

Suitability of Large Language Models for Making PDF-Documents More Accessible and Barrier-free in Enterprise Content Management



Jack Heseltine

JKU/FAW/IML Masters Examination Presentation September 29th, 2025

Presenting today

- Master Thesis Work
 - Seminar: In-context Learning Papers
 - Practical Work: ECM Integration Module basics
 - **Thesis: ECM Integration Module development, LLM Fine-tuning & related experiments**
- Topic: LLMs to **Generate PDF Source Code** (Representation Format) with Annotations, Tags, ... that make the file more readable for screenreader interpretation
- Legal relevance in 2025, generally an ECM topic/was considered in this (technical) context especially - some notes will follow
- Project assumptions subscribe to this formula [1]:

Accessibility + Usability + User Centered Design = Quality for All

- [1, p.37]: Klaas Posselt and Dirk Frölich. 2019. Barrierefreie PDF-Dokumente erstellen. ISBN: 978-3-86490-487-5.

Basics/Motivation

Inclusion, access, barrier-freedom. Barrier-freedom is mostly used in German language settings and means the “creation of a context that allows people the equal-rights, unhindered access to all areas of life” [1, p. 33] and is therefore crucial for real inclusion, the “independent, equal-rights participation of all people in social life” [1, p. 34], though it goes further than just the social sphere: access is the actual mechanism by which inclusion and barrier-freedom take place, in the present author’s definition. In the technological context, accessibility is mediated and extended by usability and user-centered design, introducing the fundamental aspect of quality, so we would also subscribe to the formula

Accessibility + Usability + User Centered Design = Quality for All

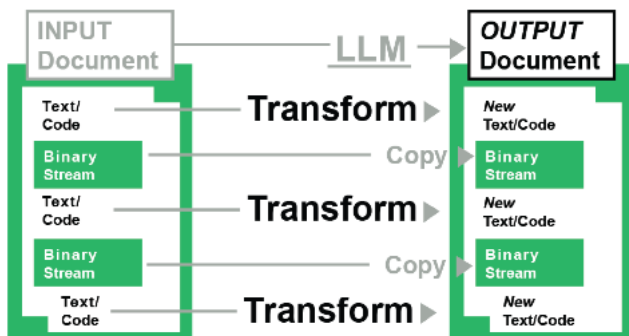
- [1, p. 37]: Klaas Posselt and Dirk Frölich. 2019. Barrierefreie PDF-Dokumente erstellen. ISBN: 978-3-86490-487-5.

Outline

- Core Challenge/Problem — Why is this an ML Topic?
- Related Work
- The Setting and Technical Situation
- Current Legal Context
- ECM Implementation (Part I - Not Focus)
- In-context Learning, Fine-tuning and Meta-Information Approaches (Part II - **Focus**)
- Disadvantages of the Chosen Approaches, (Current Work & Benchmark:) Final Experiments to Improve Scores
- **Results** for this work
- OOD Metrics
- NLP Measurements Used
- Conclusion and Outlook

Core Challenge/Problem — Why is this an ML Topic?

- Screen Readers: Demo 1
- Document Transformation ... with varying objectives



Potential extra data like checker report

Rule	Status
Accessibility permission flag	Passed
Image-only PDF	Failed
Tagged PDF	Failed
Logical Reading Order	Needs manual check
Primary language	Failed
Title	Failed
Bookmarks	Passed
Color contrast	Needs manual check

Table 1: Document-Content Accessibility by Rule

Accessibility Report

Summary

The checker found problems which may prevent the document from being fully accessible.

- Needs manual check: 2
- Passed manually: 0
- Failed manually: 0
- Skipped: 0
- Passed: 11
- Failed: 19

Detailed Report

Document

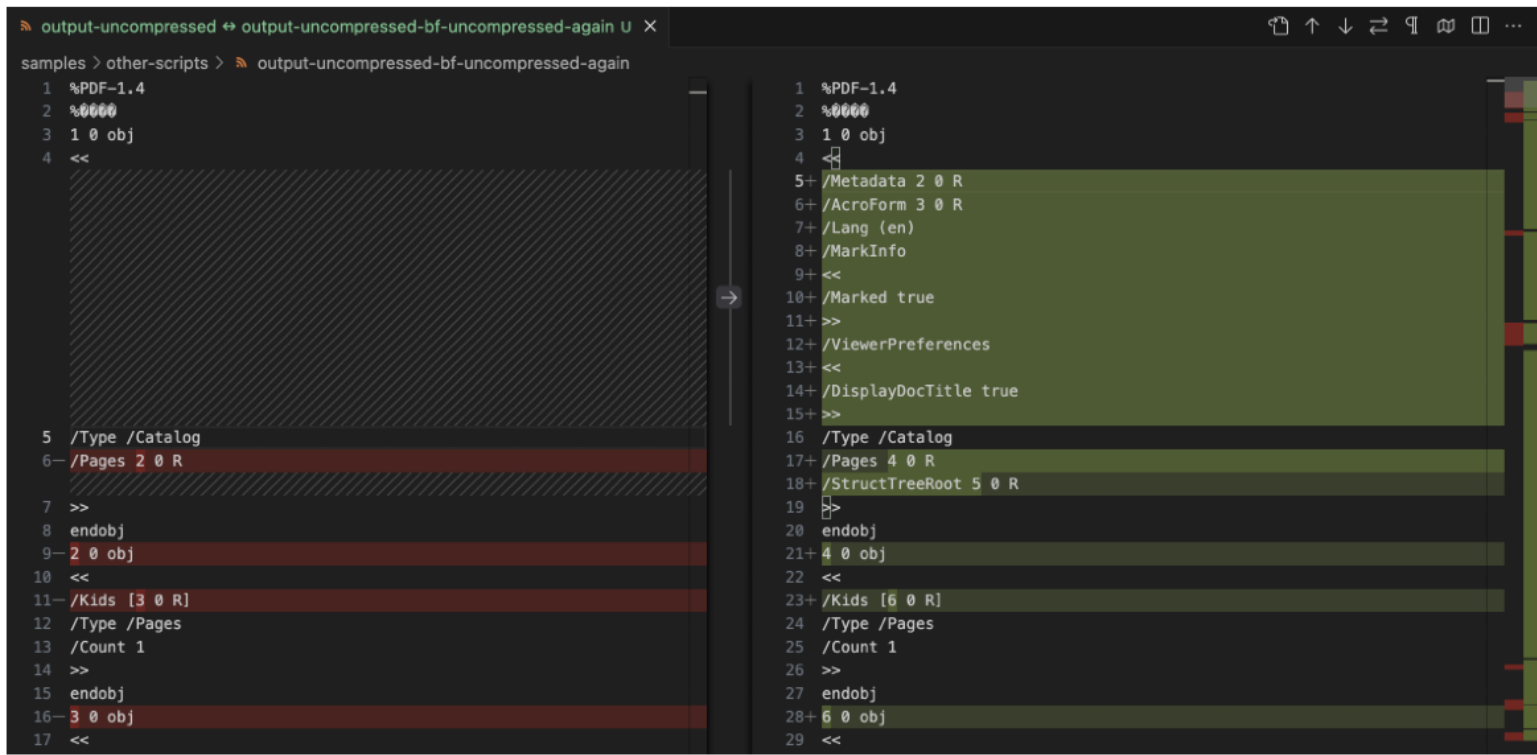
Rule Name	Status	Description
Accessibility permission flag	Passed	Accessibility permission flag must be set
Image-only PDF	Failed	Document is not image-only PDF
Tagged PDF	Failed	Document is tagged PDF
Logical Reading Order	Needs manual check	Document structure provides a logical reading order
Primary language	Failed	Text language is specified
Title	Failed	Document title is showing in title bar
Bookmarks	Passed	Bookmarks are present in large documents
Color contrast	Needs manual check	Document has appropriate color contrast

