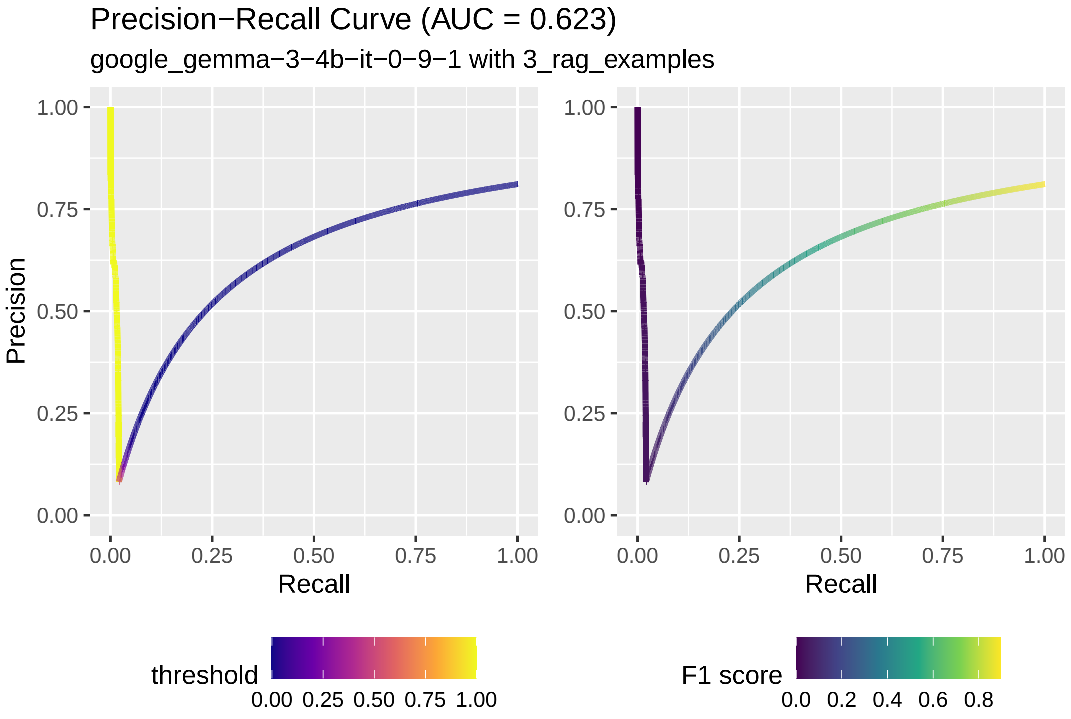
```
## The following objects are masked from 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice
```

```
## The following objects are masked from 'package:jsonlite':
##
##     flatten, unbox
```

```
## Warning: Using one column matrices in 'filter()' was deprecated in dplyr
## 1.1.0.
## i Please use one dimensional logical vectors instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this
## warning was generated.
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this
## warning was generated.
```


Precision−Recall Curve (AUC = 0.992)
mistralai_Ministral−8B−Instruct−2410 with 5_rag_examples
Precision
1.00
0.75
0.50
0.25
0.00
Recall
0.00
0.25
0.50
0.75
1.00
threshold
0.25 0.50 0.75
F1 score
0.00 0.25 0.50 0.75


Precision−Recall Curve (AUC = 0.623)
google_gemma−3−4b−it−0−9−1 with 3_rag_examples
Precision
1.00
0.75
0.50
0.25
0.00
Recall
0.00
0.25
0.50
0.75
1.00
threshold
0.00 0.25 0.50 0.75 1.00
F1 score
0.0 0.2 0.4 0.6 0.8

model_family	model	metric_type	method_family	n_examples	f1_sc
meta-llama	metallama_Llama4Scout17B16EInstruct	GuV	n_rag_examples	1	1
meta-llama	metallama_Llama4Scout17B16EInstruct	Aktiva	n_rag_examples	3	0.99
mistralai	mistralai_MistralLargeInstruct2411	Passiva	n_rag_examples	1	0.99
meta-llama	metallama_Llama4Scout17B16EInstruct	Passiva	n_rag_examples	3	0.99
mistralai	mistralai_MistralLargeInstruct2411	Aktiva	n_rag_examples	3	0.97
Qwen	Qwen_Qwen2.532BInstruct	Aktiva	n_rag_examples	3	0.97
Qwen	Qwen_Qwen3235BA22BInstruct2507	GuV	n_rag_examples	3	0.97
mistralai	mistralai_MistralLargeInstruct2411	GuV	n_rag_examples	1	0.95
mistralai	mistralai_MistralLargeInstruct2411	GuV	n_rag_examples	3	0.95
Qwen	Qwen_Qwen2.572BInstruct	Passiva	n_rag_examples	1	0.95
google	google_gemma327bit091	Aktiva	n_rag_examples	3	0.88
google	google_gemma327bit091	Passiva	n_rag_examples	1	0.81
google	google_gemma327bit091	GuV	n_rag_examples	1	0.78
tiiuae	tiiuae_Falcon310BInstruct	GuV	n_rag_examples	1	0.7
microsoft	microsoft_phi4	Passiva	n_rag_examples	1	0.66
microsoft	microsoft_phi4	Aktiva	n_random_examples	1	0.59
tiiuae	tiiuae_Falcon310BInstruct	Aktiva	n_rag_examples	1	0.55
tiiuae	tiiuae_Falcon310BInstruct	Passiva	n_rag_examples	1	0.55
microsoft	microsoft_phi4	GuV	n_rag_examples	1	0.45

5.1.3.2 Multi classification

bigger models are better with the multi classification task Llama-4-Scout almost perfect F1 for all classes

Llama-4-Scout runs fast but needs long to load because it has 109B in total with 17B actives Gemma performs much better than with binary classification

drop with Qwen-14B

```
## ‘mutate_all()‘ ignored the following grouping variables:
## ‘mutate_all()‘ ignored the following grouping variables:
## * Columns ‘model_family‘, ‘metric_type‘
## i Use ‘mutate_at(df, vars(-group_cols()), myoperation)‘ to silence
##   the message.
```

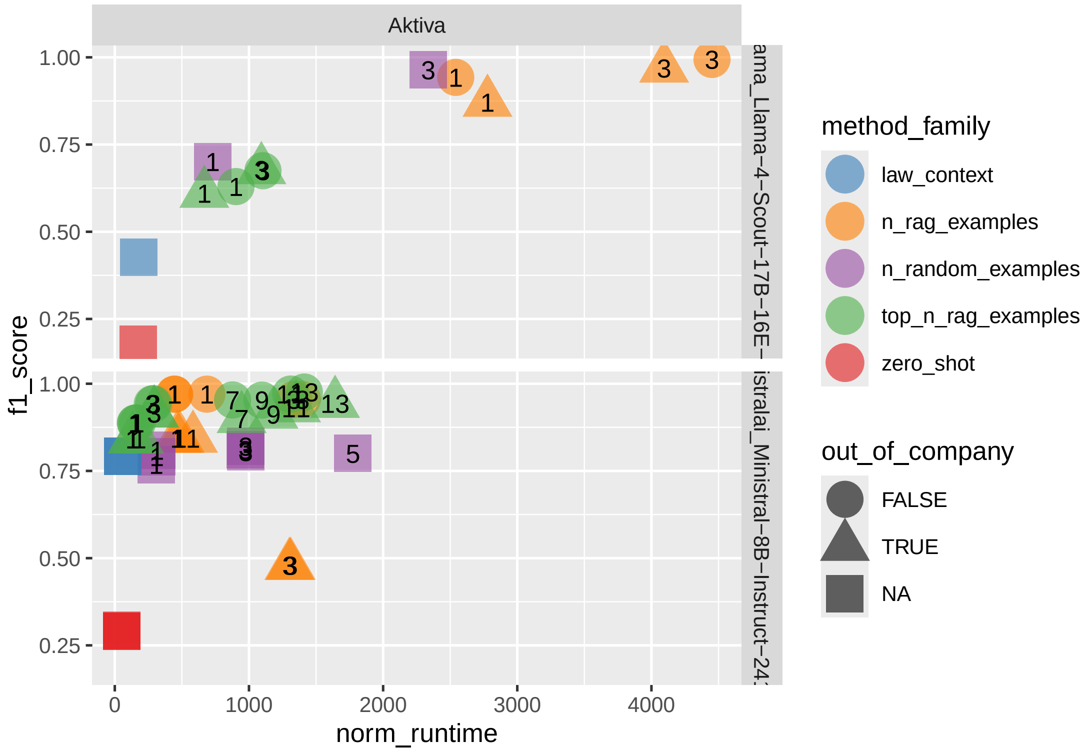
Mistral-8B-2410 almost as good as Mistral-123B-2411 but much faster

```
## ‘mutate_all()‘ ignored the following grouping variables:
## ‘mutate_all()‘ ignored the following grouping variables:
## * Columns ‘model_family‘, ‘metric_type‘
## i Use ‘mutate_at(df, vars(-group_cols()), myoperation)‘ to silence
##   the message.
```

Mistral-2410 reaches good performance already with few examples and can work with law-context approach but more examples don’t realy help any further

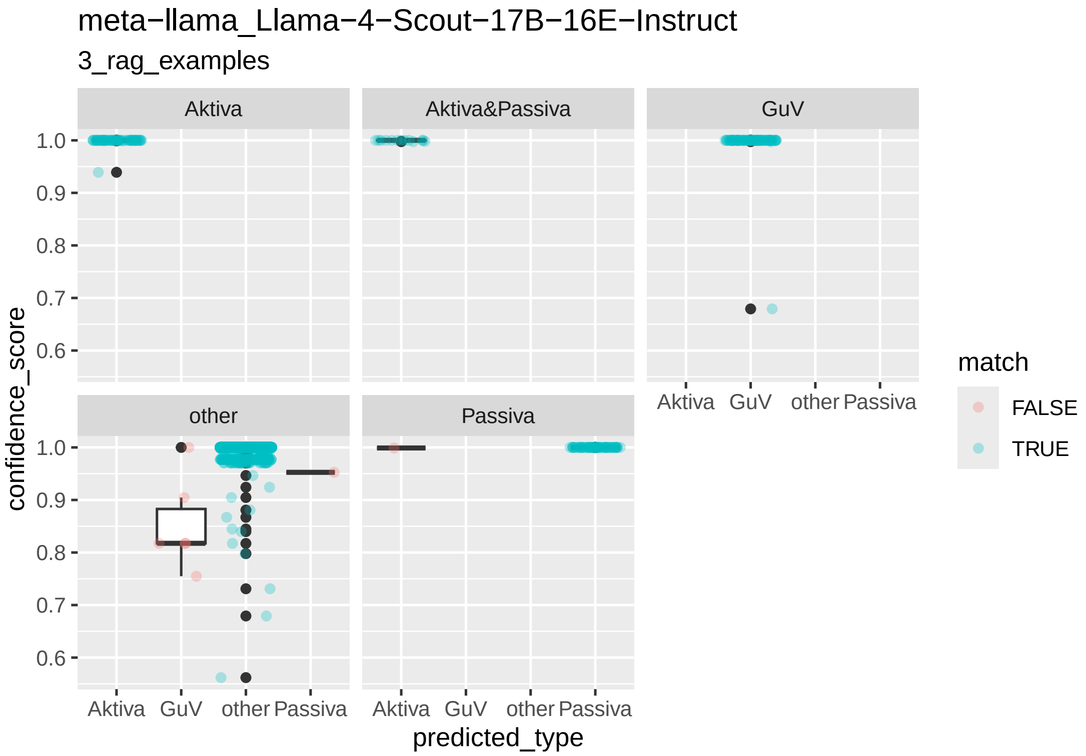
```
## Warning: Removed 8 rows containing missing values or values outside the scale
## range (‘geom_text()‘).
```

model_family	model	metric_type	method_family	n_examples	f1_score	runtime
mistralai	mistralai_Ministral8BInstruct2410	Aktiva	n_rag_examples	1	0.97	686
mistralai	mistralai_Ministral8BInstruct2410	Passiva	n_rag_examples	1	0.95	686
mistralai	mistralai_Ministral8BInstruct2410	GuV	n_rag_examples	3	0.95	1399
mistralai	mistralai_Ministral8BInstruct2410	GuV	top_n_rag_examples	3	0.95	279
meta-llama	metallama_Llama3.18BInstruct	Passiva	n_rag_examples	1	0.94	593
Qwen	Qwen_Qwen2.53BInstruct	Aktiva	n_rag_examples	1	0.86	492
meta-llama	metallama_Llama3.18BInstruct	Aktiva	top_n_rag_examples	3	0.85	269
google	google_gemma312bit091	Aktiva	n_rag_examples	3	0.84	2733
Qwen	Qwen_Qwen2.53BInstruct	Passiva	law_context	NA	0.81	28
Qwen	Qwen_Qwen2.53BInstruct	GuV	n_rag_examples	1	0.76	492
tiiuae	tiiuae_Falcon310BInstruct	GuV	n_rag_examples	1	0.7	868
google	google_gemma312bit091	Passiva	n_rag_examples	1	0.68	1259
meta-llama	metallama_Llama3.18BInstruct	GuV	n_rag_examples	1	0.62	593
meta-llama	metallama_Llama3.18BInstruct	GuV	top_n_rag_examples	1	0.62	205
tiiuae	tiiuae_Falcon310BInstruct	Aktiva	n_rag_examples	1	0.55	868
tiiuae	tiiuae_Falcon310BInstruct	Passiva	n_rag_examples	1	0.55	868
google	google_gemma312bit091	GuV	top_n_rag_examples	1	0.46	232



Most of the time pretty confident most problems with class “other” If Aktiva and Passiva on same page it predicts Aktiva. Also one Passiva missclassified as Aktiva No flipped confidence [3]

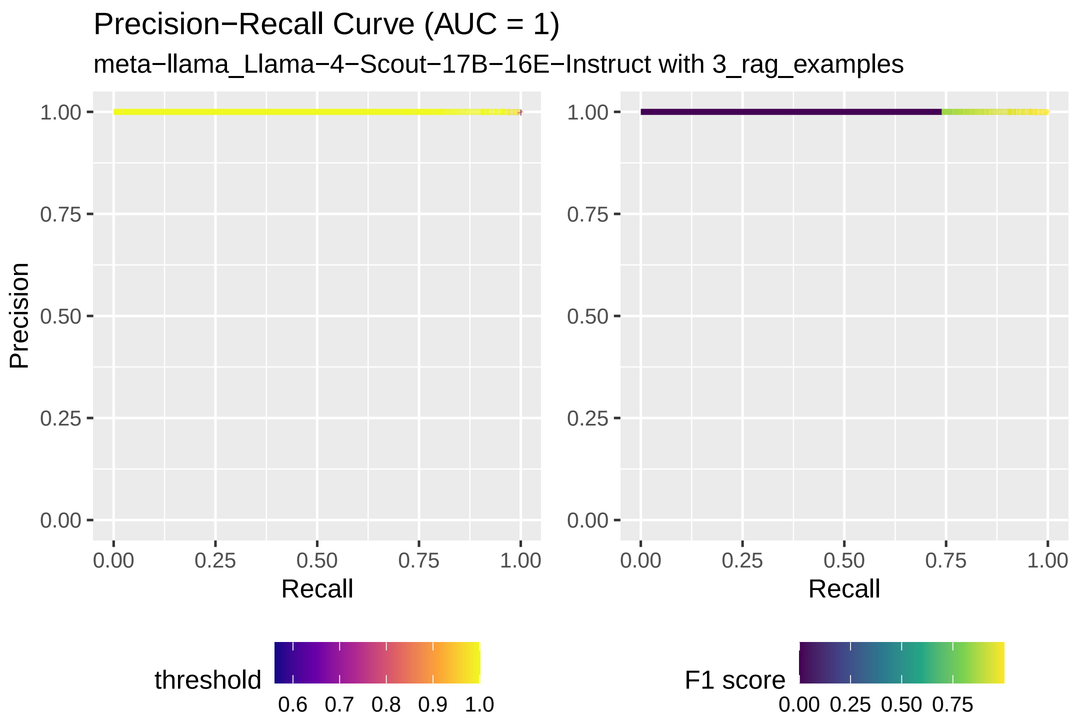
[3] classify framework in needs special models with pooling capability. Would have been interesting but time was limited and

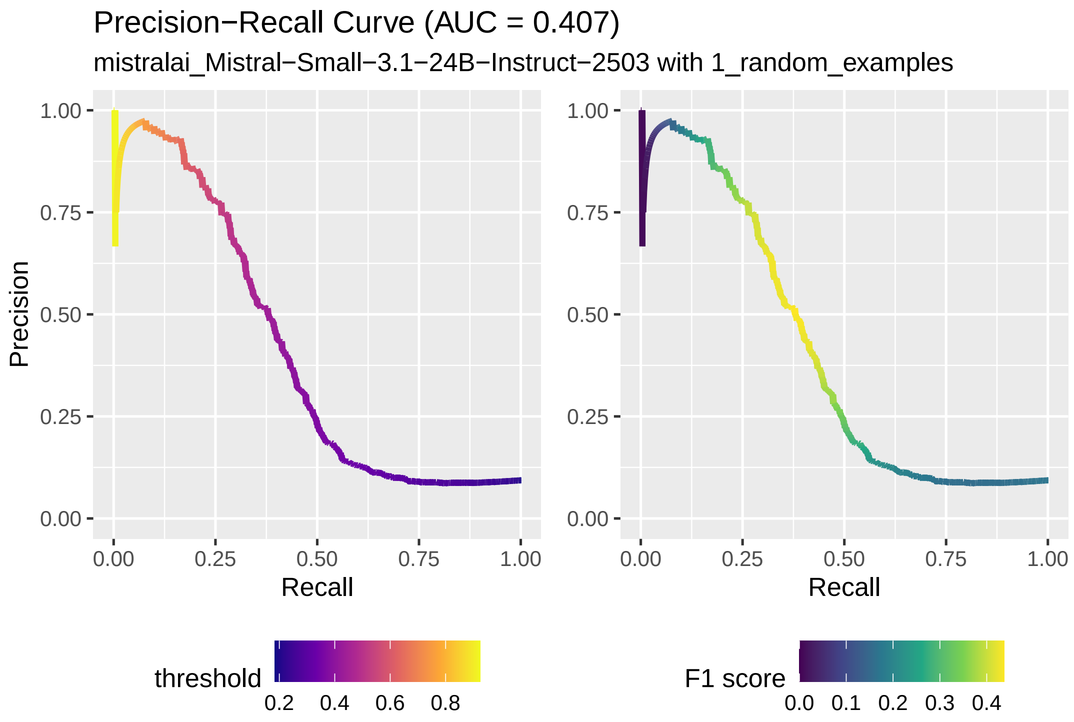


Microsoft phi 4 and Falcon 3 only ran with one and two examples because their context window is smaller.

