

Peer Review

Review of: "Arabic-Nougat: Fine-Tuning Vision Transformers for Arabic OCR and Markdown Extraction"

Evgenii Arsentev¹

1. Independent researcher

1. What the work claims

The paper adapts Meta's Nougat (a Donut/Swin vision encoder with an mBART decoder) to Arabic book pages and releases three checkpoints: arabic-small-nougat, arabic-base-nougat, and arabic-large-nougat. The large model swaps in the Aranizer-PBE-86k tokenizer, which the author says packs roughly four base-Nougat tokens into one (§3.2), and raises the decoder limit to 8192 tokens, described as a practical context of about 32k base tokens (§3.5). Training data is arabic-img2md, 13.7k synthetic pairs made by scraping Hindawi HTML, rendering it to PDF, and extracting Markdown (§3.3). On a 160-page held-out set from the same dataset (§4.1), all three fine-tuned models beat the untuned Meta checkpoints by a wide margin (Table 1). A by-product is arabic-books, a corpus of 1.1 billion Arabic tokens from over 8,500 books, extracted with the best model.

I think the most valuable thing here is practical. Arabic document OCR into structured text has few open end-to-end tools, and the author has put the models, the training set, the extracted corpus, and the code on public repositories (references [6], [18], and the GitHub link in the abstract). The Limitations section is also candid: hallucinated URLs and images in the small model, repetition in the larger ones, and the narrow Hindawi domain are all named by the author before a reviewer has to raise them.

2. Whether the conclusion holds on the data

The central claim, that fine-tuning on Arabic data turns an unusable model into a usable one, is well supported. In Table 1, Nougat Base (Meta) has BLEU 0.0094 and CER 1.3798, while Arabic Base Nougat has BLEU 0.6367 and CER 0.0926. A gap that large will not disappear under any reasonable change of the test set. I have no doubt about the direction of the result.

The finer claims are harder to support from what is shown.

Only one kind of baseline. The comparison in Table 1 is against Nougat checkpoints that were never trained on Arabic, so it mainly measures "trained on Arabic" versus "not trained on Arabic." It would help a great deal to add at least one system that already reads Arabic: Tesseract with the Arabic model is the obvious choice, and one of the general end-to-end parsers the author cites in §2.4 (Donut, GoT) would be a useful second comparison, even though §2.4 says they mainly target English. Without that, the abstract's phrase "significantly outperform existing methods" is broader than the evidence.

The test set is in-distribution. The 160 pages come from arabic-img2md itself (§4.1), which is rendered from Hindawi HTML. These are clean, born-digital pages with the same fonts and layout templates as training. I could not find whether the split was made by page or by book. If by page, neighbouring pages of the same book may sit on both sides of the split, which would inflate every number in Table 1. A small set of real scanned pages from another source would tell readers much more about practical accuracy; the Limitations section already points in this direction.

No uncertainty. With 160 pages, differences in the third decimal cannot be read as rankings. Structure Accuracy in Table 1 is 0.9866 (Arabic Small), 0.9834 (Arabic Base), and 0.9884 (Arabic Large), a spread of 0.005. I would not call the large model the best on structure without a confidence interval or a paired bootstrap over pages.

The text does not match Table 1 in several places. These are factual inconsistencies, and I list them plainly.

- §5 states that arabic-large-nougat achieves "the highest Markdown Structure Accuracy (94.7%) and lowest Character Error Rate (6.1%)". Table 1, row Arabic Large Nougat, gives Structure Acc 0.9884 and CER 0.0662, that is 98.84% and 6.62%. Neither 94.7% nor 6.1% appears anywhere else in the paper.
- §5 also says the large model is "surpassing its smaller counterparts". On BLEU, it is not: Table 1 gives 0.7565 (Arabic Small) against 0.6771 (Arabic Large). On WER, the large model is the worst of the three fine-tuned models: 0.1916, against 0.1523 (Small) and 0.1042 (Base).
- §4.5 recommends arabic-base-nougat for general text extraction. By Table 1, it has the lowest BLEU of the three (0.6367) and the highest CER (0.0926), though the lowest WER. The recommendation may well be right for reasons of stability or speed, but the reason should be stated.
- §4.1 lists the metrics as MSA, CER, and TER. Table 1 reports BLEU, CER, WER, and Structure Acc. TER is never tabulated; it appears only as two numbers in §4.4 (1.05 for large, 1.25 for small), with no value for

the base model.

CER and WER above 1. Table 1 gives CER 2.8849 and WER 3.0748 for Nougat Small (Meta). Rates above 1 are possible when the model inserts far more text than the reference (runaway repetition would do it), but a reader needs to be told this, and told whether the error rate was clipped, how empty or truncated outputs were counted, and whether Markdown syntax was stripped before CER was computed.

An unexplained pattern. The large model has the lowest CER (0.0662) and the highest WER (0.1916) of the fine-tuned models. That combination usually means word boundaries or diacritics are being handled differently, or that repetition adds whole spurious words. A short error analysis on a few pages would explain it and would also support the repetition discussion in §6.

The tokenizer efficiency claim is asserted rather than measured. §3.2 says one Aranizer token is "the equivalent of nearly four base Nougat tokens", and §3.5 turns this into $8192 \times \sim 4 \approx 32k$. The arithmetic is consistent, but the ratio itself is not measured on arabic-img2md in the paper. The TER definition in §4.1 ("the ratio of tokens produced by the tokenizer to ground truth tokens") does not say which tokenizer defines the ground-truth count, so I could not connect 1.05 and 1.25 in §4.4 to the factor of four.