Page Content

Rule Name	Status	Description
Tagged content	Failed	All page content is tagged
Tagged annotations	Passed	All annotations are tagged
Tab order	Failed	Tab order is consistent with structure order
Character encoding	Passed	Reliable character encoding is provided
Tagged multimedia	Passed	All multimedia objects are tagged
Screen flicker	Passed	Page will not cause screen flicker
Scripts	Passed	No inaccessible scripts
Timed responses	Passed	Page does not require timed responses
Navigation links	Passed	Navigation links are not repetitive

Core Challenge/Problem — Why is this an ML Topic?

- Code Generation Perspective ...



```
output-uncompressed ↔ output-uncompressed-bf-uncompressed-again U x
samples > other-scripts > output-uncompressed-bf-uncompressed-again

1 %PDF-1.4
2 %0000
3 1 0 obj
4 <<
5 /Type /Catalog
6 /Pages 2 0 R
7 >>
8 endobj
9 2 0 obj
10 <<
11 /Kids [3 0 R]
12 /Type /Pages
13 /Count 1
14 >>
15 endobj
16 3 0 obj
17 <<

1 %PDF-1.4
2 %0000
3 1 0 obj
4 <<
5+ /Metadata 2 0 R
6+ /AcroForm 3 0 R
7+ /Lang (en)
8+ /MarkInfo
9+ <<
10+ /Marked true
11+ >>
12+ /ViewerPreferences
13+ <<
14+ /DisplayDocTitle true
15+ >>
16 /Type /Catalog
17+ /Pages 4 0 R
18+ /StructTreeRoot 5 0 R
19 >>
20 endobj
21+ 4 0 obj
22 <<
23+ /Kids [6 0 R]
24 /Type /Pages
25 /Count 1
26 >>
27 endobj
28+ 6 0 obj
29 <<
```

Core Challenge/Problem — Why is this an ML Topic?

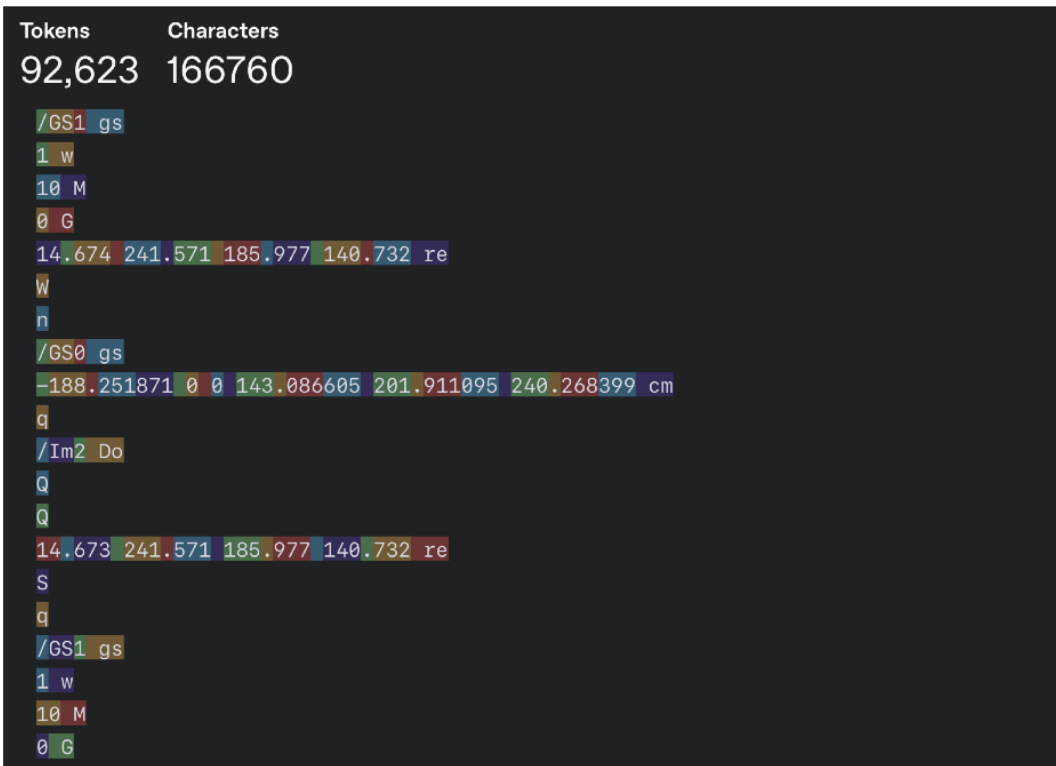
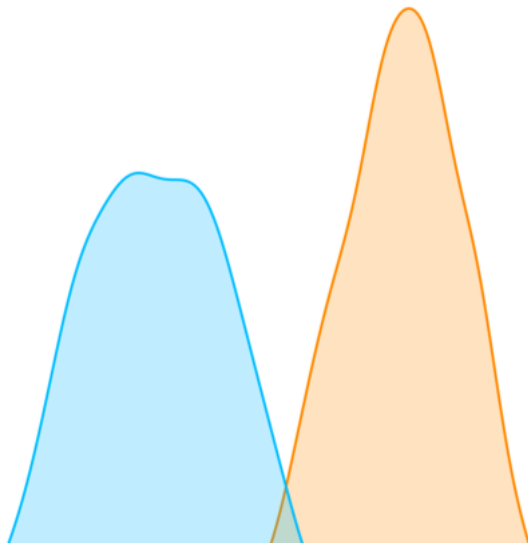
- **Code Generation Perspective ...**
 - **Considering PDF file source code or structured representation**
 - not arbitrary byte sequences including binary, which is found in PDF files
- **Encoding barrier:** LLM interfaces are designed to output text, not raw binary. Raw binary tends to be interpreted as UTF-8/ASCII text, which often shows up as garbled symbols. Direct binary output is usually corrupted unless wrapped in a safe encoding (Base64, hex)
- **Tokenization limits:** LLMs don't think in bytes, but in tokens. A model can try to produce sequence of tokens that looks binary, but whether it is byte-perfect is another matter
 - For structured binaries (e.g., PDF, ZIP, ELF executable), a single incorrect byte breaks the file