- f1
- multiple models
- best model detail (different methods / settings)

would have needed new special models in most cases


Precision−Recall Curve (AUC = 1)
meta−llama_Llama−4−Scout−17B−16E−Instruct with 3_rag_examples
Precision
1.00
0.75
0.50
0.25
0.00
Recall
0.00
0.25
0.50
0.75
1.00
threshold
0.6
0.7
0.8
0.9
1.0
F1 score
0.00
0.25
0.50
0.75


Precision−Recall Curve (AUC = 0.407)
mistralai_Mistral−Small−3.1−24B−Instruct−2503 with 1_random_examples
Precision
1.00
0.75
0.50
0.25
0.00
Recall
0.00
0.25
0.50
0.75
1.00
threshold
0.2
0.4
0.6
0.8
F1 score
0.0
0.1
0.2
0.3
0.4

5.1.4 Term frequency based classifier

RandomForest performs much better than a logistic regression Better results with * undersampling * training on all types simultaniousely

5.1.4.1 Two predictors

Term frequency of nouns of the law about Aktiva Float freqency (floats divided by word count)

5.1.4.2 Four predictors

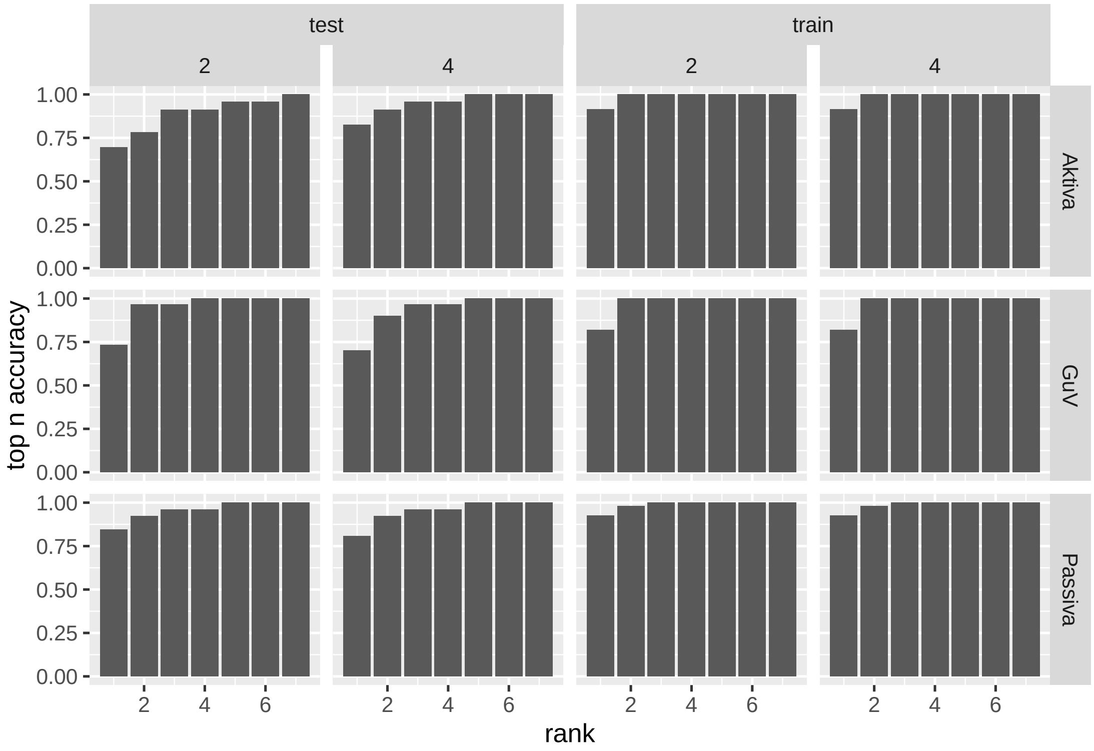
Count of integers Count of dates

- top 1
- top k

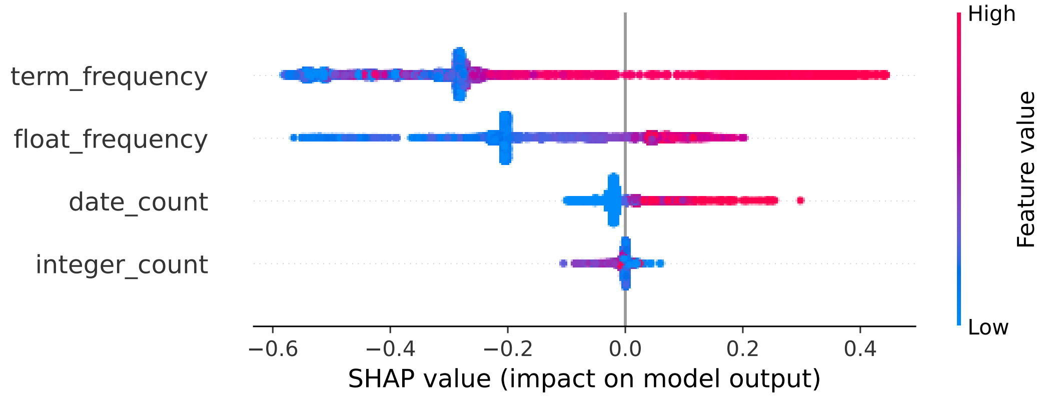
low precision llm linked to position of correct page? numeric frequency?

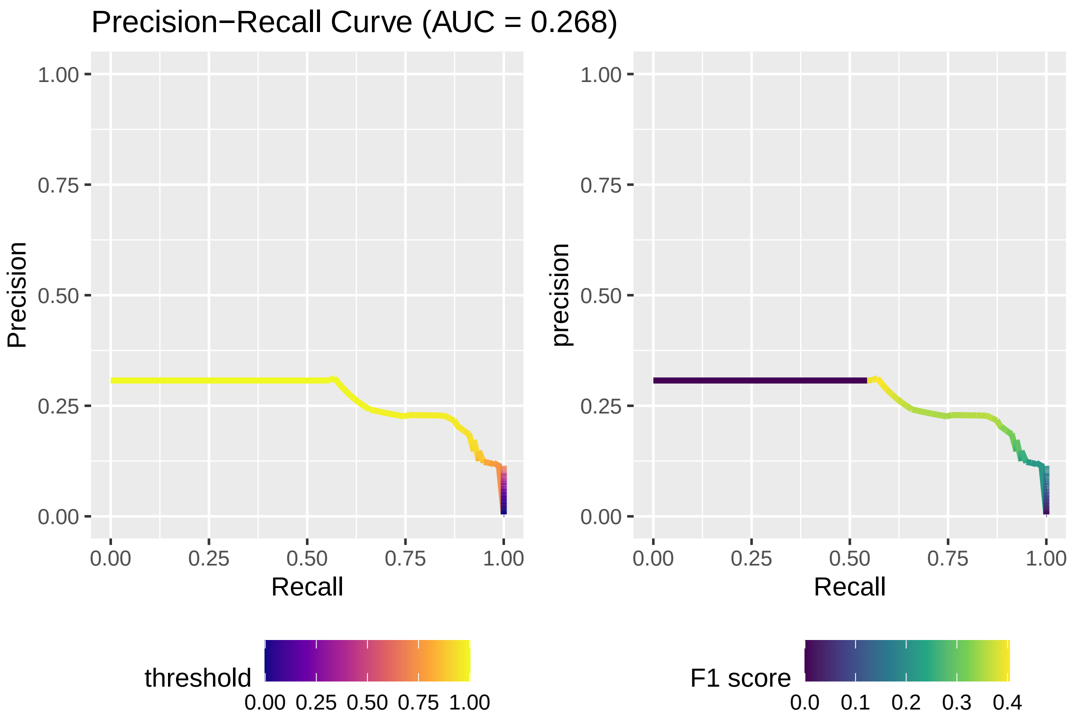
```python
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

df_train_us = pd.read_csv("../benchmark_results/page_identification/term_frequency_table.csv")

# Drop rows without ground truth
# df_train_us = df_word_counts.merge(df_truth, on=["filepath", "type"], how="left")
df_train_us["is_truth"] = (df_train_us["page"] == df_train_us["page_truth"]).astype(int)
df_train_us = df_train_us.dropna(subset=["page_truth"])

# Undersample the majority class (is_truth == 0)
df_true = df_train_us[df_train_us["is_truth"] == 1]
df_false = df_train_us[df_train_us["is_truth"] == 0]
df_false_undersampled = df_false.sample(n=len(df_true), random_state=42)
df_train_us_balanced = pd.concat([df_true, df_false_undersampled]).sample(frac=1, random_state=42)
# df_train_us_balanced

# Features and target
X = df_train_us_balanced[["term_frequency", "float_frequency"]].values
y = df_train_us_balanced["is_truth"].values

# Train-test split (70% train, 30% test)
X_train, X_test, y_train, y_test, df_train_split, df_test_split = train_test_split(
    X, y, df_train_us_balanced, test_size=0.3, random_state=42, stratify=y
)
```

```python
# Train Random Forest model
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)
score = clf.score(X_train, y_train)
# print(f"Training accuracy: {score:.2%}")
score = clf.score(X_test, y_test)
# print(f"Test accuracy: {score:.2%}")

# Predict and rerank: get predicted probabilities for each page
df_train_split["score"] = clf.predict_proba(X_train)[:, 1]
df_test_split["score"] = clf.predict_proba(X_test)[:, 1]

# Add all not-chosen negatives from df_false to test split
df_false_unused = df_false.loc[~df_false.index.isin(df_false_undersampled.index)]
df_false_unused = df_false_unused.copy()
df_false_unused["score"] = clf.predict_proba(df_false_unused[["term_frequency", "float_frequency"]].value
df_false_unused["rank"] = np.nan  # Not ranked yet

# Concatenate with test split
df_test_split = pd.concat([df_test_split, df_false_unused], ignore_index=True)

# For each group (filepath, type), sort by score descending
df_train_split["rank"] = df_train_split.groupby(["filepath", "type"])["score"].rank(ascending=False, meth
df_test_split["rank"] = df_test_split.groupby(["filepath", "type"])["score"].rank(ascending=False, method
```



```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

df_train_us = pd.read_csv("../benchmark_results/page_identification/term_frequency_table.csv")

# Drop rows without ground truth
# df_train_us = df_word_counts.merge(df_truth, on=["filepath", "type"], how="left")
df_train_us["is_truth"] = (df_train_us["page"] == df_train_us["page_truth"]).astype(int)
df_train_us = df_train_us.dropna(subset=["page_truth"])

# Undersample the majority class (is_truth == 0)
df_true = df_train_us[df_train_us["is_truth"] == 1]
df_false = df_train_us[df_train_us["is_truth"] == 0]
df_false_undersampled = df_false.sample(n=len(df_true), random_state=42)
df_train_us_balanced = pd.concat([df_true, df_false_undersampled]).sample(frac=1, random_state=42)
# df_train_us_balanced

predictors = [
    "term_frequency",
    "float_frequency",
    "date_count",
    "integer_count"
]

# Features and target
X = df_train_us_balanced[predictors].values # only better with date and integer counts; otherwise worse
y = df_train_us_balanced["is_truth"].values

# Train-test split (70% train, 30% test)
X_train, X_test, y_train, y_test, df_train_split, df_test_split = train_test_split(
    X, y, df_train_us_balanced, test_size=0.3, random_state=42, stratify=y
)

# Train Random Forest model
clf = RandomForestClassifier(n_estimators=100, random_state=42)
dummy = clf.fit(X_train, y_train)

# Predict and rerank: get predicted probabilities for each page
df_train_split["score"] = clf.predict_proba(X_train)[:, 1]
df_test_split["score"] = clf.predict_proba(X_test)[:, 1]

# Add all not-chosen negatives from df_false to test split
df_false_unused = df_false.loc[~df_false.index.isin(df_false_undersampled.index)]
df_false_unused = df_false_unused.copy()
df_false_unused["score"] = clf.predict_proba(df_false_unused[predictors].values)[:, 1]
df_false_unused["rank"] = np.nan  # Not ranked yet

# Concatenate with test split
df_test_split = pd.concat([df_test_split, df_false_unused], ignore_index=True)

# For each group (filepath, type), sort by score descending
df_train_split["rank"] = df_train_split.groupby(["filepath", "type"])["score"].rank(ascending=False, meth
df_test_split["rank"] = df_test_split.groupby(["filepath", "type"])["score"].rank(ascending=False, method
```





5.1.5 Comparison

5.1.5.1 Prediction performance

F1 scores for llms are much higher

5.1.5.2 Energy usage and runtime

Multiclassification more effective than three times single classification Combine term frequency with llm approach to limit page range

