

Peer Review

Review of: "GeAR: Generation Augmented Retrieval"

Evgenii Arsentev¹

1. Independent researcher

1. What the work claims

The paper proposes GeAR, a retriever that keeps a standard bi-encoder for document scoring but adds two components borrowed from vision-language pre-training (the authors credit ALBEF, BLIP, and MoCo in §3.3 and §3.4): a fusion encoder that shares weights with the query encoder and cross-attends to the document token embeddings, and a causal text decoder that generates the query-relevant content of the document. Training is the plain sum of a contrastive loss with momentum-distilled soft labels and a language-modelling loss (Eq. 3). At inference, the bi-encoder alone retrieves, so indexing and search costs are those of a bi-encoder; the cross-attention weights of the fusion encoder are then read out to localise the most relevant sentence inside a retrieved document, and the decoder can produce an answer or a relevant sentence as a readable "explanation" (§3.5). To train the relevant-information scenario (RIR), the authors build a synthetic set from Wikipedia: sentences of 8 to 20 tokens are rewritten into keyword queries by Llama-3.1-70B-Instruct and filtered with BGE at similarity 0.5 or more (Appendix A), which yields the figure of 5.8M triples given in §3.2. The QA scenario (QAR) is trained on 30M PAQ samples (§4.1).

The empirical claims are three. GeAR is competitive in document retrieval (Table 1). It is clearly better than all baselines at sentence localisation (Table 2; §4.2 says "GeAR came out ahead on all metrics"). And generation and localisation improve together during training, which the authors read as evidence that the generation objective drives localisation (§4.3, Figure 4a/b). I think the idea is well motivated and simply built, and the paper is candid about one trade-off: §4.3 openly reports that removing the LM loss slightly improves retrieval. The data-synthesis appendix, including the prompt and the negative result about LLM-expanded documents, is useful to anyone building similar data. My concerns are about

whether the localisation comparison isolates what the paper says it isolates, several table values that cannot be correct as printed, and the train/test separation for RIR.

2. Whether the conclusion holds on the data

Document retrieval: the gain comes from the training recipe, not from generation. The most informative rows in Table 1 are the last two. "GeAR w/o L_{LM} " is at least as good as full GeAR on nine of the ten columns: SQuAD R@5 0.889 vs 0.883 and M@5 0.776 vs 0.762; NQ R@5 0.755 vs 0.747; TriviaQA R@5 0.660 vs 0.660 and M@5 0.399 vs 0.398; PAQ R@5 0.955 vs 0.940 and M@5 0.877 vs 0.855; RIR R@5 0.963 vs 0.961 and M@5 0.907 vs 0.903. The only exception is NQ M@5, 0.565 vs 0.567. The authors acknowledge this in §4.3, and I appreciate that. What follows from it, though, is not drawn out. The advantage of GeAR over the retrained baselines SBERT_{RT} and BGE_{RT} (for example, PAQ R@5 0.940 vs 0.859 and 0.901) is already present without any generation loss, so it must come from something else in the recipe: the momentum encoder, the 57,600-sized queue, the soft-label ramp to 0.4 (Appendix C), or the two unshared encoders. The sentence in §4.2, "GeAR achieves superior performance, underscoring the effectiveness of our training approach", is true only in that narrower sense. It is also not uniform. On NQ, BGE_{RT} has R@5 0.751 against GeAR's 0.747, and among the pre-trained models, GTE beats GeAR on NQ (R@5 0.767, M@5 0.587) and on TriviaQA (R@5 0.726, M@5 0.443 vs 0.660 and 0.398).

The paper does not say whether the retrained baselines received the same number of epochs, the same effective batch (48 per GPU on 16 GPUs, so 768 for QAR and 256 for RIR by my multiplication of the Table 4 values), or any comparable pool of negatives. If BGE_{RT} was trained with in-batch negatives only while GeAR had a 57,600-item queue, the comparison in the "Retrained retrieval model" block is between recipes, not between architectures.

Localisation: the comparison is asymmetric, and the key control is missing. This is the central claim of the paper, and I do not think Table 2 supports it yet. For the baselines, §4.2 says documents are split into sentences and each sentence is embedded independently, then ranked against the query. GeAR instead sees the whole document and the query jointly through cross-attention. That is the difference between a context-free sentence scorer and a cross-encoder-style reader. Some of the gap in Table 2 (NQ R@1 0.765 for GeAR vs 0.590 for E5, a difference of 0.175; PAQ R@1 0.885 vs 0.573, a difference of 0.312) may simply reflect access to context, which any cross-attention model has, whether or not it was trained to generate. Three baselines would separate these effects:

- GeAR w/o L_{LM} used as a sentence scorer. This is the direct ablation for §3.4's statement that "the introduction of LM loss is key to enhancing the information localization capability". It exists for Table 1 but not for Table 2.
- A cross-encoder or reranker applied at sentence level with the document as context, which answers the question "is generation needed, or just joint encoding?"
- A token-level late-interaction model. ColBERT is cited ([14]) and its MaxSim token scores localise naturally; landmark embedding [7] and late chunking [8] are cited in the introduction as addressing exactly this problem, but none of them appears in Table 2.