Related Work

- **Machine translation:** “conversion from **visual to logical format** can be seen as language task” and use case “LATEX Tagged PDF” project [6a]
 - Document is “rewritten” from one style to another more accessible style
 - also include intra-language tasks like summarization and style transfer, generally
 - another example: LLMs have been used to generate alternative text for images in assessing PDF accessibility **evaluation** [6b]
- **Code generation:** in the view the system must produce **structured output**, so this aspect is underlined — not free-form the way translation tasks are typically viewed
 - similar to writing code that encapsulates a document’s semantics
 - *CodeA11y*: AI coding assistant for suggesting accessibility-compliant code [6c]
 - LaTeX-to-HTML [6d] — **practical** to produce **intermediary formats** (for accessibility purposes)
- **OCR:** transformer-based *LayoutReader/LayoutLM* (near-perfect reading order prediction) [6e]
 - *Donut* exemplifies skipping the classic OCR text-extraction step (image-to-structured-result) [6f]
 - *SciA11y* also worth mentioning: multiple models combined to produce accessible HTML paper-versions [6g] — again not PDF code produced here

Core Challenge/Problem — Why is this an ML Topic?

- Code Generation Perspective ... Preview: the **Challenge**
 - OOD?



Core Challenge/Problem — Why is this an ML Topic?

- **Theory: Sequence-to-Sequence Task**

$$P(\mathbf{y} | \mathbf{x}) = \prod_{t=1}^n P(y_t | y_{<t}, \mathbf{x})$$

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} P(\mathbf{y} | \mathbf{x})$$

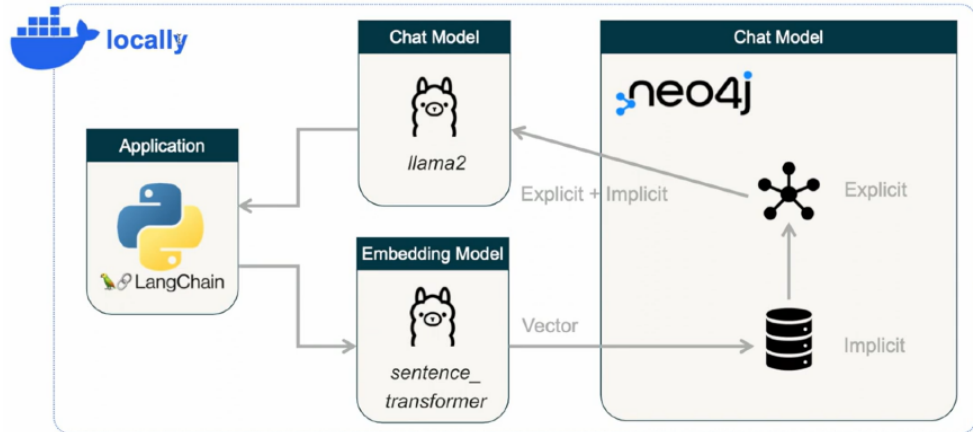
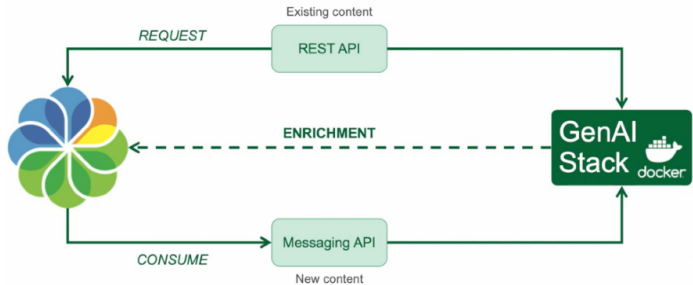
$$\mathcal{L} = - \sum_{t=1}^n \log P(y_t | y_{<t}, \mathbf{x})$$

- **Models** and Prompt Engineering Analysis:
- Causal LLMs, decoder-only transformers
 - Later tests of o3/R1, “Reasoning LLMs” which introduce extra training regimes/ intermediate “scratchpad” results

Model	Observation using Prompt B.3
Mistral (used so far)	See Table 6.1, Prompt B.3.
Llama3	Comparable to Mistral - LLM does generate code but rather a high level description and interpretation of the task. Generates code, consulted for comparison: this is promising, finally. Elides certain content still, however, including in the output lines like ... (content truncated for brevity) ..., for example. Code outputs worked directly (no descriptions like Llama 3 for instance) but there was looping without reaching an end of the document code observed, in three of the five experimental runs. Examples up to error messages by the client were included in the outputs collection. Inference run-times were all above five minutes, further suggesting uncontrolled looping.
ChatGPT o3 via Web Client/Chat	
Gemini 2.5 Pro via Web Client/Google AI Studio	
Llama3.3	Similarly to Llama3, does not generate code but rather a high level description and interpretation of the task. (Not a single code file was generated even by this advanced base model, suggesting a different focus of the model and/or training data. This is not analyzed in detail as part of this work.)
Llama3.1	Similar to Llama3 and 3.3.
DeepSeek-R1:1.5b	Similar again.
DeepSeek-R1(-0528):8b	Responses include thought blocks as shown in the main text, but responses are comparable to other models: no direct PDF source code is prodced in any of the five examples as it turns out.