5.2 Table extraction

5.2.1 Baseline: Regex

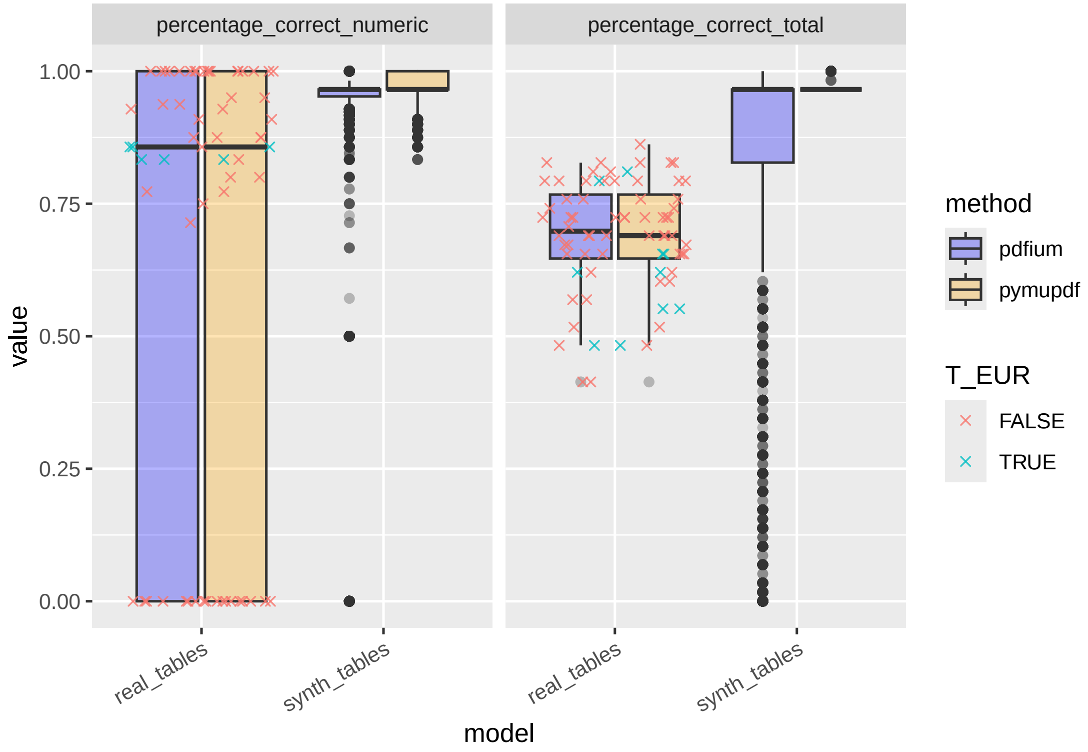
```
df_table_extraction_regex <- read_csv("data_storage/table_extraction_regex.rds")
```

```
## Rows: 24648 Columns: 24
## -- Column specification --------------------------------------------
## Delimiter: ","
## chr  (3): filepath, model, method
## dbl (17): NA_true_positive, NA_false_positive, NA_false_negative, NA_true_ne...
## lgl  (4): json_error, grammar_error, predictions, T_EUR
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

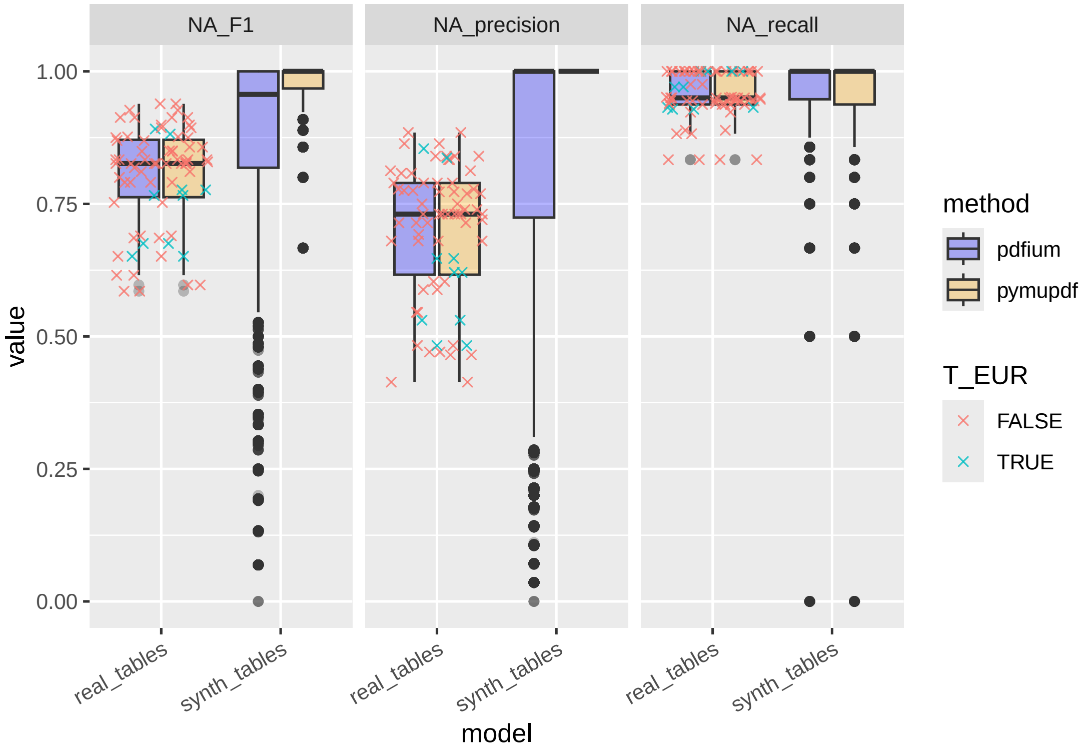
Regex approach not even perfect with synthetic tables. Mistakes in the text parsed with pdfium.

```
## Warning: Removed 327 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

```
## Warning: Removed 10 rows containing missing values or values outside the
## scale range (‘geom_point()‘).
```



```
## Warning: Removed 36914 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



5.2.2 Extraction with LLMs

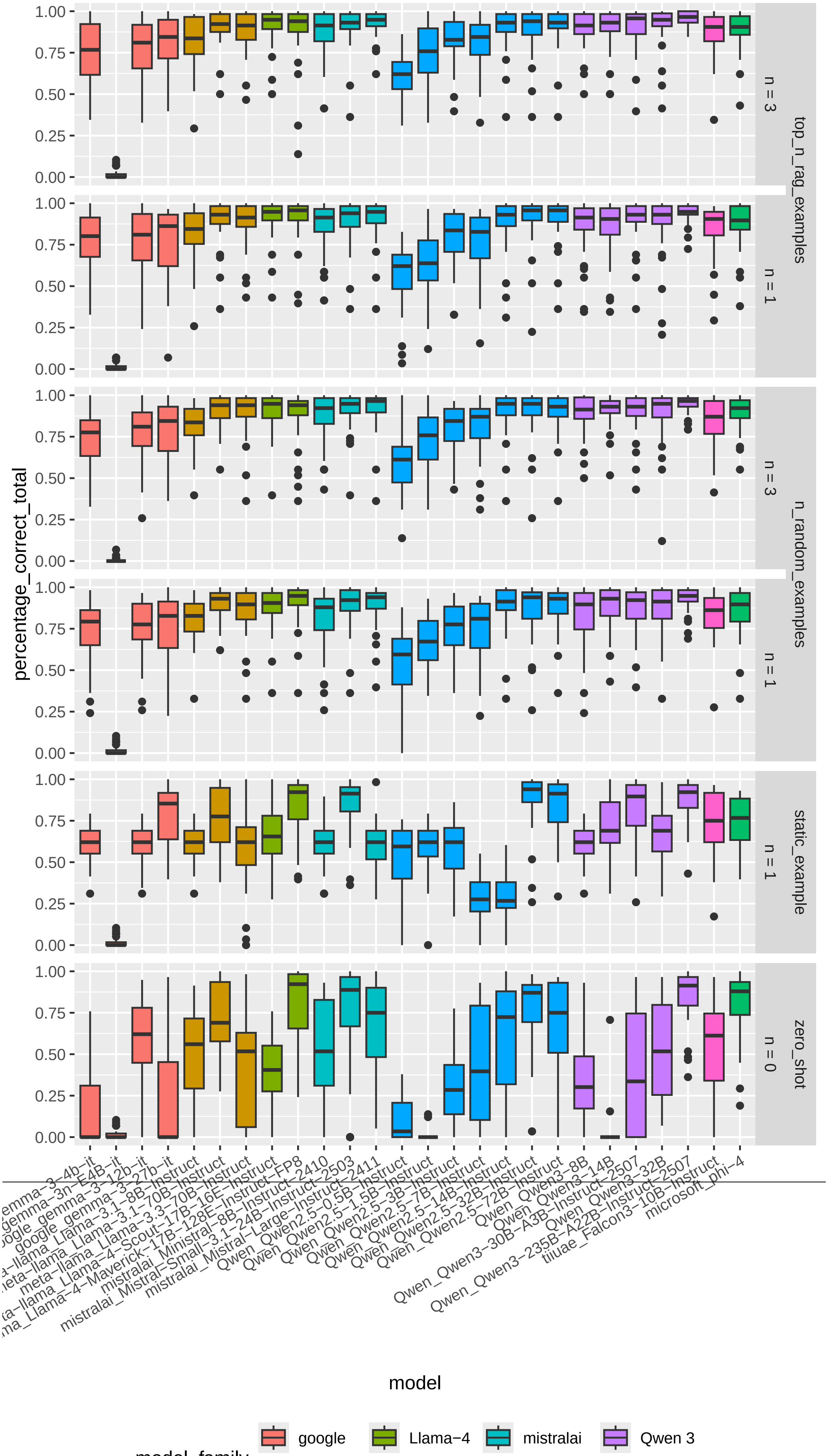
```
## Warning: One or more parsing issues, call ‘problems()‘ on your data frame for
## details, e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 40 Columns: 16
## -- Column specification --------------------------------------------
## Delimiter: ","
## chr  (3): company, filename, spacer
## dbl (13): T_in_year, T_in_previous_year, sum_same_line, n_columns, vorjahr, ...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## Joining with ‘by = join_by(filepath)‘
```

- confidence usable to head for user checks?
- not handled new entries
- five examples bring not much more
- xgboost

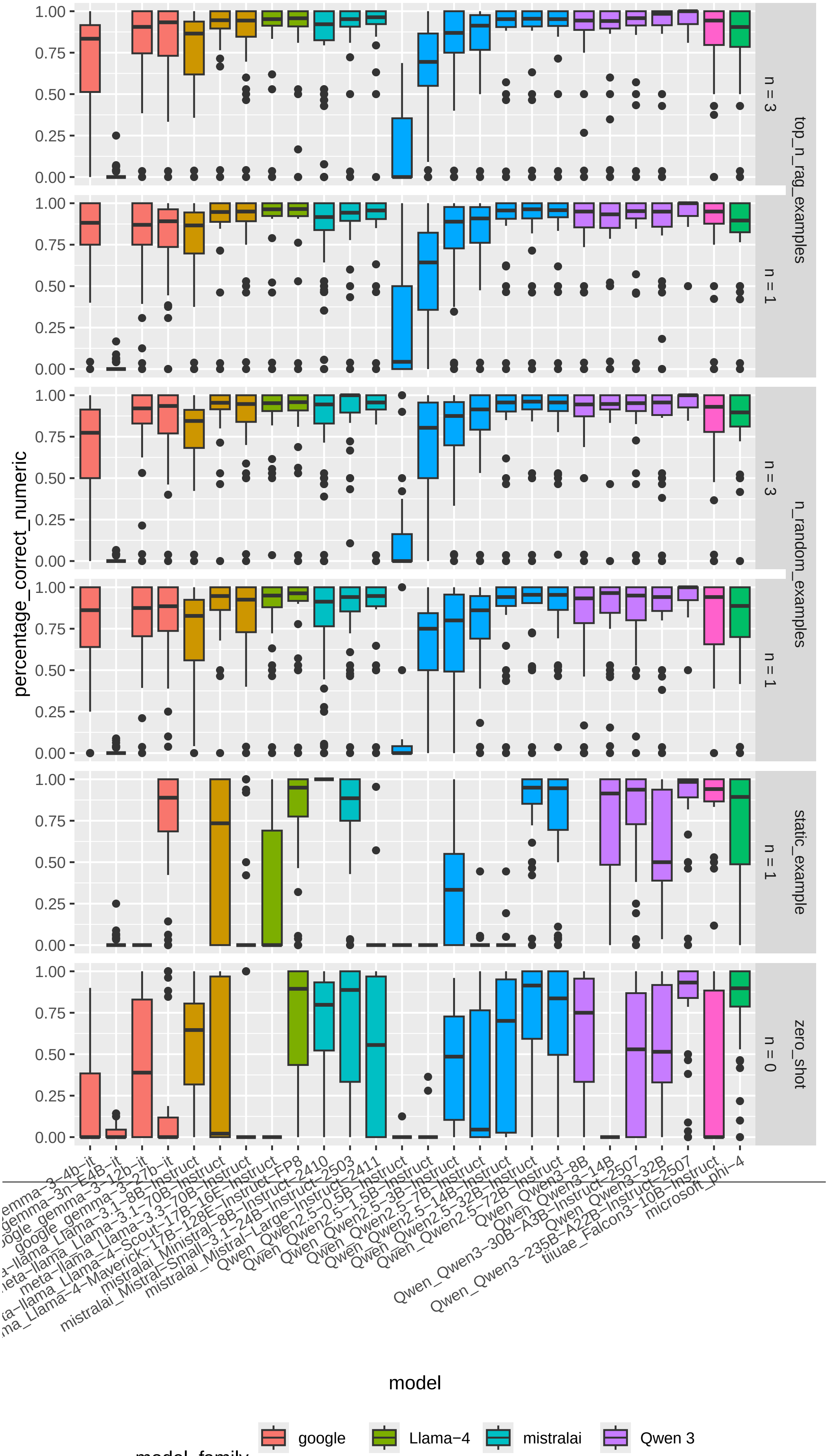
5.2.2.1 Real tables only

```r
df_overview %>%
  ggplot() +
  geom_boxplot(aes(x = model, y = percentage_correct_total, fill = model_family)) +
  scale_x_discrete(guide = guide_axis(angle = 30)) +
  facet_nested(method_family + n_examples ~ .) +
  theme(
    legend.position = "bottom"
  )
```


percentage_correct_total
1.00
0.75
0.50
0.25
0.00
top_n_rag_examples
n = 3
n = 1
n_random_examples
n = 3
n = 1
static_example
n = 1
zero_shot
n = 0
emma−3−4b−it
gemma−3n−E4B−it
ogle_gemma−3−12b−it
google_gemma−3−27b−it
a−llama_Llama−3.1−8B−Instruct
eta−llama_Llama−3.1−70B−Instruct
meta−llama_Llama−3.3−70B−Instruct
ta−llama_Llama−4−Scout−17B−16E−Instruct
ma_Llama−4−Maverick−17B−128E−Instruct−FP8
mistralai_Ministral−8B−Instruct−2410
mistralai_Mistral−Small−3.1−24B−Instruct−2503
mistralai_Mistral−Large−Instruct−2411
Qwen_Qwen2.5−0.5B−Instruct
Qwen_Qwen2.5−1.5B−Instruct
Qwen_Qwen2.5−3B−Instruct
Qwen_Qwen2.5−7B−Instruct
Qwen_Qwen2.5−14B−Instruct
Qwen_Qwen2.5−32B−Instruct
Qwen_Qwen2.5−72B−Instruct
Qwen_Qwen3−8B
Qwen_Qwen3−14B
Qwen_Qwen3−30B−A3B−Instruct−2507
Qwen_Qwen3−32B
Qwen_Qwen3−235B−A22B−Instruct−2507
tiiuae_Falcon3−10B−Instruct
microsoft_phi−4
model
model_family
google
Llama−4
mistralai
Qwen 3

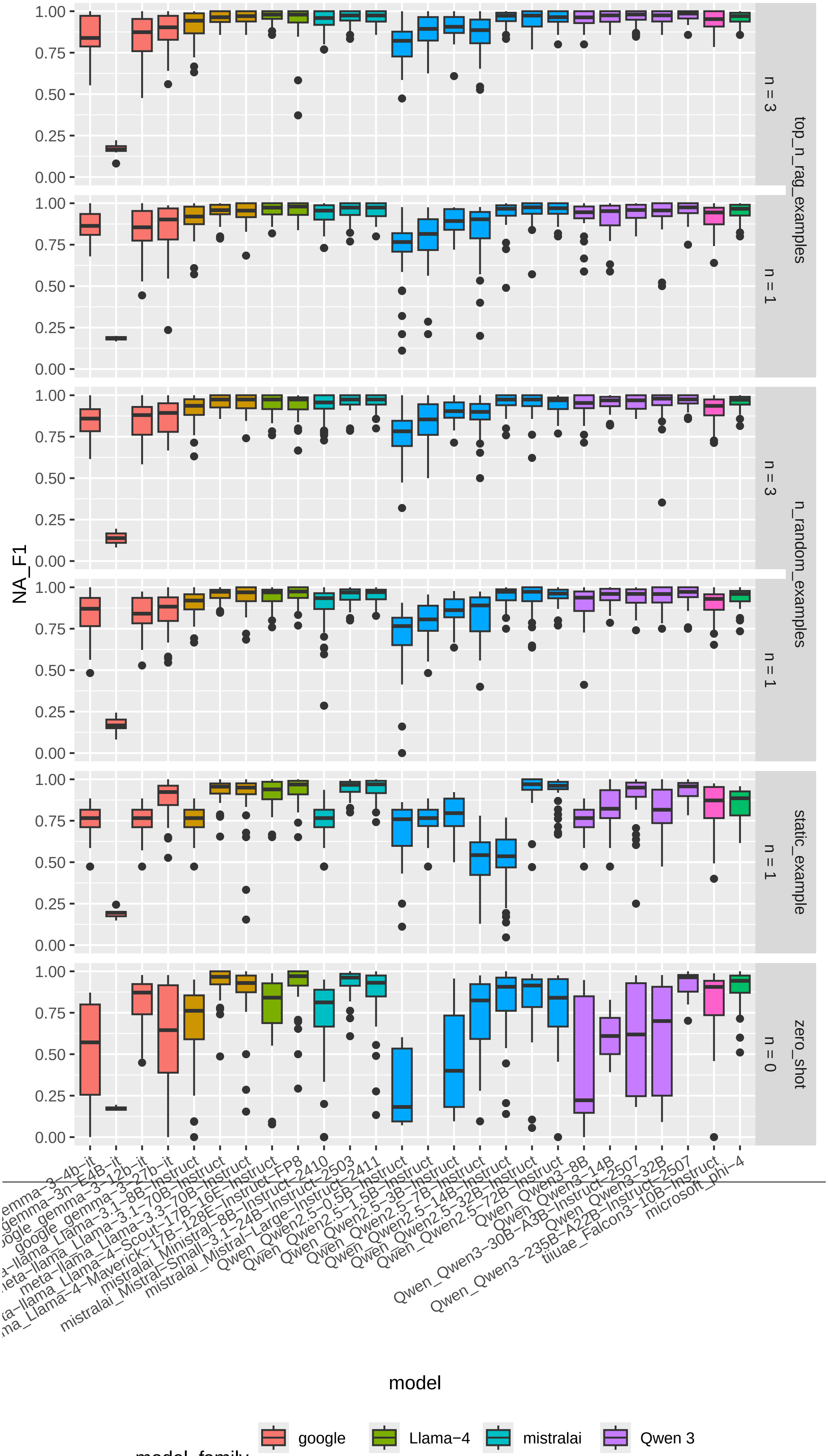
```r
df_overview %>%
  ggplot() +
  geom_boxplot(aes(x = model, y = percentage_correct_numeric, fill = model_family)) +
  scale_x_discrete(guide = guide_axis(angle = 30)) +
  facet_nested(method_family + n_examples ~ .) +
  theme(
    legend.position = "bottom"
  )
```