I checked the encoder arithmetic in §3.1: 896×672 reduced by 32 in each dimension is 28×21 , and $28 \times 21 = 588$. That part is correct.

3. Reproducibility

The open release is the strongest part of the paper's reproducibility. arabic-img2md and arabic-books are on Hugging Face (references [18] and [6]), the abstract states that all models are open-sourced, and the code is on GitHub, so a reader can at least rerun inference. The learning rate ($1e-4$), the epoch cap (100), early stopping, bfloat16, gradient checkpointing, and Flash Attention 2 are all stated in §3.4.

What I could not find in the Qeios text:

- **Training data for the small model.** §3.3 says arabic-img2md "was exclusively used to train arabic-base-nougat and arabic-large-nougat". The small model's data is not described, and §6 only says it had a "smaller training dataset". Since the small model has the best BLEU in Table 1, this matters.
- **How the large model was built.** §3.1 calls it "a new model with an expanded decoder". I could not tell which weights were used for initialization, how the new embedding matrix for the 86k vocabulary was initialized, or how many parameters each of the three models has.

- **Split construction.** Train, validation, and test sizes, the split unit (page or book), and the seed. The early-stopping metric and patience are also missing.
- **Hardware and schedule.** "Multiple GPUs" (§3.4) with no type or count, no batch size, no optimizer, no warm-up or decay, and no number of epochs actually run before early stopping.
- **Decoding settings used for Table 1.** §4.5 recommends a repetition penalty above 1, but I could not tell whether Table 1 was produced with such a penalty, with which value, and with greedy or beam decoding. The maximum generation length per model is also needed because the small model's limit is given as 2048 in §3.1 while §3.5 gives 3584 for nougat-small.
- **Metric implementation.** Which BLEU (sacreBLEU or other, which tokenization for Arabic), how CER/WER treat Markdown symbols and diacritics, and the formula for the custom Structure Accuracy described in §4.3.
- **Versions.** No commit hash for the code, no revision identifiers for the Hugging Face checkpoints and datasets.
- **arabic-books quality.** The 1.1-billion-token figure has no stated tokenizer, and the corpus has no quality estimate. Even a manual check of 20 pages would give downstream users a sense of its error rate.

4. *What specifically to fix*

1. **§5, second paragraph:** replace "94.7%" and "6.1%" with the values from Table 1, row Arabic Large Nougat (Structure Acc 0.9884, CER 0.0662), or explain where the other numbers come from.
2. Also in §5, drop "surpassing its smaller counterparts," or restrict it to CER and Structure Acc. Table 1 shows Arabic Small ahead on BLEU and both smaller models ahead on WER.
3. **Table 1:** add at least one Arabic-capable baseline (Tesseract Arabic) on the same 160 pages, and if possible, a general end-to-end parser from §2.4.
4. I would add a second test set of real scanned Arabic pages from outside Hindawi. Even 50 pages would do. This is the single change that would most raise the paper's value to practitioners.
5. **§4.1:** state the split unit. If it was by page, re-split by book and rerun Table 1.
6. Add 95% bootstrap intervals over the 160 pages to every cell of Table 1, and base the rankings in §4.2 on them.
7. **§4.1 versus Table 1:** use one metric list. Either add MSA and TER columns to Table 1 (with TER for all three models, not only two as in §4.4), or remove them from §4.1 and the abstract.

8. Explain the CER 2.8849 and WER 3.0748 for Nougat Small (Meta): whether they come from repetition, how insertions are counted, and whether any clipping was applied.
9. §3.3: describe the small model's training data, and give parameter counts and initialization for all three models in a short table next to §3.1.
10. §3.4: GPU type and count, batch size, optimizer, schedule, epochs actually run, and the early-stopping criterion.
11. State the decoding configuration behind Table 1, especially the repetition penalty value. It would also help to report Table 1 with and without it, since §6 names repetition as the main failure of the larger models.
12. §3.2 and §3.5: measure the Aranizer-to-Nougat token ratio on the test pages and report it, instead of "nearly four."
13. Give the formula for Structure Accuracy (§4.3). As written, I could not recompute it.
14. Explain the token limit for the small model: 2048 in §3.1, although its parent nougat-small handles 3584 according to §3.5.
15. Explain the reasoning for §4.5, which recommends arabic-base-nougat for general use although it has the lowest BLEU in Table 1.
16. A short qualitative figure would help: two or three pages with outputs from each model, including one hallucination by the small model and one repetition case by the large model.
17. Pin versions: a commit hash for the repository and revision IDs for the three checkpoints and two datasets.
18. Minor: the roadmap at the end of §1 says "Section 6 identifies limitations ... while Section 5 concludes." It names Section 6 before Section 5; I would list them in the order they appear.

5. Recommendation

Minor revisions. The core result is real and large, and the open release of models and data is a genuine service to Arabic document processing. What needs work is mostly reporting: the conclusion quotes numbers that are not in Table 1, the metric list changes between §4.1 and Table 1, and the training and decoding details are too thin to rerun the experiments. Adding one Arabic-capable baseline and a small out-of-domain scanned test set would turn this from a solid technical report into a paper whose claims about practical OCR quality can be trusted as stated.

Declarations

Potential competing interests: No potential competing interests to declare.