The Setting and Technical Situation

- Cheaper, Big LLMs; API Vendors
- Not of Interest for this Project
 - OCR mainly
- ECM: fully on-premises vs API integration
 - Overview of the Solution Design

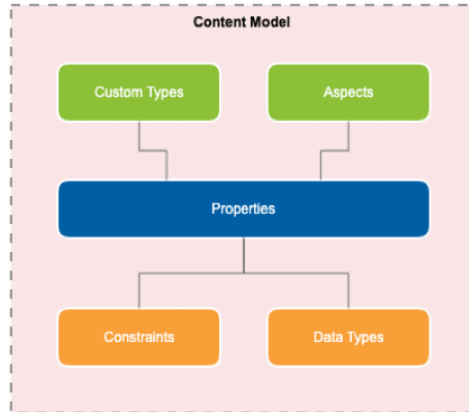


Current Legal Context

- **European Accessibility Act (EAA)** (Directive 2019/882/EU), building on:
 - Web Accessibility Directive (2016/2102/EU) [2a]
 - European Public Procurement Directives (2014/24/EU [2b] and 2014/25/EU [2d])
 - European Electronic Communications Code (2018/1972/EU) [2c]
- EAA explicitly applies to a wide range of products placed on the market after 28 June 2025
- On the service side, obligations cover telecommunications, audiovisual media services, passenger transport (websites, apps, e-tickets, information systems, self-service terminals), consumer banking, e-books and dedicated software, and e-commerce services
- requirements directly link to standards such as **WCAG** and **PDF/UA**
- various harmonized European standards EN for aligning products and services now
- for **ECM**: platforms evolve from passive storage and retrieval systems to active guardians of compliance for baseline of accessibility as required for regulated products and services

ECM Implementation (Part I)

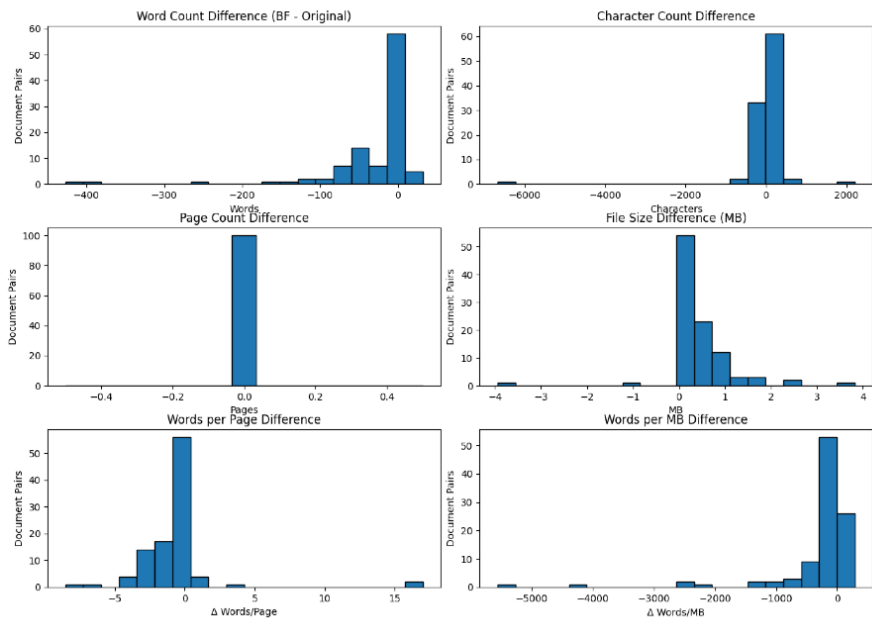
- *Brief Demo of Alfresco: Demo 2*
- **Outline:** Integration into the Content Model of a simple Double-Loop LLM Call Routine, additionally prepare a Accessibility Checker Report for a meta-informational approach
- Potential value for **future work** in this domain as a relatively solid **platform**
 - Working with PDFs is intuitive and clear
 - Relevant API integrations



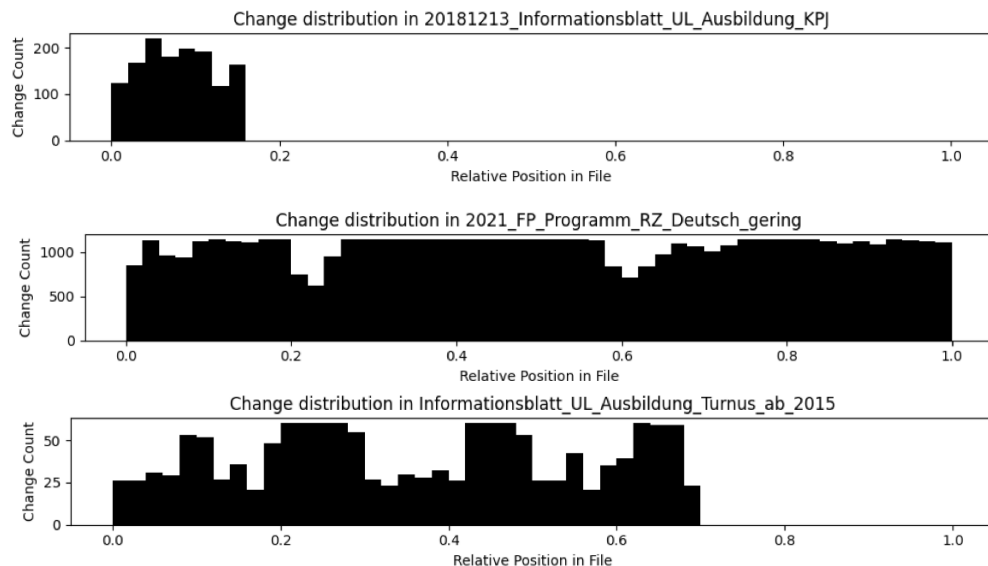
In-context Learning, Fine-tuning and Meta-Information Approaches (Part II - Focus)

- Data: Non-accessible and accessible PDF counterparts (**Fine-tuning dataset**, 100 pairs)

Differences Between Original and Accessible PDF Versions



Smaller **Test set** of 12 pairs is publishable



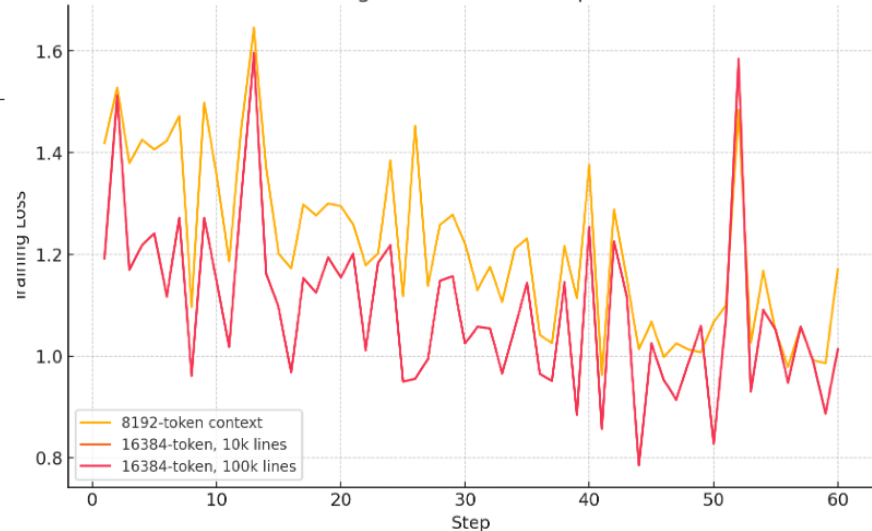
Method: Fine-tuning

- Best prompt was used to test with different fine-tuned models
- Base-models: DeepSeek-R1-0528:8b and Llama3.1
- Tools for Fine-tuning: Unsloth (like HuggingFace Transformer Library) & Alpaca/Ollama
- Fine-tuning Approach: PEFT and LoRA