```
## Warning: Removed 444 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```


percentage_correct_numeric
1.00
0.75
0.50
0.25
0.00
top_n_rag_examples
n = 3
n = 1
n_random_examples
n = 3
n = 1
static_example
n = 1
zero_shot
n = 0
model
emma−3−4b−it
gemma−3n−E4B−it
ogle_gemma−3−12b−it
google_gemma−3−27b−it
a−llama_Llama−3.1−8B−Instruct
eta−llama_Llama−3.1−70B−Instruct
meta−llama_Llama−3.3−70B−Instruct
a−llama_Llama−4−Scout−17B−16E−Instruct
ma_Llama−4−Maverick−17B−128E−Instruct−FP8
mistralai_Ministral−8B−Instruct−2410
mistralai_Mistral−Small−3.1−24B−Instruct−2503
mistralai_Mistral−Large−Instruct−2411
Qwen_Qwen2.5−0.5B−Instruct
Qwen_Qwen2.5−1.5B−Instruct
Qwen_Qwen2.5−3B−Instruct
Qwen_Qwen2.5−7B−Instruct
Qwen_Qwen2.5−14B−Instruct
Qwen_Qwen2.5−32B−Instruct
Qwen_Qwen2.5−72B−Instruct
Qwen_Qwen3−8B
Qwen_Qwen3−14B
Qwen_Qwen3−30B−A3B−Instruct−2507
Qwen_Qwen3−32B
Qwen_Qwen3−235B−A22B−Instruct−2507
tiiuae_Falcon3−10B−Instruct
microsoft_phi−4
model_family
google
Llama−4
mistralai
Qwen 3

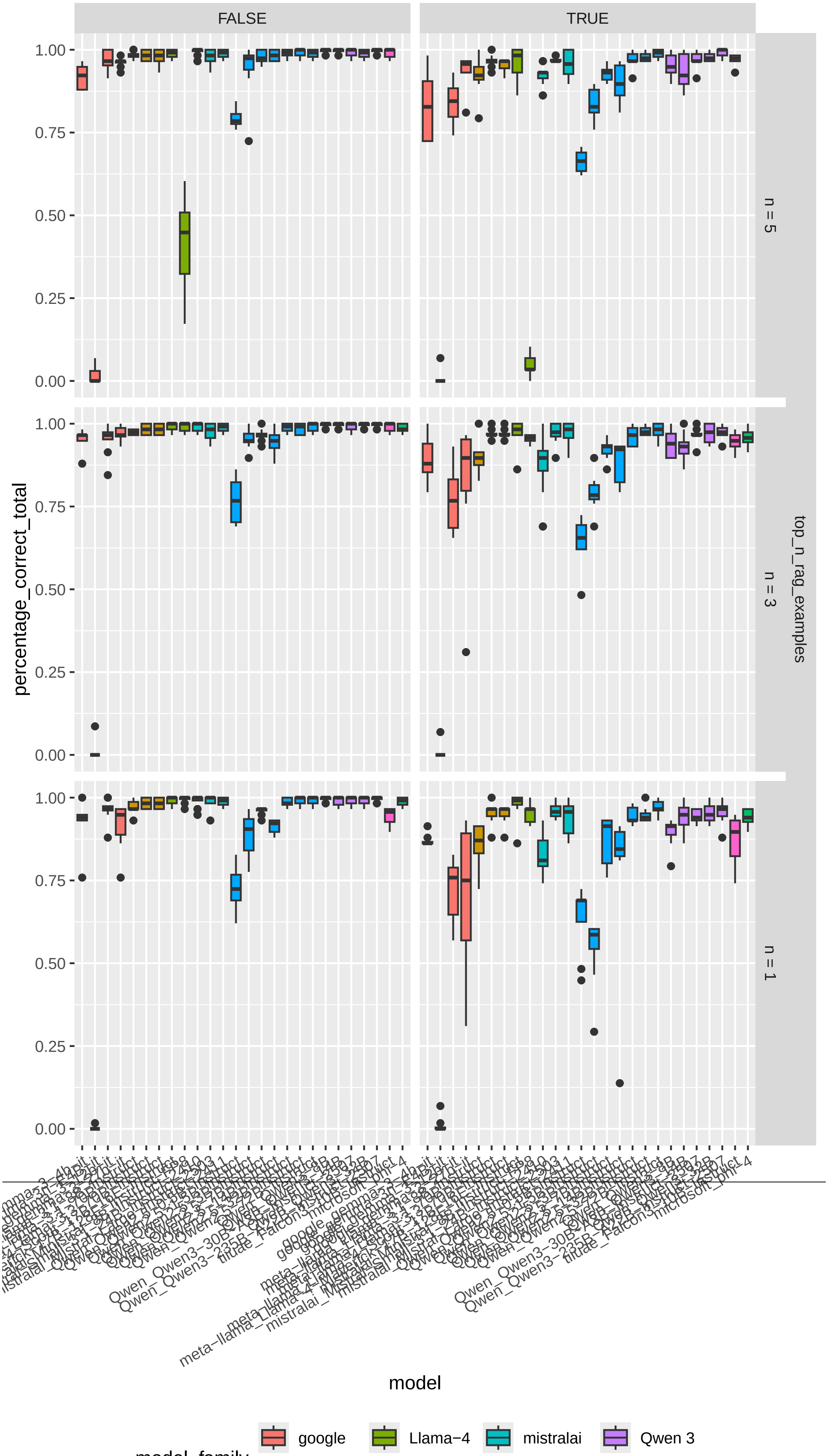
```
df_overview %>%
  ggplot() +
  geom_boxplot(aes(x = model, y = NA_F1, fill = model_family)) +
  scale_x_discrete(guide = guide_axis(angle = 30)) +
  facet_nested(method_family + n_examples ~ .) +
  theme(
    legend.position = "bottom"
  )
```

```
## Warning: Removed 435 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

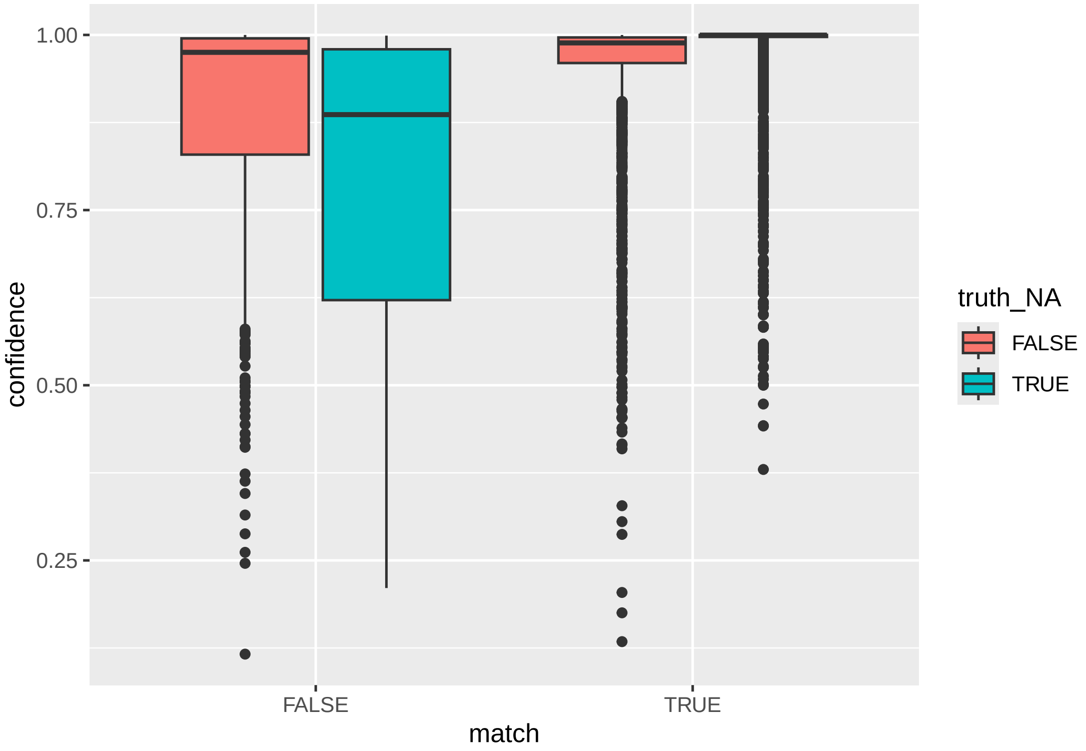

NA_F1
1.00
0.75
0.50
0.25
0.00
top_n_rag_examples
n = 3
n = 1
n_random_examples
n = 3
n = 1
static_example
n = 1
zero_shot
n = 0
emma−3−4b−it
gemma−3n−E4B−it
ogle_gemma−3−12b−it
google_gemma−3−27b−it
a−llama_Llama−3.1−8B−Instruct
eta−llama_Llama−3.1−70B−Instruct
meta−llama_Llama−3.3−70B−Instruct
ta−llama_Llama−4−Scout−17B−16E−Instruct
ma_Llama−4−Maverick−17B−128E−Instruct−FP8
mistralai_Ministral−8B−Instruct−2410
mistralai_Mistral−Small−3.1−24B−Instruct−2503
mistralai_Mistral−Large−Instruct−2411
Qwen_Qwen2.5−0.5B−Instruct
Qwen_Qwen2.5−1.5B−Instruct
Qwen_Qwen2.5−3B−Instruct
Qwen_Qwen2.5−7B−Instruct
Qwen_Qwen2.5−14B−Instruct
Qwen_Qwen2.5−32B−Instruct
Qwen_Qwen2.5−72B−Instruct
Qwen_Qwen3−8B
Qwen_Qwen3−14B
Qwen_Qwen3−30B−A3B−Instruct−2507
Qwen_Qwen3−32B
Qwen_Qwen3−235B−A22B−Instruct−2507
tiiuae_Falcon3−10B−Instruct
microsoft_phi−4
model
model_family
google
Llama−4
mistralai
Qwen 3

Documents from same company help, especially gemma and gpt nano Maverick with 5 examples does not work anymore

```r
bind_rows(df_real_table_extraction, df_real_table_extraction_azure) %>%
  filter(str_detect(filepath, "Statistik"), method_family == "top_n_rag_examples") %>%
  filter(model %in% model_by_size) %>%
  mutate(
    model = factor(model, levels = model_by_size),
    method_family = factor(method_family, levels = method_order),
    n_examples = fct_rev(ordered(paste("n =", n_examples)))
  ) %>%
  # pivot_longer(cols = -c(model, method, model_family)) %>%
  ggplot() +
  geom_boxplot(aes(x = model, y = percentage_correct_total, fill = model_family)) +
  # facet_wrap(~name, ncol = 1) +
  scale_x_discrete(guide = guide_axis(angle = 30)) +
  facet_nested(method_family + n_examples ~ out_of_company) +
  theme(
    legend.position = "bottom"
  )
```

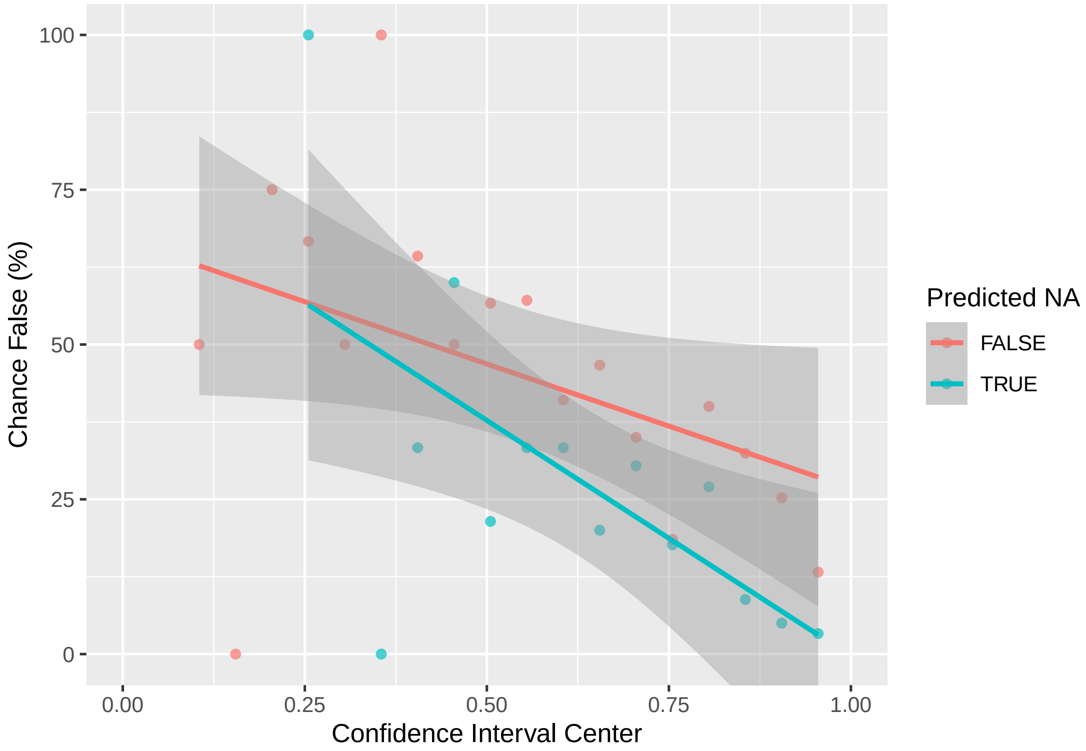

FALSE
TRUE
n = 5
n = 3
n = 1
top_n_rag_examples
percentage_correct_total
1.00
0.75
0.50
0.25
0.00
microsoft_phi-4
tiiuae_Falcon3
Qwen_Qwen3-30B
Qwen_Qwen3-235B
meta-llama_Llama-4
mistralai_Mistral
google_gemma
model
google
Llama−4
mistralai
Qwen 3
model_family