A caveat on that first bullet. In the architecture described in §3.3, the cross-attention module of the fusion encoder receives gradient only through the decoder, so without L_{LM} it is untrained. That makes the correct control a bi-encoder trained with GeAR's recipe and evaluated as a sentence scorer. It is cheap to run, since the checkpoint behind the last row of Table 1 already exists.

The retrained baselines also look like weak comparators for this task. SBERT_{RT} drops from 0.739 (pre-trained SBERT) to 0.516 on SQuAD R@1, and BGE_{RT} from 0.768 to 0.455. Models trained to match queries to whole documents are then asked to score isolated sentences, which is off their training distribution. That is not the authors' fault, but it means the "Retrained" block of Table 2 flatters GeAR.

Two values in the tables cannot be right as printed. First, Table 2, row SBERT_{RT}, RIR: R@3 = 0.899 and M@3 = 0.991. In RIR each query is written from one sentence (Appendix A), so each query has a single gold unit. With a single relevant item, MAP@k equals the reciprocal rank when the item is in the top k and zero otherwise, so it can never exceed Recall@k. A value of 0.991 against 0.899 is impossible under that definition. It is also the only cell in the RIR column where a baseline beats GeAR (GeAR has M@3 0.897), which directly contradicts "GeAR came out ahead on all metrics". Either the cell is a typo (0.891 would fit the neighbouring rows) or the metric is defined differently from what I assume; either way it needs correcting. Second, Table 3, NQ: EM 66.1 and F1 61.0. Under the standard SQuAD-style token F1, an exact match scores F1 = 1 for that example, so corpus F1 cannot fall below EM. The other three QA columns have F1 above EM (SQuAD 61.2/65.3, TriviaQA 47.4/60.0, PAQ 88.1/92.4). I suspect a swap or a different normalisation, but I cannot tell which from the text.

A related point. For the QA datasets in Table 2, M@1 is higher than R@1 in every row (e.g. GeAR SQuAD 0.810 vs 0.874). With one relevant unit, R@1 and MAP@1 would be identical, so the QA evaluation must allow several gold sentences per query. The paper never says how gold sentences are defined for SQuAD,

NQ, and TriviaQA. Footnote 1 says only that the generation target in QAR is the answer string. For TriviaQA in particular, where the answer string may occur in many sentences, the labelling rule changes the numbers a great deal.

Possible leakage in RIR. §4.1 says 95% of the synthesised data is used for training and 5% held out, and Appendix B.2 says that for RIR retrieval "the candidate documents are all documents within the entirety of the dataset". Appendix A selects three sentences per document and turns each into a query. If the split was made at the triple level, most test documents were seen in training paired with a different query, and the retrieval model has had the chance to memorise them. That would explain part of the very large RIR gap in Table 1 between in-domain training (GeAR R@5 0.961, BGE_{RT} 0.953) and pre-trained models (0.376 to 0.528). PAQ is handled carefully ("no intersection with the training data", B.2); RIR needs the same statement, at the document level.

There is also a count mismatch. Table 5 gives 5,676,877 RIR training triples. If that is 95%, the total is $5,676,877 / 0.95 = 5,975,660$, about 6.0M rather than the 5.8M stated in §3.2. The 5% test part would be about 298,783 triples, yet Table 6 lists 145,562 RIR queries and Table 7 lists 10,000. Probably these are subsamples, but the reader should not have to guess.

The QA test sets are described as "held out to observe the generalizability" (§4.1). PAQ questions were generated over Wikipedia with models trained on NQ, TriviaQA, and SQuAD, as I recall from the PAQ paper (ref. [38]), and SQuAD, NQ, and TriviaQA all draw on Wikipedia text. So "held out" should be qualified. The corpus sizes in Table 6 (20,239 documents for SQuAD, 64,501 for NQ, 104,160 for TriviaQA) are also far smaller than the full-Wikipedia settings most dense-retrieval papers report, so absolute R@5 values are not comparable to the literature. The query counts do not match the standard splits either: SQuAD v1.1 dev has 10,570 questions, and Table 6 lists 5,928. The filtering should be described.