Model and PEFT Configuration

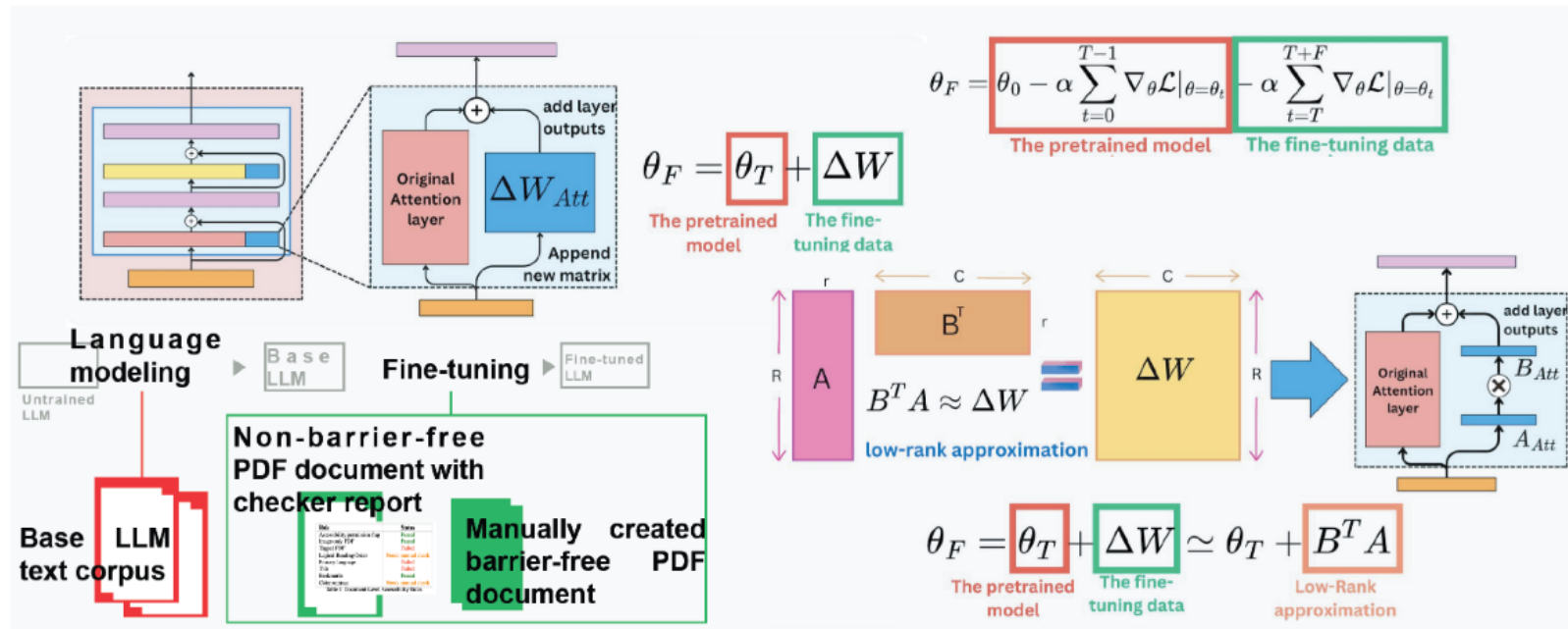
```
1 tokenizer = AutoTokenizer.from_pretrained("meta-llama/Llama-3.1-8B-Instruct")
2 model = FastLanguageModel.from_pretrained("unsloth/Meta-Llama-3.1-8B-Instruct-bnb-4bit")
3 model = FastLanguageModel.get_peft_model(
4     model,
5     r=16,
6     target_modules=["q_proj", "k_proj", "v_proj", "o_proj", "gate_proj", "up_proj",
7     "down_proj"],
8     lora_alpha=16,
9     lora_dropout=0.0,
10    bias="none",
11    use_gradient_checkpointing="unsloth",
```

Training Loss Curves Comparison



Meta-Information Report-Addition, Fine-tuning with LoRA

- parameter-efficient fine-tuning (PEFT) strategy centered on Low-Rank Adaptation (LoRA) [4]



Disadvantages of the Chosen Approaches, (Current Work & Benchmark:) Final Experiments to Improve Scores

- **Fine-tuning including accessibility reports** to test adding prompt meta-information
- Observations: OOD? Small token sizes, LLM repetition loops? Suggesting uncertainty?
- Method:
 - Load report
 - Build one bigger input string
 - Wrap into Alpaca prompt with instruction and reference
 - SFT training: loss applied on response

Forms

Rule Name	Status	Description
Tagged form fields	Passed	All form fields are tagged
Field descriptions	Passed	All form fields have description

Alternate Text

Rule Name	Status	Description
Figures alternate text	Failed	Figures require alternate text
Nested alternate text	Failed	Alternate text that will never be read
Associated with content	Failed	Alternate text must be associated with some content
Hides annotation	Failed	Alternate text should not hide annotation
Other elements alternate text	Failed	Other elements that require alternate text

Tables

Rule Name	Status	Description
Rows	Failed	TR must be a child of Table, THead, TBody, or TFoot
TH and TD	Failed	TH and TD must be children of TR
Headers	Failed	Tables should have headers
Regularity	Failed	Tables must contain the same number of columns in each row and rows in each column
Summary	Failed	Tables must have a summary

Lists

Rule Name	Status	Description
List items	Failed	LI must be a child of L
Lb1 and LBody	Failed	Lb1 and LBody must be children of LI