```r
confidence_vs_truth <- df_real_table_extraction %>%
  # filter(model == "Qwen_Qwen3-8B") %>%
  filter(model == "mistralai_Ministral-8B-Instruct-2410") %>%
  group_by(method, model) %>% mutate(
    mean_percentage_correct_total = mean(percentage_correct_total, na.rm=TRUE), .before = 1
    ) %>% ungroup() %>%
  arrange(desc(mean_percentage_correct_total)) %>%
  slice_max(mean_percentage_correct_total, n = 1, with_ties = TRUE) %>%
  mutate(predictions_processed = map(predictions, ~{
    .x %>%
      select(-"_merge") %>%
      mutate(
        match = (year_truth == year_result) | (is.na(year_truth) & is.na(year_result)),
        confidence = confidence_this_year,
        truth_NA = is.na(year_truth),
        predicted_NA = is.na(year_result),
        .before = 4
      ) %>% nest(
        tuple_year = c(match, confidence, truth_NA, predicted_NA)
      ) %>%
      mutate(
        confidence = confidence_previous_year,
        match = (previous_year_truth == previous_year_result) | (is.na(previous_year_truth) & is.na(previ
        truth_NA = is.na(previous_year_truth),
        predicted_NA = is.na(previous_year_result),
        .before = 4
      ) %>% nest(
        tuple_previous_year = c(match, confidence, truth_NA, predicted_NA)
      ) %>% select(
        -c(year_truth, previous_year_truth, year_result, previous_year_result,
           confidence_this_year, confidence_previous_year)
      ) %>%
      pivot_longer(-c("E1", "E2", "E3")) %>%
      unnest(cols = value) %>% mutate(
        match = if_else(is.na(match), FALSE, match)
      )
  })) %>%
  unnest(predictions_processed) %>% mutate(
    match = factor(match, levels = c(F, T)),
    truth_NA = factor(truth_NA, levels = c(F, T))
  )

confidence_vs_truth %>% ggplot() +
  geom_boxplot(
    aes(x = match, y = confidence, fill = truth_NA),
    position = position_dodge2(preserve = "single")) +
  scale_fill_discrete(drop = FALSE) +
  scale_x_discrete(drop = FALSE)
```



```
confidence_vs_truth %>%
  mutate(
    conf_interval = cut(confidence, breaks = seq(0, 1, by = 0.05), include.lowest = TRUE),
    conf_center = as.numeric(sub("\\((.+),(.+)\\]", "\\1", levels(conf_interval))[conf_interval])
  ) %>%
  group_by(conf_center, predicted_NA) %>%
  summarize(
    n_true = sum(match == TRUE, na.rm = TRUE),
    n_false = sum(match == FALSE, na.rm = TRUE),
    total = n_true + n_false,
    chance_false = if_else(total > 0, n_false / total * 100, NA_real_),
    .groups = "drop"
  ) %>%
  ggplot(aes(x = conf_center, y = chance_false, color = predicted_NA)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", se = TRUE) +
  labs(x = "Confidence Interval Center", y = "Chance False (%)", color = "Predicted NA") +
  coord_cartesian(ylim = c(0, 100), xlim = c(0,1))
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

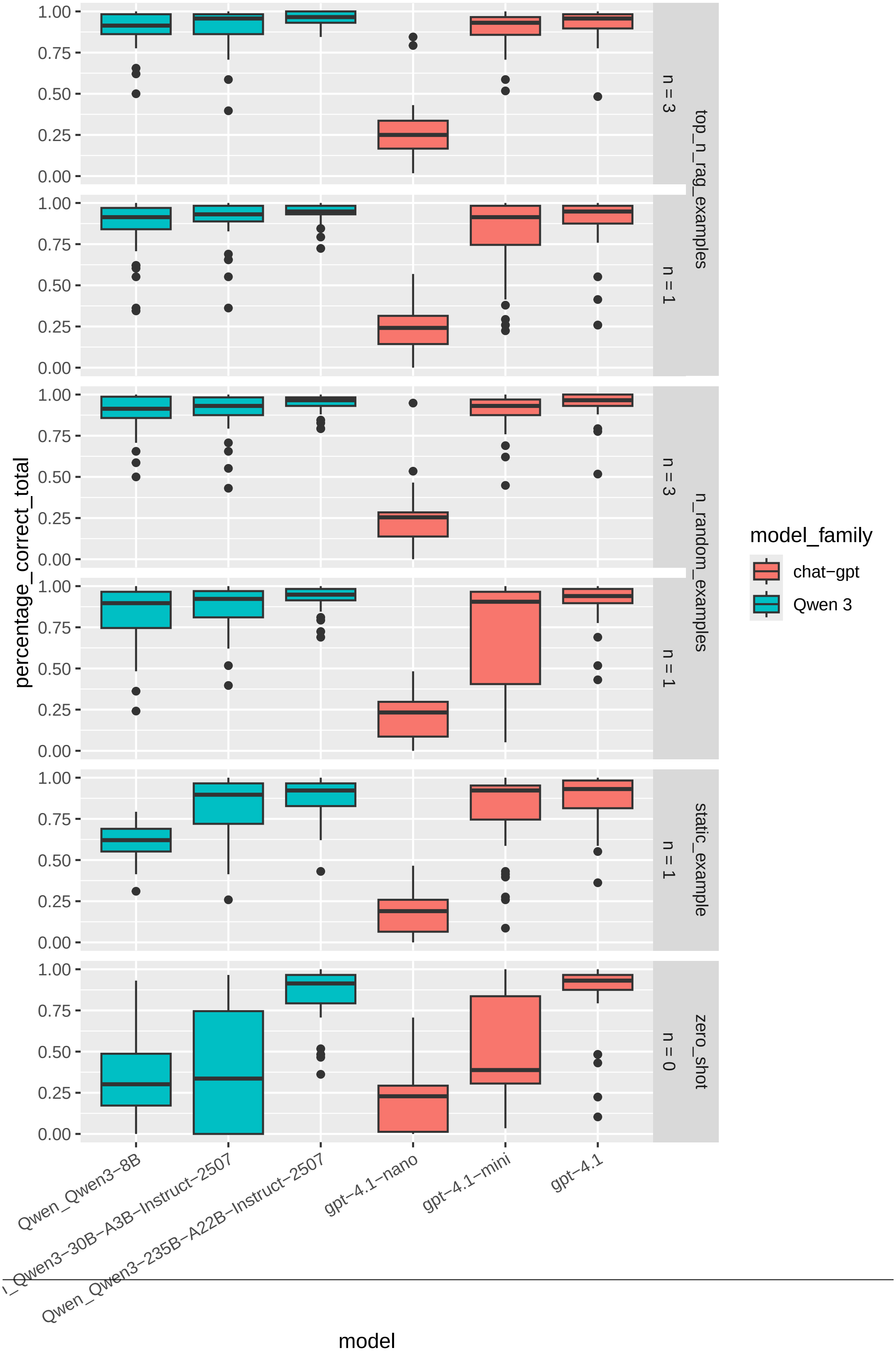


5.2.2.1.1 GPT GPT 4.1 nano not sufficiently powerful, GPT 4.1 mini good to moderate in most cases, GPT 4.1 performs very good.

GPT 4.1 costs five times of mini and 20 times of nano. But nano is useless for the task

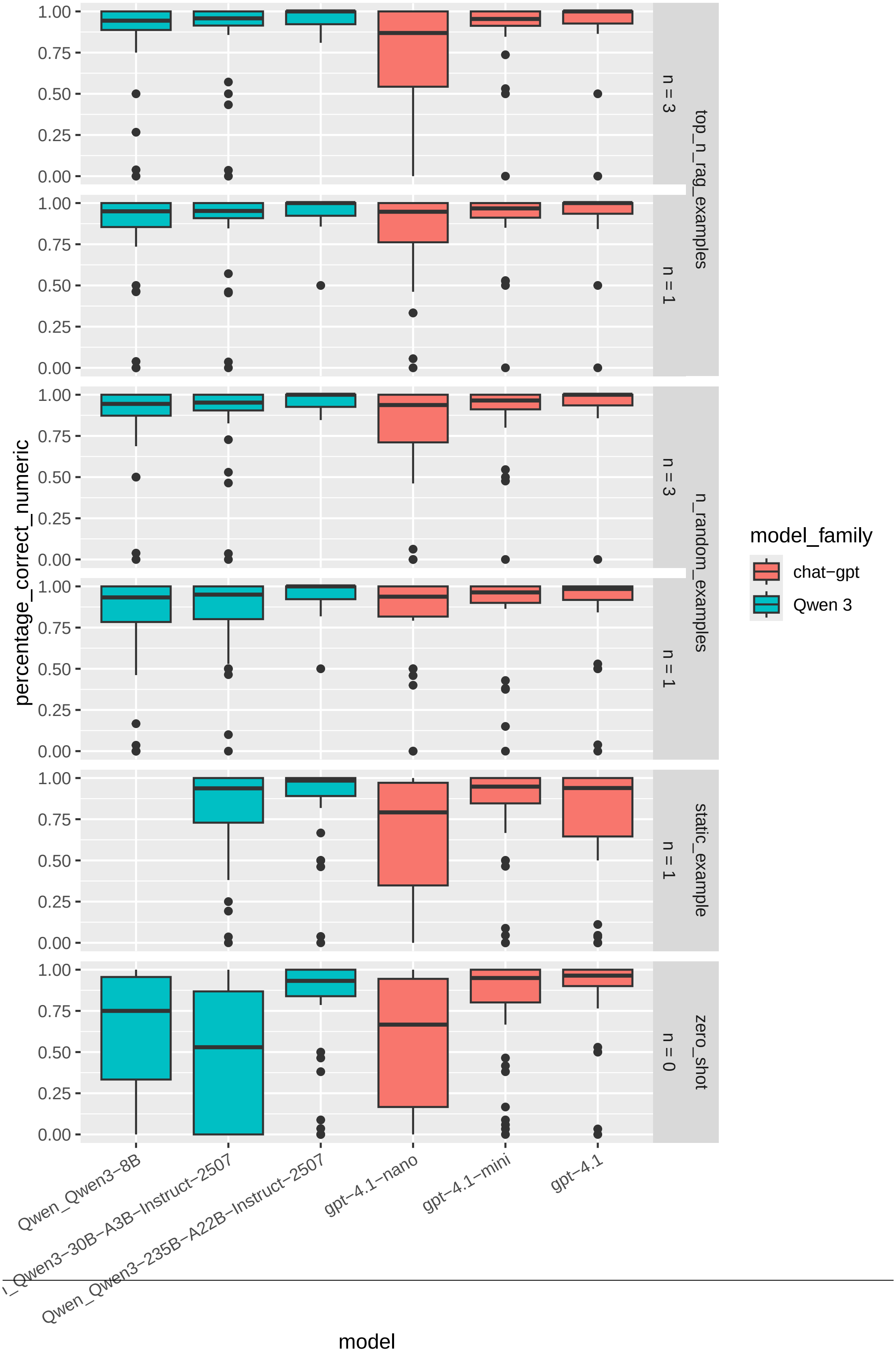
```
model_by_size_gpt <- c(
  "Qwen_Qwen3-8B", "Qwen_Qwen3-30B-A3B-Instruct-2507", "Qwen_Qwen3-235B-A22B-Instruct-2507",
 "gpt-4.1-nano", "gpt-4.1-mini", "gpt-4.1"
)

df_overview_gpt <- bind_rows(df_real_table_extraction, df_real_table_extraction_azure) %>%
  filter(out_of_company != TRUE | is.na(out_of_company), n_examples <= 3) %>%
  filter(model %in% model_by_size_gpt) %>%
  mutate(
    model = factor(model, levels = model_by_size_gpt),
    method_family = factor(method_family, levels = method_order),
    n_examples = fct_rev(ordered(paste("n =", n_examples)))
    )
```

```
df_overview_gpt %>%
  ggplot() +
  geom_boxplot(aes(x = model, y = percentage_correct_total, fill = model_family)) +
  scale_x_discrete(guide = guide_axis(angle = 30)) +
  facet_nested(method_family + n_examples ~ .)
```


percentage_correct_total
1.00
0.75
0.50
0.25
0.00
top_n_rag_examples
n = 3
n = 1
n_random_examples
n = 3
n = 1
static_example
n = 1
zero_shot
n = 0
model_family
chat−gpt
Qwen 3
Qwen_Qwen3−8B
_Qwen3−30B−A3B−Instruct−2507
Qwen_Qwen3−235B−A22B−Instruct−2507
gpt−4.1−nano
gpt−4.1−mini
gpt−4.1
model

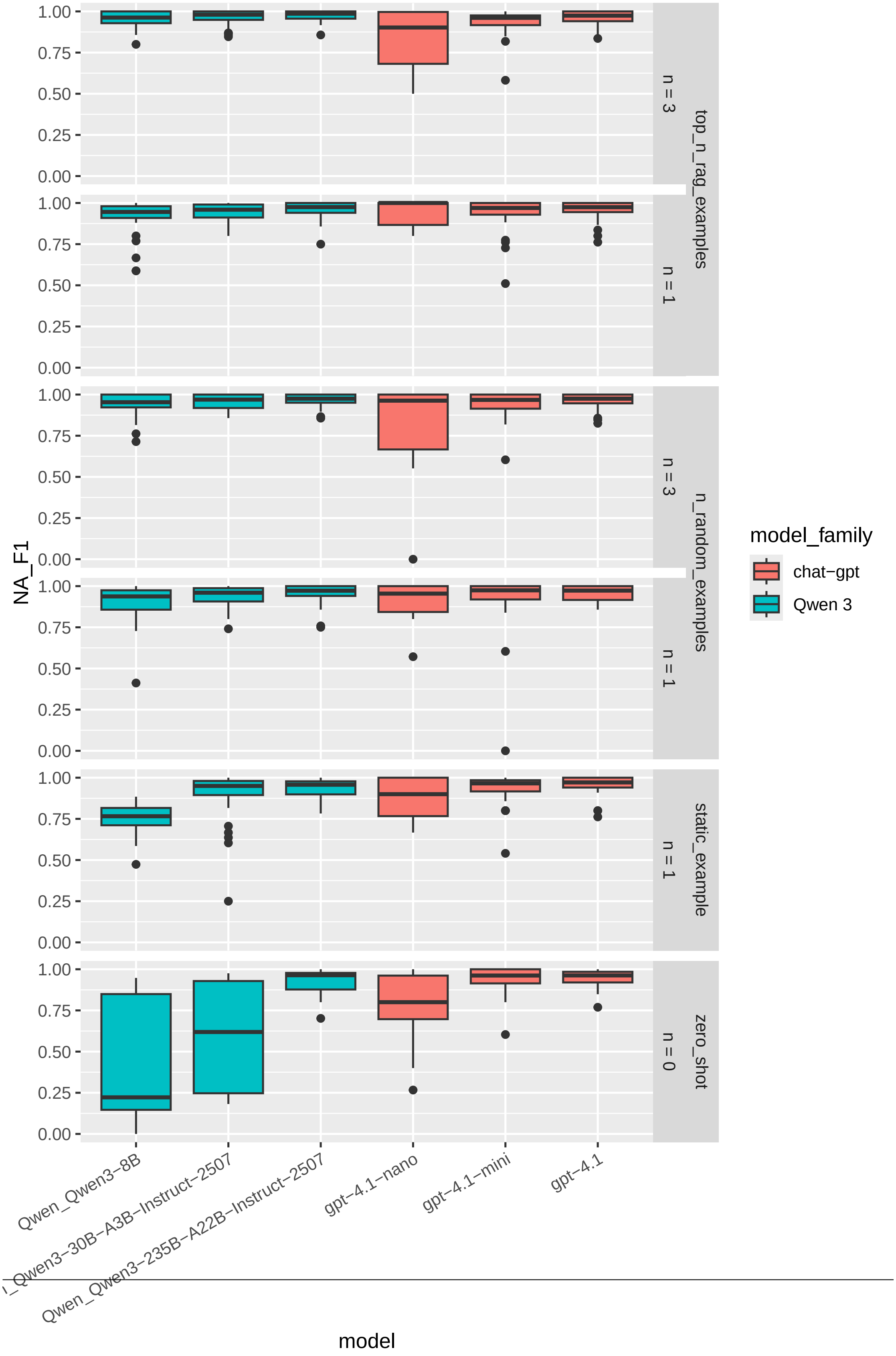
```
df_overview_gpt %>%
  ggplot() +
  geom_boxplot(aes(x = model, y = percentage_correct_numeric, fill = model_family)) +
  scale_x_discrete(guide = guide_axis(angle = 30)) +
  facet_nested(method_family + n_examples ~ .)
```

```
## Warning: Removed 50 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```


percentage_correct_numeric
1.00
0.75
0.50
0.25
0.00
top_n_rag_examples
n = 3
n = 1
n_random_examples
n = 3
n = 1
static_example
n = 1
zero_shot
n = 0
model_family
chat−gpt
Qwen 3
Qwen_Qwen3−8B
n_Qwen3−30B−A3B−Instruct−2507
Qwen_Qwen3−235B−A22B−Instruct−2507
gpt−4.1−nano
gpt−4.1−mini
gpt−4.1
model

```r
df_overview_gpt %>%
  ggplot() +
  geom_boxplot(aes(x = model, y = NA_F1, fill = model_family)) +
  scale_x_discrete(guide = guide_axis(angle = 30)) +
  facet_nested(method_family + n_examples ~ .)
```

```
## Warning: Removed 222 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```


NA_F1
1.00
0.75
0.50
0.25
0.00
n = 3
n = 1
top_n_rag_examples
n_random_examples
static_example
zero_shot
n = 0
model_family
chat−gpt
Qwen 3
Qwen_Qwen3−8B
i_Qwen3−30B−A3B−Instruct−2507
Qwen_Qwen3−235B−A22B−Instruct−2507
gpt−4.1−nano
gpt−4.1−mini
gpt−4.1
model

```
costs_azure <- read_csv("../CostManagement_master-thesis_2025.csv")
```

```
## Rows: 6 Columns: 3
## -- Column specification --------------------------------------------
## Delimiter: ","
## chr (2): Meter, Currency
## dbl (1): Cost
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
token_prop <- df_real_table_extraction %>% group_by(model, method, n_examples) %>% summarize(
    request_tokens_total = sum(request_tokens[[1]])) %>%
    group_by(method, n_examples) %>%
    summarize(mean = mean(request_tokens_total, na.rm = TRUE)) %>% mutate(five_examples = n_examples == 5
```

```
## 'summarise()' has grouped output by 'model', 'method'. You can
## override using the '.groups' argument.
## 'summarise()' has grouped output by 'method'. You can override using
## the '.groups' argument.
```

```
five_ex_tokens <- token_prop %>% filter(five_examples == TRUE) %>% pull(sum)
other_tokens <- token_prop %>% filter(five_examples == FALSE) %>% pull(sum)

costs_azure %>% mutate(
  Cost_all_tasks = Cost,
  Cost_all_tasks = if_else(Meter == "gpt 4.1 Inp glbl Tokens", Cost_all_tasks+Cost_all_tasks*five_ex_toke
  Cost_all_tasks = if_else(Meter == "gpt 4.1 Outp glbl Tokens", Cost_all_tasks+Cost_all_tasks*3/11, Cost_a
  ) %>% kbl()
```

Meter	Cost	Currency	Cost_all_tasks
gpt 4.1 Inp glbl Tokens	3.5314178	EUR	6.9920400
gpt 4.1 Outp glbl Tokens	2.7056303	EUR	3.4435295
gpt 4.1 mini Inp glbl Tokens	1.2276150	EUR	1.2276150
gpt 4.1 mini Outp glbl Tokens	0.7081203	EUR	0.7081203
gpt 4.1 nano Inp glbl Tokens	0.3148390	EUR	0.3148390
gpt 4.1 nano Outp glbl Tokens	0.1466051	EUR	0.1466051

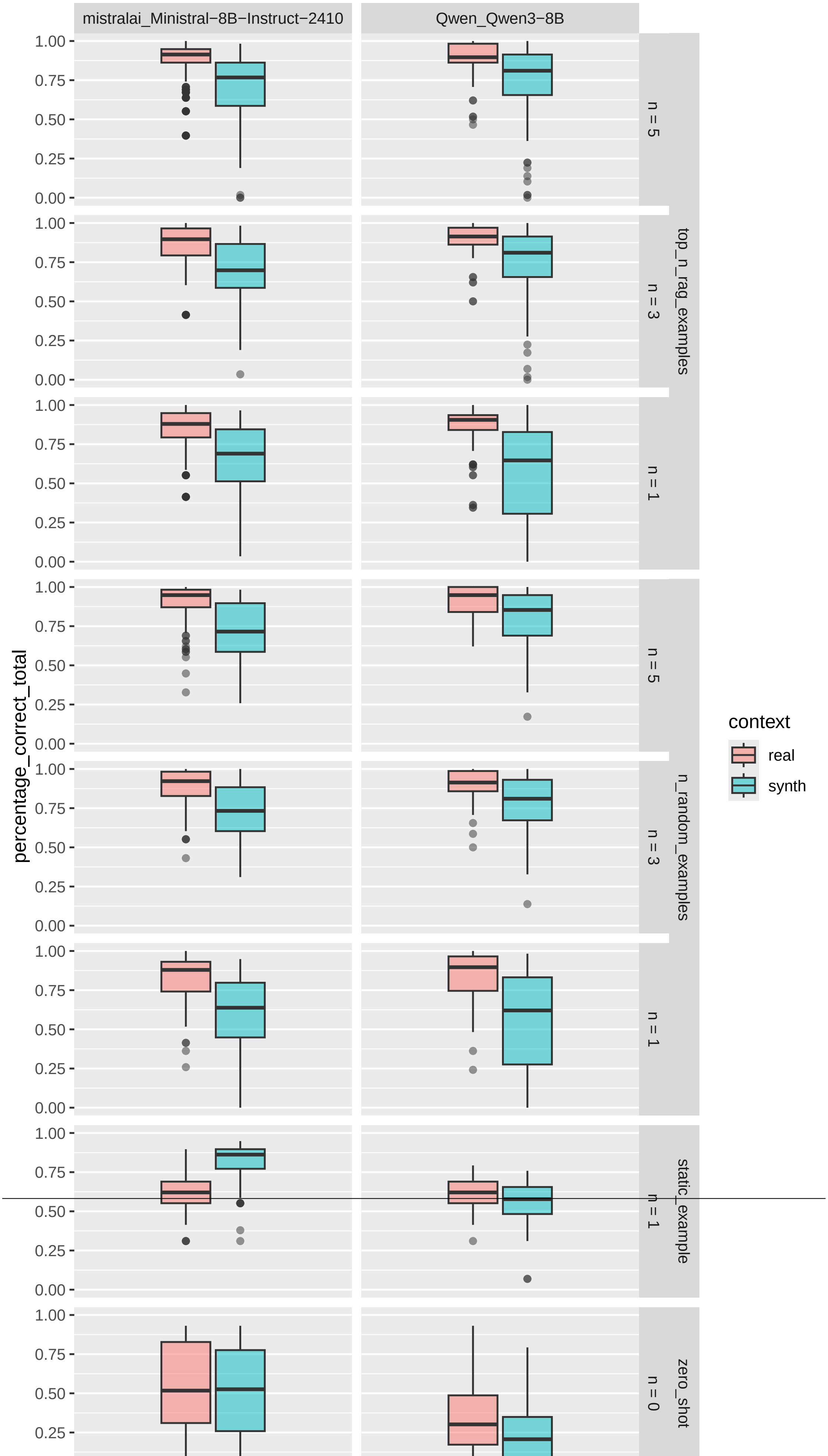
5.2.2.2 Synthetic tables only

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

5.2.2.3 Extract from real tables with synthetic content

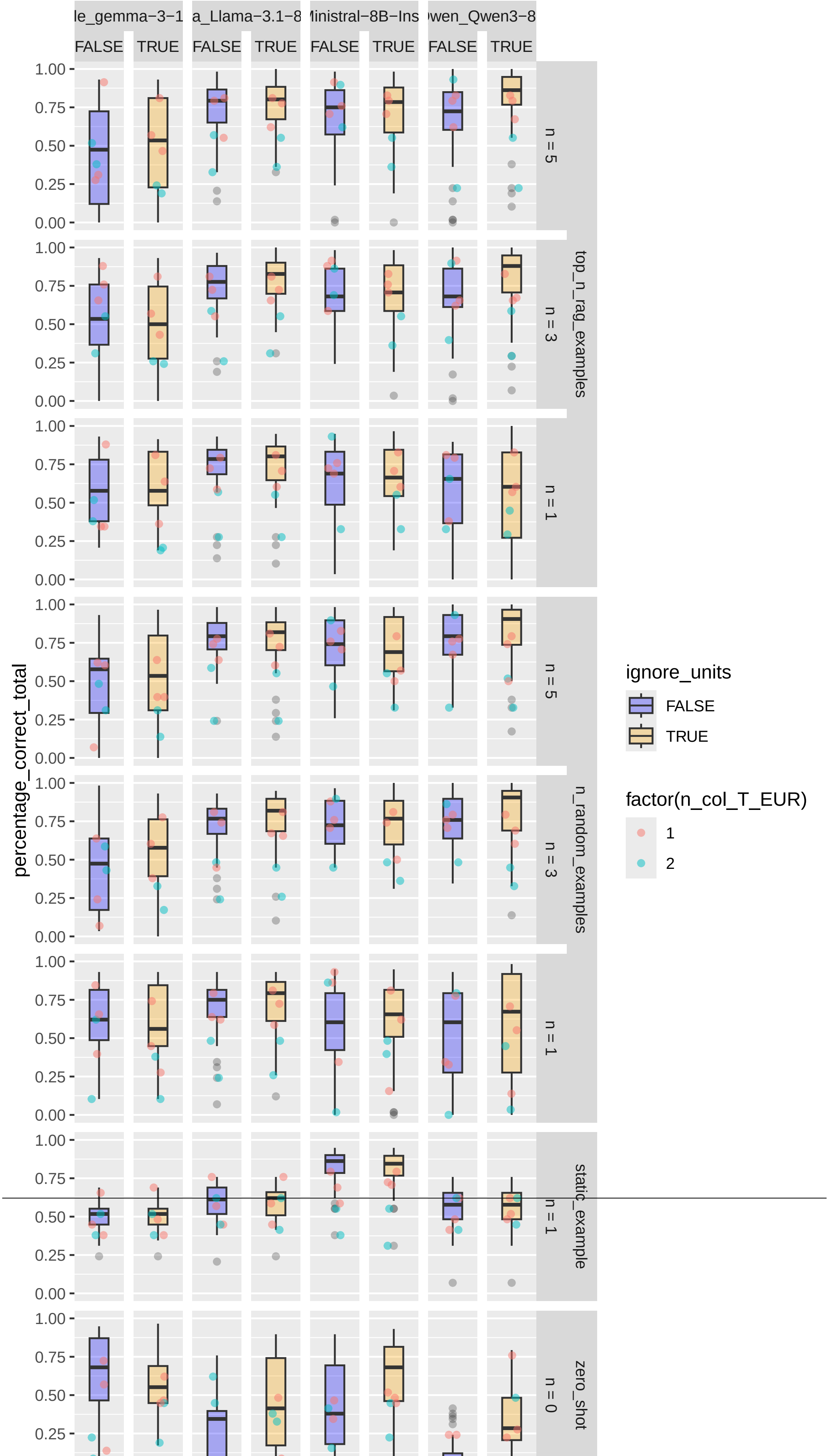
Real context better for real tables. But not useless.

```
bind_rows(
  df_real_table_extraction_synth %>% mutate(context = "synth"),
  df_real_table_extraction %>% mutate(context = "real")
  ) %>%
  filter(model %in% c("Qwen_Qwen3-8B","mistralai_Ministral-8B-Instruct-2410")) %>%
  mutate(
    model = factor(model, levels = model_by_size),
    method_family = factor(method_family, levels = method_order),
    n_examples = fct_rev(ordered(paste("n =", n_examples)))
  ) %>%
  ggplot() +
  geom_boxplot(aes(x = 1,, fill=context, y = percentage_correct_total), alpha = .5) +
  # scale_fill_manual(values = c("blue", "orange")) +
  scale_x_discrete(guide = guide_axis(angle = 30)) +
  facet_nested(method_family+n_examples~model)
```


mistralai_Ministral−8B−Instruct−2410
Qwen_Qwen3−8B
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
context
real
synth

no examples with units only for one column. Can learn from synth context new skills

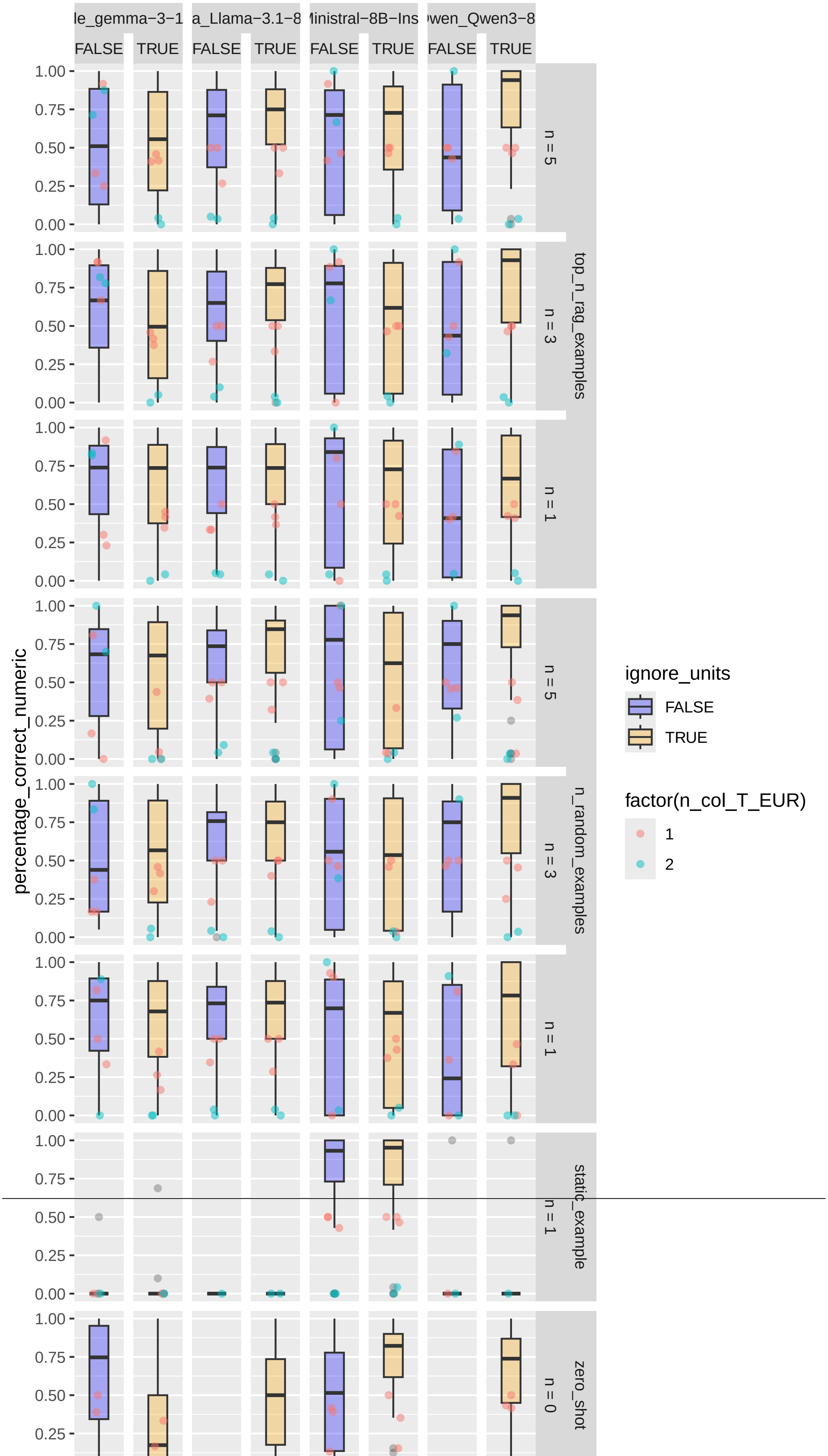
```
df_real_table_extraction_synth %>%
  mutate(n_col_T_EUR = T_EUR_both + T_EUR) %>%
  mutate(
    model = factor(model, levels = model_by_size),
    method_family = factor(method_family, levels = method_order),
    n_examples = fct_rev(ordered(paste("n =", n_examples)))
  ) %>%
  ggplot() +
  geom_boxplot(aes(x = 1, fill=ignore_units, y = percentage_correct_total), alpha = .3) +
  geom_jitter(
    data = . %>% filter(n_col_T_EUR > 0),
    aes(x = 1, group=ignore_units, color = factor(n_col_T_EUR), y = percentage_correct_total),
    height = 0, alpha = .5, width = 0.3
    ) +
  scale_fill_manual(values = c("blue", "orange")) +
  scale_x_discrete(guide = guide_axis(angle = 30)) +
  facet_nested(method_family+n_examples~model+ignore_units)
```