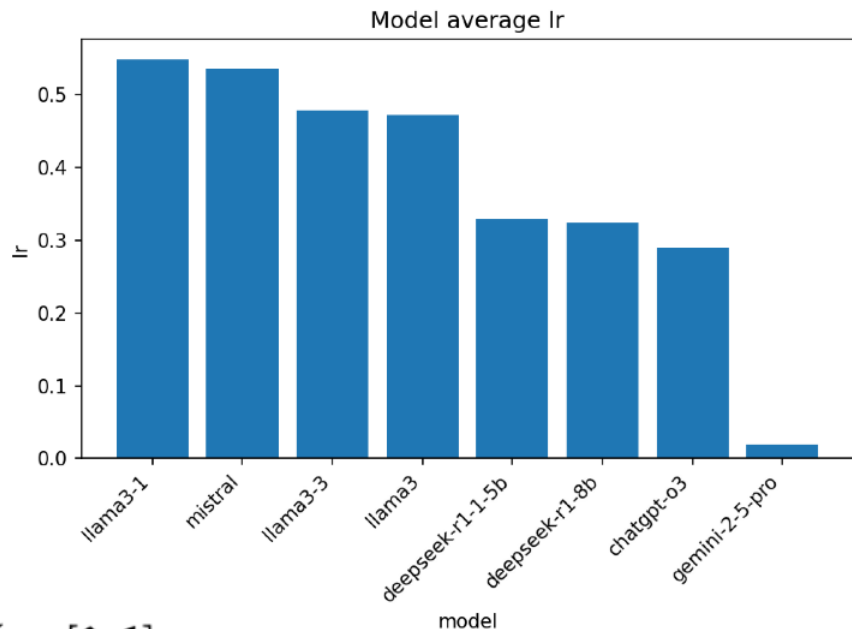
Headings

Rule Name	Status	Description
Appropriate nesting	Failed	Appropriate nesting

Results

- We start by looking at edit distances
- **Model-average LR** (Levenshtein Ratio, higher is better): Locally run models ranked by normalized edit similarity to references.
- LR summarizes closeness after accounting for length; higher bars indicate candidates that are closer in edit space.
 - We will come to metrics in detail

(Without meta-info)



$$\text{Levenshtein Ratio (LR)} = \frac{|R| + |C| - d}{|R| + |C|} \in [0, 1]$$

Given reference R and candidate C with Levenshtein distance $d = \text{lev}(R, C)$ and lengths

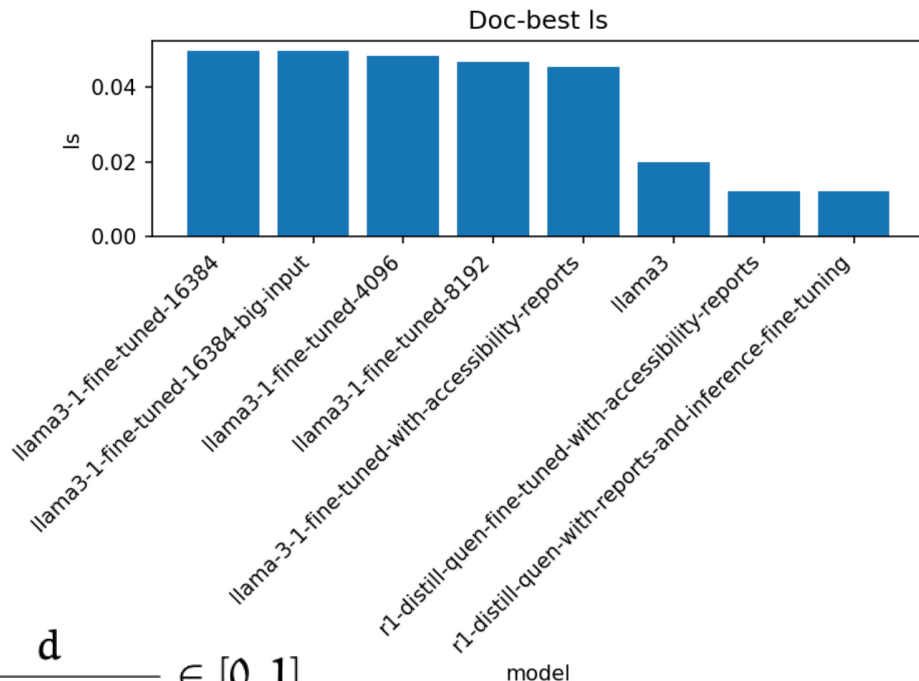
Results

- BLEU/ROUGE-L/METEOR are ~0–0.06 (very low)
- CER/WER are ~0.83–0.99 (very high → bad)
- Length: hyp_length_tokens is usually 5–20× smaller than ref_length_tokens (e.g., 4–611 vs 428–15k) — which is to be expected to a degree
 - Need to understand scoring between models better to get at this
 - But first: **are these model certain of what they are producing, when hypothesis/references do not match well in the end?**

$$\text{Levenshtein Similarity (LS)} = 1 - \frac{d}{\max(|R|, |C|)} \in [0, 1]$$

With meta-info

- Similar results for this experiment

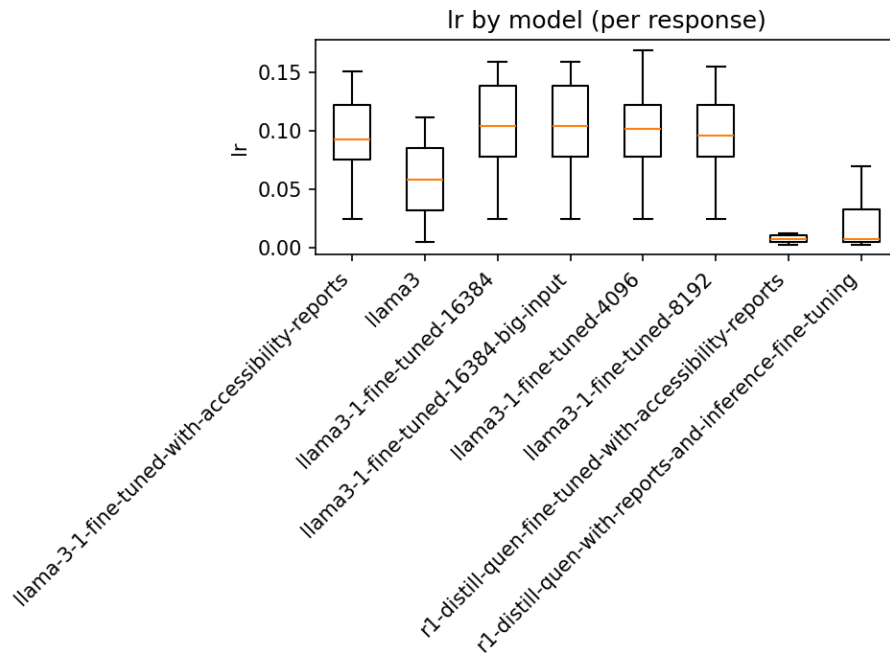


Results

- BLEU/ROUGE-L/METEOR are ~0–0.06 (very low)
- CER/WER are ~0.83–0.99 (very high → bad)
- Length: hyp_length_tokens is usually 5–20× smaller than ref_length_tokens (e.g., 4–611 vs 428–15k) — which is to be expected to a degree
 - Need to understand scoring between models better to get at this
 - But first: **are these model certain of what they are producing, when hypothesis/references do not match well in the end?**