le_gemma−3−1
a_Llama−3.1−8
Ministral−8B−Ins
wen_Qwen3−8
FALSE
TRUE
1.00
0.75
0.50
0.25
0.00
percentage_correct_total
n = 5
n = 3
n = 1
n = 0
top_n_rag_examples
n_random_examples
static_example
zero_shot
ignore_units
FALSE
TRUE
factor(n_col_T_EUR)
1
2

```
df_real_table_extraction_synth %>%
  mutate(n_col_T_EUR = T_EUR_both + T_EUR) %>%
  mutate(
    model = factor(model, levels = model_by_size),
    method_family = factor(method_family, levels = method_order),
    n_examples = fct_rev(ordered(paste("n =", n_examples)))
  ) %>%
  ggplot() +
  geom_boxplot(aes(x = 1, fill=ignore_units, y = percentage_correct_numeric), alpha = .3) +
  geom_jitter(
    data = . %>% filter(n_col_T_EUR > 0),
    aes(x = 1, group=ignore_units, color = factor(n_col_T_EUR), y = percentage_correct_numeric),
    height = 0, alpha = .5, width = 0.3
    ) +
  scale_fill_manual(values = c("blue", "orange")) +
  scale_x_discrete(guide = guide_axis(angle = 30)) +
  facet_nested(method_family+n_examples~model+ignore_units)
```

```
## Warning: Removed 106 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

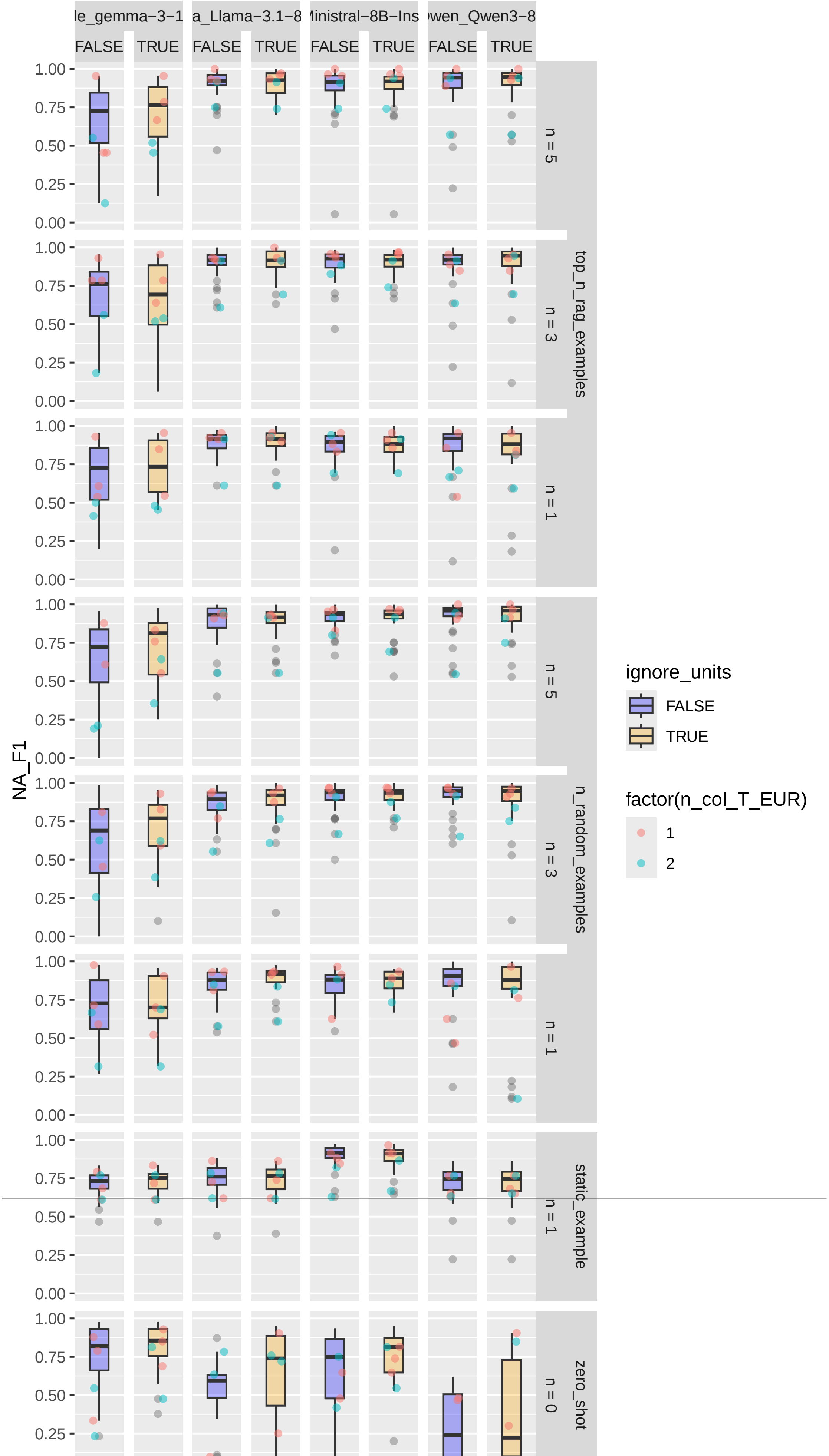
```
## Warning: Removed 16 rows containing missing values or values outside the
## scale range (‘geom_point()‘).
```


le_gemma−3−1
a_Llama−3.1−8
Ministral−8B−Ins
wen_Qwen3−8
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
ignore_units
FALSE
TRUE
factor(n_col_T_EUR)
1
2

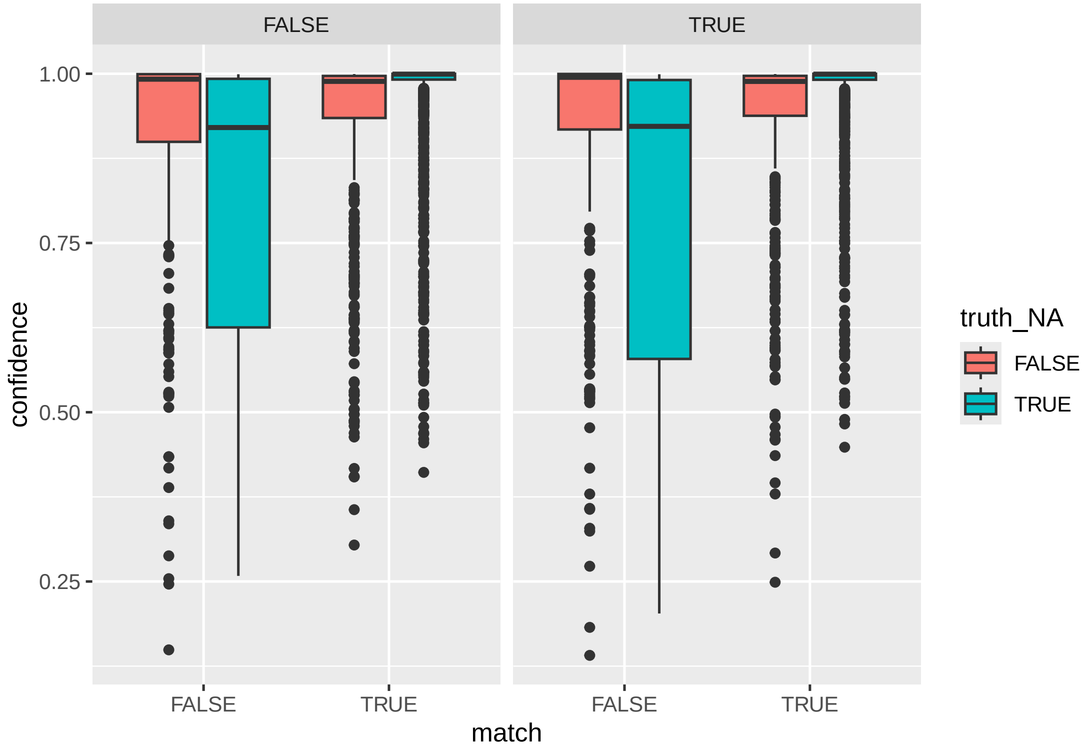
```
df_real_table_extraction_synth %>%
  mutate(n_col_T_EUR = T_EUR_both + T_EUR) %>%
  mutate(
    model = factor(model, levels = model_by_size),
    method_family = factor(method_family, levels = method_order),
    n_examples = fct_rev(ordered(paste("n =", n_examples)))
  ) %>%
  ggplot() +
  geom_boxplot(aes(x = 1, fill=ignore_units, y = NA_F1), alpha = .3) +
  geom_jitter(
    data = . %>% filter(n_col_T_EUR > 0),
    aes(x = 1, group=ignore_units, color = factor(n_col_T_EUR), y = NA_F1),
    height = 0, alpha = .5, width = 0.3
    ) +
  scale_fill_manual(values = c("blue", "orange")) +
  scale_x_discrete(guide = guide_axis(angle = 30)) +
  facet_nested(method_family+n_examples~model+ignore_units)
```

```
## Warning: Removed 232 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

```
## Warning: Removed 14 rows containing missing values or values outside the
## scale range (‘geom_point()‘).
```