With/without meta-info

- Fairly consistently similar results



OOD-Metrics: Background

$$\mathcal{L} = -\frac{1}{N} \sum_{t=1}^N \log p(x_t \mid x_{<t}) \quad (\text{average cross-entropy loss})$$

$$\text{PPL} = \exp(\mathcal{L}) \quad (\text{perplexity})$$

- **Perplexity:** sequence-level measure derived from average negative log-likelihood
 - Lower values \Rightarrow model finds the sequence highly probable (more confident)
 - Higher values \Rightarrow model finds the sequence unlikely (more uncertain)
- **Conditional vs. Unconditional Scoring**
 - Self-likelihood: scoring the completion alone tends to be optimistic
 - Conditional perplexity: score completion while conditioning on the prompt (ignoring the prompt in the loss)
- **Token-Level Uncertainty Signals**
 - Predictive entropy (distribution spread over the next token)
higher \Rightarrow more uncertainty; lower \Rightarrow more confidence
 - Top-1 probability: simple proxy for confidence at each step

$$H_t = - \sum_{v \in V} p(v \mid x_{<t}) \log p(v \mid x_{<t})$$

$$\bar{H} = \frac{1}{N} \sum_{t=1}^N H_t$$

OOD-Metrics: Chosen Approach and Measured Perplexity

1. Generate completion (greedy for determinism or sampled for probing)
2. Compute conditional perplexity on the completion (prompt masked out of the loss)
3. Compute mean entropy and mean top-1 probability across completion steps, using next-token logits at each step

Conditional perplexity for the top-performing models so far (completion only) gets values $\sim 1-1.5$, which is **extremely low, i.e. the model is very confident**/the tokens were highly predictable — but low perplexity \neq good output (we see loops, repetition, filler, not real PDF object code)

Mode collapse/degenerate loop: When the model falls into a repeating pattern (e.g., "BT ... ET BT ... ET" forever), the next token is very predictable — known to occur during fine-tuning, as the model learns to generate text that accomplishes the specific task, but loses ability to generate other forms of text.

Low perplexity reflects **predictability**, not quality: also called the likelihood trap [5]

[5] Zhang et al. 2020. Trading Off Diversity and Quality in Natural Language Generation.

Speaking of Scores: NLP Measurements Used

- BLEU

$$\text{BLEU} = \text{BP} \exp\left(\sum_{n=1}^4 w_n \log p_n\right) \quad \text{BP} = \begin{cases} 1, & \text{if } c > r \\ \exp\left(1 - \frac{r}{c}\right), & \text{if } c \leq r \end{cases}$$

- ROUGE

$$\text{ROUGE-N} = \frac{\sum_{g_n \in \text{Ref}} \min(\text{count}_C(g_n), \text{count}_R(g_n))}{\sum_{g_n \in \text{Ref}} \text{count}_R(g_n)}$$

- METEOR

$$\text{METEOR} = F_{\text{mean}} (1 - \text{Pen}), \quad F_{\text{mean}} = \frac{10PR}{R + 9P}$$

Speaking of Scores: NLP Measurements Used

- Edit Distance (Levenshtein)

$$\text{lev}(a, b) = \begin{cases} |a| & \text{if } |b| = 0, \\ |b| & \text{if } |a| = 0, \\ \text{lev}(\text{tail}(a), \text{tail}(b)) & \text{if } \text{head}(a) = \text{head}(b), \\ 1 + \min\{\text{lev}(\text{tail}(a), b), \text{lev}(a, \text{tail}(b)), \text{lev}(\text{tail}(a), \text{tail}(b))\} & \text{otherwise.} \end{cases}$$

- and Combinations —

- LS and LR were plotted so far

Given reference R and candidate C with Levenshtein distance $d = \text{lev}(R, C)$ and lengths

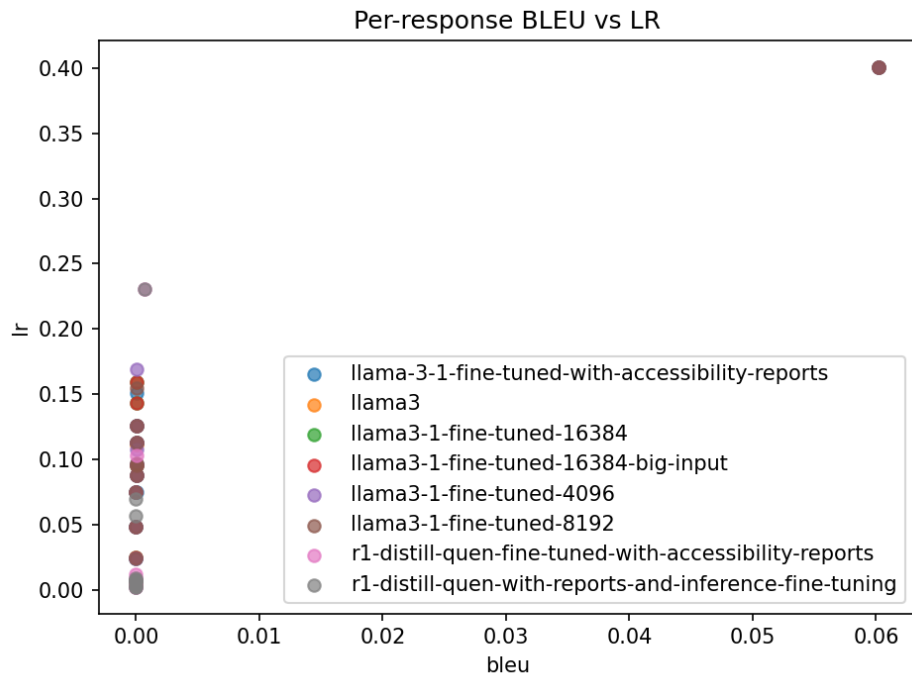
$$\text{CER} = \frac{d}{|R|}, \quad \text{WER} = \frac{d_w}{|R_w|} \quad (\text{edit distance at token level}),$$

$$\text{Levenshtein Similarity (LS)} = 1 - \frac{d}{\max(|R|, |C|)} \in [0, 1],$$

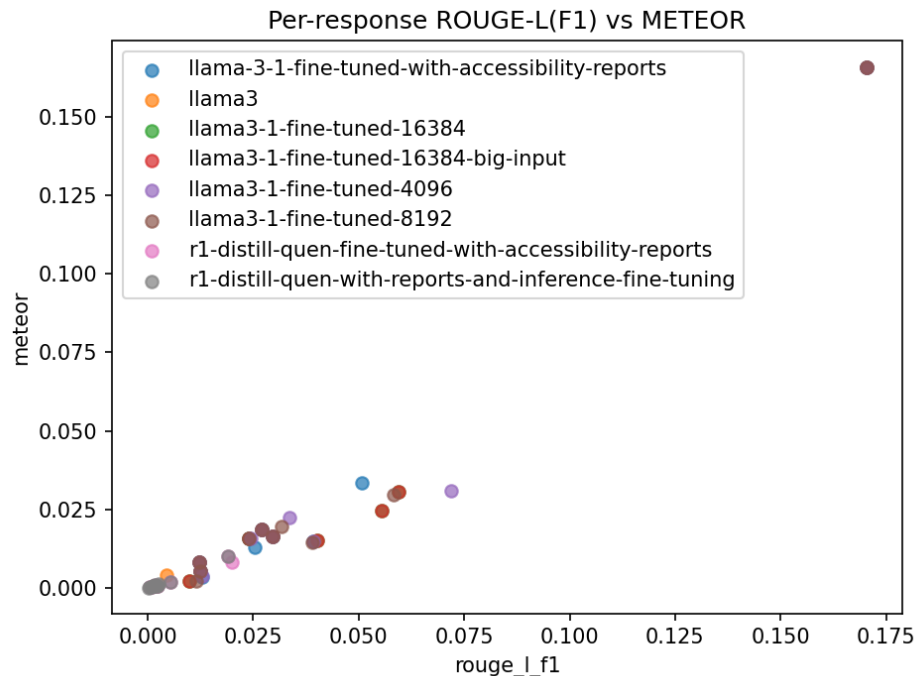
$$\text{Levenshtein Ratio (LR)} = \frac{|R| + |C| - d}{|R| + |C|} \in [0, 1].$$

Results

- Best model here?



Both With/without meta-info



Conclusion and Outlook

- **Conclusion:** positional logic and other sub-token numerical data pose an inference challenge to the chosen class of (fine-tuned) LLMs, despite low perplexity/high certainty
 - NLP-specific metrics for measuring reference similarities of the hypothesis documents were referenced to measure quality of the output, trend based on model complexity was observed, but no improvements when using meta-information in training and inference
- **Outlook:** nuanced problem with potentially large payoff, so it might be worth:
 - (Tangent:) Exploring accessibility scoring via neural network
 - Finding ways to break down the task of PDF code generation
 - Testing future or current, but more complex, models
 - Adding test document set domains like the one introduced with this work
- **Contribution:** ECM platform, basic methodology proposal, basic model testing/observations

Summary

We considered:

- Core Challenge/Problem — Why is this an ML Topic?
- Related Work
- The Setting and Technical Situation
- Current Legal Context
- ECM Implementation (Part I - Not Focus)
- In-context Learning, Fine-tuning and Meta-Information Approaches (Part II - **Focus**)
- Disadvantages of the Chosen Approaches, (Current Work & Benchmark:) Final Experiments to Improve Scores
- **Results** for this work
- Speaking of Scores: NLP Measurements Used
- OOD Metrics
- Conclusion and Outlook

References

- [1]: Klaas Posselt and Dirk Frölich. 2019. Barrierefreie PDF-Dokumente erstellen. ISBN: 978-3-86490-487-5.
- [2]
 - a) 2015. Vorschlag für eine RICHTLINIE DES EUROPÄISCHEN PARLAMENTS UND DES RATES zur Angleichung der Rechts- und Verwaltungsvorschriften der Mitgliedstaaten über die Barrierefreiheitsanforderungen für Produkte und Dienstleistungen, de. (2015). Retrieved 08/20/2025 from <https://eur-lex.europa.eu/legal-content/DE/TEXT/?uri=COM%3A2015%3A615%3AFIN>
 - b) 2016. Richtlinie (EU) 2016/2102 des Europäischen Parlaments und des Rates vom 26. Oktober 2016 über den barrierefreien Zugang zu den Websites und mobilen Anwendungen öffentlicher Stellen (Text von Bedeutung für den EWR). de. (October 2016). Retrieved 08/20/2025 from <http://data.europa.eu/eli/dir/2016/2102/oj/deu>
 - c) [n. d.] Richtlinie (EU) 2018/1972 des Europäischen Parlaments und des Rates vom 11. Dezember 2018 über den europäischen Kodex für die elektronische Kommunikation (Neufassung) Text von Bedeutung für den EWR. de.
 - d) 2014. Richtlinie 2014/25/EU des Europäischen Parlaments und des Rates vom 26. Februar 2014 über die Vergabe von Aufträgen durch Auftraggeber im Bereich der Wasser-, Energie- und Verkehrsversorgung sowie der Postdienste und zur Aufhebung der Richtlinie 2004/17/EG Text von Bedeutung für den EWR. de. (February 2014). Retrieved 08/20/2025 from <http://data.europa.eu/eli/dir/2014/25/oj/deu>
- [3] Damien Benveniste. 2024. Understanding How LoRA Adapters Work! en. (November 2024). Retrieved 07/10/2025 from <https://newsletter.theaiedge.io/p/understanding-how-lora-adapters-work>
- [4] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs]. (October 2021). doi: 10.48550/arXiv.2106.09685. Retrieved 07/04/2025 from <http://arxiv.org/abs/2106.09685>
- [5] Zhang et al. 2020. Trading Off Diversity and Quality in Natural Language Generation. <https://arxiv.org/abs/2004.10450>
- [6]
 - a) Frank Mittelbach et al. 2024. Automatically producing accessible and reusable PDFs with LATEX. <https://dl.acm.org/doi/10.1145/3685650.3685670>
 - b) Anukriti Kumar. 2025. Benchmarking PDF Accessibility Evaluation. https://lhwang.net/assets/pdf/2025_kumar_a11ybenchmark_assets.pdf
 - c) Peya Mowar at al. 2025. CodeA11y: Making AI Coding Assistants Useful for Accessible Web Development. <https://arxiv.org/abs/2502.10884>
 - d) Gerd Kortemeyer. 2023. Using artificial-intelligence tools to make LaTeX content accessible to blind readers. <http://arxiv.org/abs/2306.02480>
 - e) Wang et al. 2021. LayoutReader: Pre-training of Text and Layout for Reading Order Detection. <https://arxiv.org/abs/2108.11591>
 - f) Kim et al. 2022. OCR-free Document Understanding Transformer. <https://arxiv.org/abs/2111.15664>
 - g) Wang et al. 2021. Improving the Accessibility of Scientific Documents. <https://arxiv.org/abs/2105.00076>