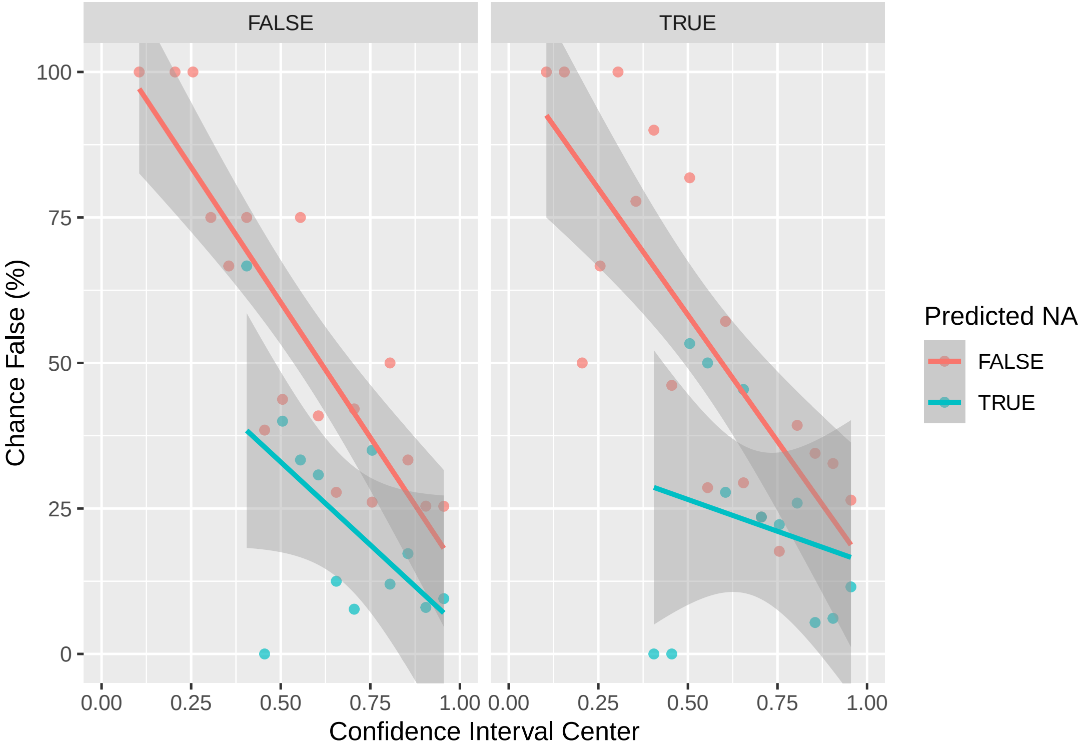

le_gemma−3−1
a_Llama−3.1−8
Ministral−8B−Ins
wen_Qwen3−8
FALSE
TRUE
1.00
0.75
0.50
0.25
0.00
NA_F1
n = 5
n = 3
n = 1
n = 0
top_n_rag_examples
n_random_examples
static_example
zero_shot
ignore_units
FALSE
TRUE
factor(n_col_T_EUR)
1
2

```r
confidence_vs_truth <- df_real_table_extraction_synth %>%
  # filter(model == "Qwen_Qwen3-8B") %>%
  filter(model == "mistralai_Ministral-8B-Instruct-2410") %>%
  group_by(method, model) %>% mutate(
    mean_percentage_correct_total = mean(percentage_correct_total, na.rm=TRUE), .before = 1
  ) %>% group_by(ignore_units) %>%
  arrange(desc(mean_percentage_correct_total)) %>%
  slice_max(mean_percentage_correct_total, n = 1, with_ties = TRUE) %>%
  mutate(predictions_processed = map(predictions, ~{
    .x %>%
      select(-"_merge") %>%
      mutate(
        match = (year_truth == year_result) | (is.na(year_truth) & is.na(year_result)),
        confidence = confidence_this_year,
        truth_NA = is.na(year_truth),
        predicted_NA = is.na(year_result),
        .before = 4
      ) %>% nest(
        tuple_year = c(match, confidence, truth_NA, predicted_NA)
      ) %>%
      mutate(
        confidence = confidence_previous_year,
        match = (previous_year_truth == previous_year_result) | (is.na(previous_year_truth) & is.n
        truth_NA = is.na(previous_year_truth),
        predicted_NA = is.na(previous_year_result),
        .before = 4
      ) %>% nest(
        tuple_previous_year = c(match, confidence, truth_NA, predicted_NA)
      ) %>% select(
        -c(year_truth, previous_year_truth, year_result, previous_year_result,
           confidence_this_year, confidence_previous_year)
      ) %>%
      pivot_longer(-c("E1", "E2", "E3")) %>%
      unnest(cols = value) %>% mutate(
        match = if_else(is.na(match), FALSE, match)
      )
  })) %>%
  unnest(predictions_processed) %>% mutate(
    match = factor(match, levels = c(F, T)),
    truth_NA = factor(truth_NA, levels = c(F, T))
  )

confidence_vs_truth %>% ggplot() +
  geom_boxplot(
    aes(x = match, y = confidence, fill = truth_NA),
    position = position_dodge2(preserve = "single")) +
  scale_fill_discrete(drop = FALSE) +
  scale_x_discrete(drop = FALSE) +
  facet_wrap(~ignore_units)
```



```
confidence_vs_truth %>%
  mutate(
    conf_interval = cut(confidence, breaks = seq(0, 1, by = 0.05), include.lowest = TRUE),
    conf_center = as.numeric(sub("\\((.+),(.+)\\]", "\\1", levels(conf_interval))[conf_interval]) + 0.005
  ) %>%
  group_by(conf_center, predicted_NA, ignore_units) %>%
  summarize(
    n_true = sum(match == TRUE, na.rm = TRUE),
    n_false = sum(match == FALSE, na.rm = TRUE),
    total = n_true + n_false,
    chance_false = if_else(total > 0, n_false / total * 100, NA_real_),
    .groups = "drop"
  ) %>%
  ggplot(aes(x = conf_center, y = chance_false, color = predicted_NA)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", se = TRUE) +
  labs(x = "Confidence Interval Center", y = "Chance False (%)", color = "Predicted NA") +
  coord_cartesian(ylim = c(0, 100), xlim = c(0,1)) +
  facet_wrap(~ignore_units)
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```


FALSE
TRUE
Chance False (%)
100
75
50
25
0
0.00
0.25
0.50
0.75
1.00
Confidence Interval Center
Predicted NA
FALSE
TRUE

Chapter 6

Discussion

- ensemble prediction
- check for halluzination vs wrong placed / repeated numbers

6.1 Not covered

- OCR

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

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

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

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

- **Windowing System:** Wayland
- **Kernel Version:** Linux 6.11.0-29-generic

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.

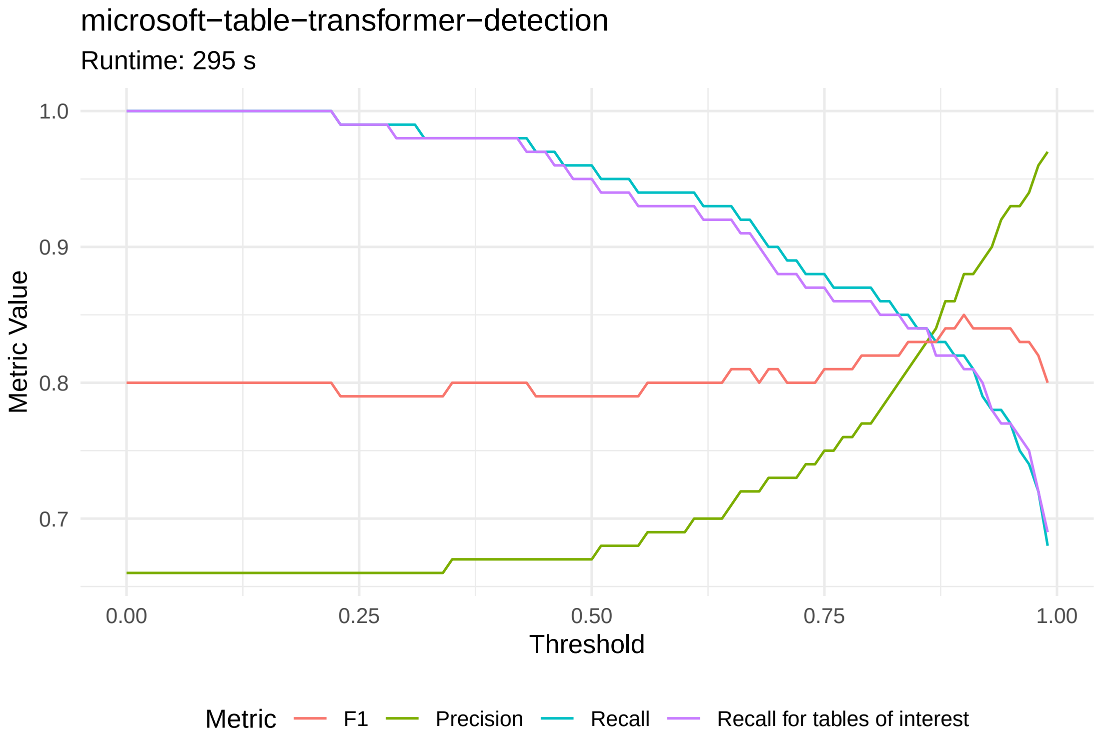
The result of docling-parse is not formated as markdown yet but also just plain text.

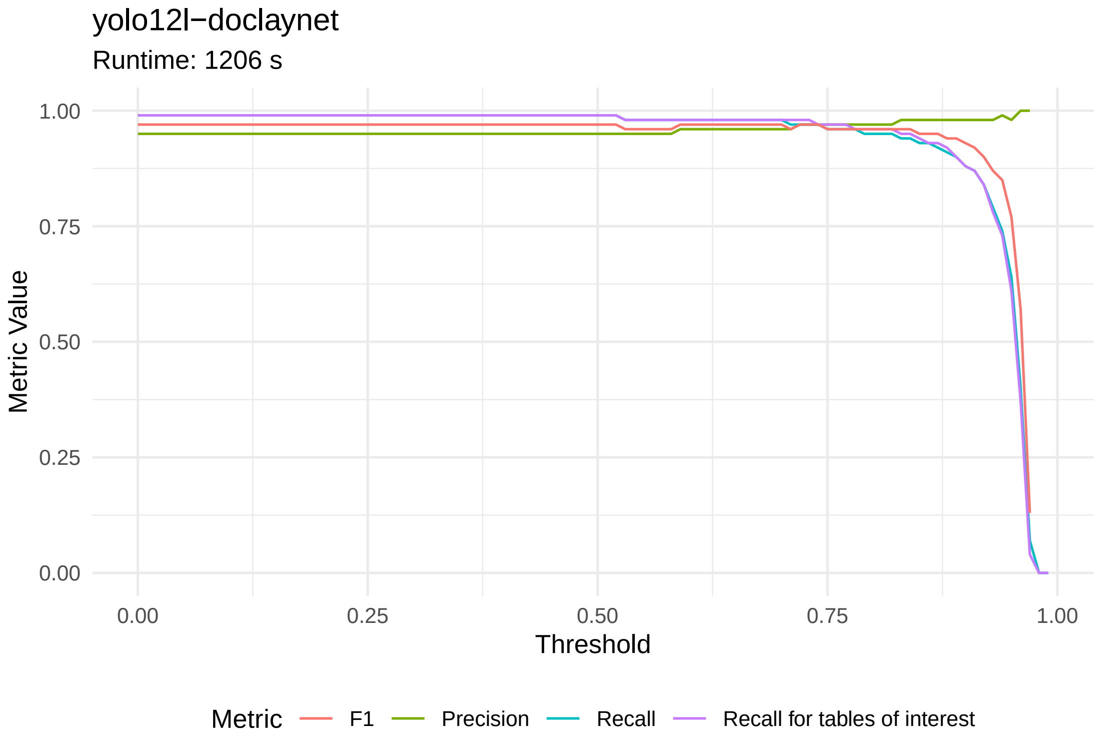
For implementation in a system where the text has to get extracted live or frequently the speed of the library might be paramount. But in special cases it can be important to invest more computational power into text extraction if this assures extraction according a more complicated document layout. 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.

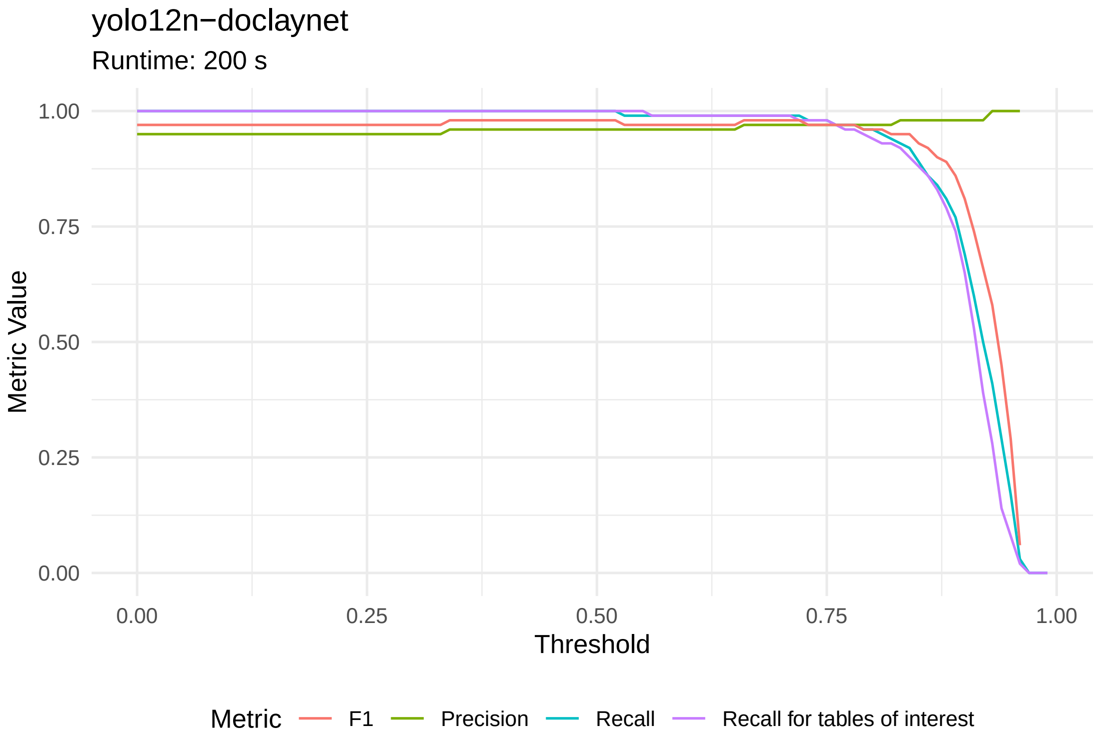
A.2.2 Table detection

- yolo benchmark and table transformer
- skip classification with llm

not so important anymore


microsoft−table−transformer−detection
Runtime: 295 s
Metric Value
1.0
0.9
0.8
0.7
0.00
0.25
0.50
0.75
1.00
Threshold
Metric
F1
Precision
Recall
Recall for tables of interest


yolo12l−doclaynet
Runtime: 1206 s
Metric Value
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
Threshold
Metric
F1
Precision
Recall
Recall for tables of interest



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 Regular expressions

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

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

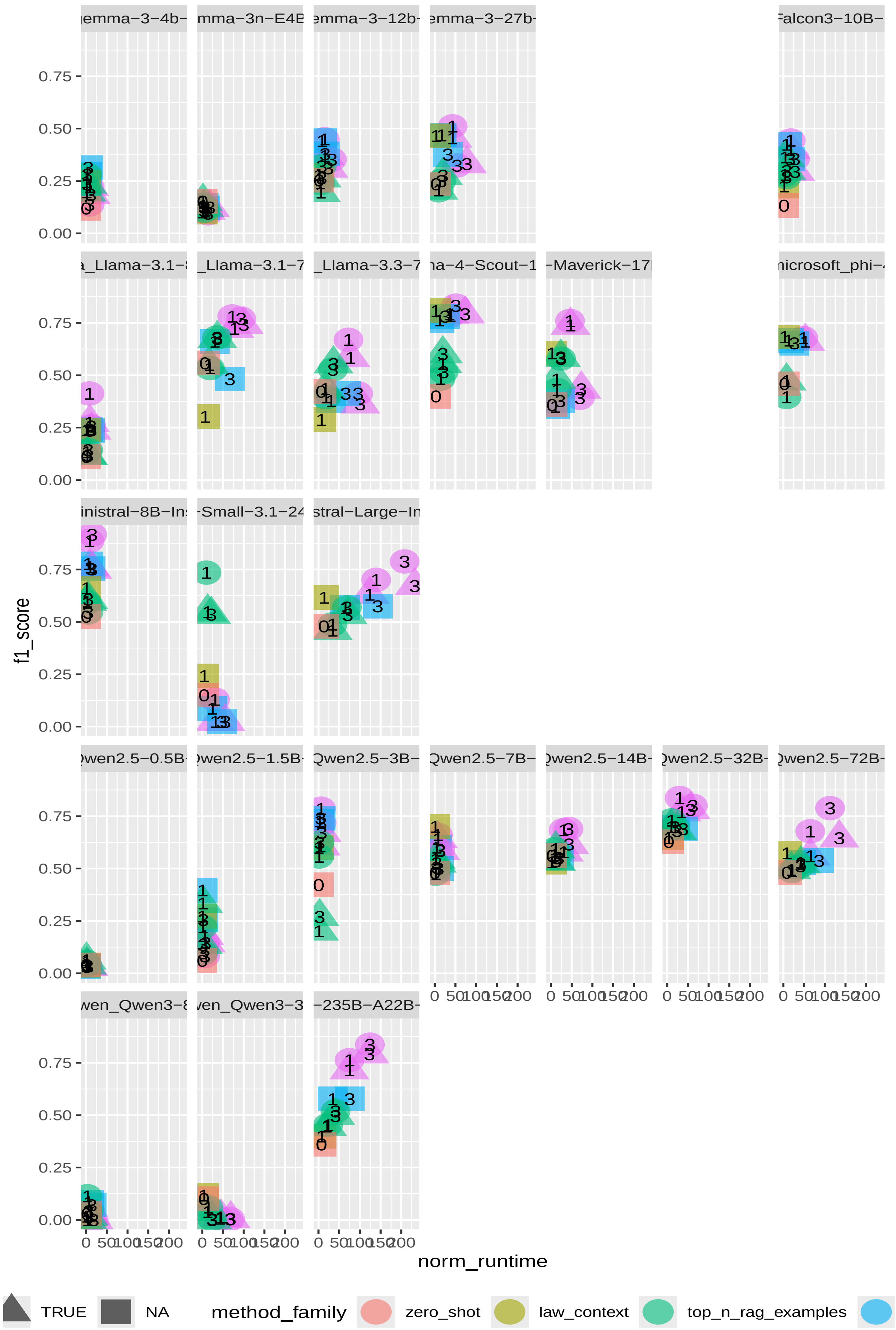
Model parameters (in B)	Transformers	vLLM	vLLM batched
0.5	330	65	NA
3.0	628	130	20
7.0	940	217	30

```
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|Jahresfel
        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.4 Figures



emma−3−4b−
mma−3n−E4E
emma−3−12b−
emma−3−27b−
Falcon3−10B−
_Llama−3.1−8
_Llama−3.1−7
_Llama−3.3−7
na−4−Scout−1
−Maverick−17
nicrosoft_phi−
inistral−8B−Ins
−Small−3.1−24
stral−Large−In
Qwen2.5−0.5B−
Qwen2.5−1.5B−
Qwen2.5−3B−
Qwen2.5−7B−
Qwen2.5−14B−
Qwen2.5−32B−
Qwen2.5−72B−
wen_Qwen3−8
wen_Qwen3−3
−235B−A22B−
f1_score
0.00
0.25
0.50
0.75
norm_runtime
0
50
100
150
200
TRUE
NA
method_family
zero_shot
law_context
top_n_rag_examples

A.5 Extraction framework flow chart

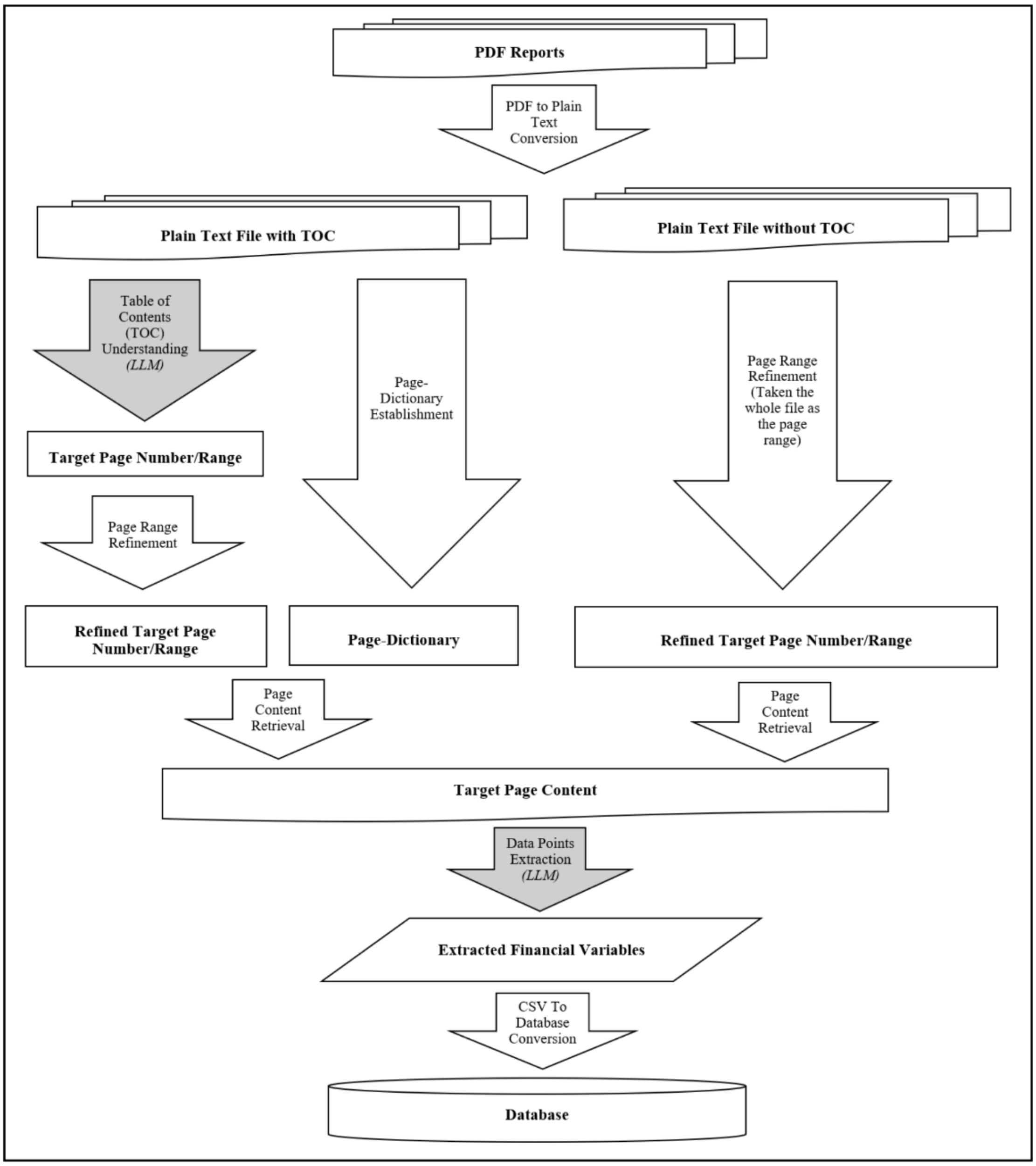

PDF Reports
PDF to Plain Text Conversion
Plain Text File with TOC
Plain Text File without TOC
Table of Contents (TOC) Understanding (LLM)
Page-Dictionary Establishment
Page Range Refinement (Taken the whole file as the page range)
Target Page Number/Range
Page Range Refinement
Refined Target Page Number/Range
Page-Dictionary
Refined Target Page Number/Range
Page Content Retrieval
Page Content Retrieval
Target Page Content
Data Points Extraction (LLM)
Extracted Financial Variables
CSV To Database Conversion
Database

Figure A.1: Framework of