The correlation argument is weaker than stated. Figure 4(a)/(b) plots ten points, one per epoch, with generation on one axis and localisation on the other. Both quantities rise with training, so a strong correlation across epochs is expected even if neither causes the other. No correlation coefficient is reported. The conclusion "This observation demonstrates that the generation task ... effectively enhances the model's ability" (§4.3) needs the ablation I described above, not a training curve.

Layer choice on the test set. Footnote 4 says layer 10 was used for evaluation, and Figure 4(c) shows performance by layer. If layer 10 was picked by looking at that curve on the test sets, Table 2 carries a small selection bias. A validation split for this choice would remove the doubt.

No uncertainty. All results are single runs with no variance, confidence intervals, or significance tests. With 2,889 NQ queries, a difference such as 0.751 vs 0.747 R@5 (Table 1) is about a dozen queries ($0.004 \times 2,889 \approx 11.6$). I would not draw conclusions from differences of that size either way.

"Interpretation of retrieval results". The abstract and §5 present GeAR as making retrieval more explainable. But the retrieval score comes from the bi-encoder, while the highlighted tokens and generated text come from the fusion encoder and decoder. Nothing in the paper tests whether these explanations reflect why the bi-encoder ranked a document where it did. Figures 3 and 5 are selected examples. I would soften the claim to "GeAR provides query-specific highlights alongside retrieval" unless a faithfulness test is added. Similarly, "does not add any computational burden" (abstract) holds for retrieval only; localisation and generation require a fusion-encoder pass per query-document pair, whose cost is not reported.

3. Reproducibility

Some of the reporting is good, and I want to be specific about it. Table 4 and Appendix C give the optimiser, learning-rate schedule, warmup, weight decay, epochs, per-GPU batch sizes, maximum length, momentum coefficient 0.995, queue size 57,600, soft-label weight 0.4, hardware (16 AMD MI200 for training, 8 V100 for evaluation), training time (about 5 days QAR, about 3 days RIR), and total parameter count (about 330M). The rewriting prompt is quoted verbatim in Appendix A, and the model is named ("Llama-3.1-70B-Instruct" served with vLLM). That is more than many retrieval papers provide.

What is missing:

- **Code, data, and checkpoints.** The abstract says they "will be released after completing technical review". In the Qeios version I reviewed, there is no repository link, so nothing can be run.
- **LLM generation settings.** Temperature, top-p, maximum new tokens, and any seed for the Llama rewriting are not given. Neither is the handling of malformed outputs.
- **Data sources.** No Wikipedia dump date; "high-quality documents" is not defined. The sentence splitter and the tool used to detect whether the subject is a pronoun are not named. The exact BGE checkpoint used for the 0.5 filter is not named, and neither is the procedure for sampling 30M items from PAQ.
- **Baseline identities.** "Base-level" E5, BGE, and GTE are not given with checkpoint names or versions (only SBERT has a link, footnote 2). Whether query and passage instruction prefixes were used for E5 and BGE is not stated, and this matters noticeably for those models.

- **Retrained baseline settings**, as noted in §2: epochs, batch, negatives, learning rate.
- **Localisation read-out**. How are token-level cross-attention weights turned into sentence scores? Summed or averaged over tokens, over heads, over query tokens? This is the core of the method's localisation, and it is described in one sentence (§3.5).
- **Metric definitions** for R@k and MAP@k when several gold units exist, and the rule for gold units in the QA datasets.
- **Generation decoding** for Table 3 (greedy or beam, maximum length) and the EM/F1 normalisation.
- **Index configuration**. FAISS is named, but not the index type. An approximate index would lower recall compared with exact search.
- **Seeds and run count**. Not reported anywhere.

4. What specifically to fix

1. Table 2, row SBERT_{RT}, RIR, M@3 = 0.991 exceeds R@3 = 0.899 on a task with one gold sentence per query. Check the cell. If the value is real, define the metric; if it stays, the §4.2 sentence "GeAR came out ahead on all metrics" must go.
2. Table 3, NQ: EM 66.1 is above F1 61.0, which standard token F1 does not allow. Recheck and state the normalisation.
3. Add GeAR's recipe without L_{LM} as a sentence-level scorer to Table 2. Without it, §3.4's claim that the LM loss is "key" to localisation is untested.
4. In Table 2, add at least one context-aware baseline (a sentence-level cross-encoder with the document as context) and one token-interaction baseline, preferably ColBERT, or landmark embedding [7] or late chunking [8], since the introduction positions GeAR against them.
5. §4.1 and Appendix B: state whether the RIR 95/5 split is by document or by triple, and whether RIR test documents appear in training. If splitting was by triple, rerun the RIR evaluation with a document-disjoint split.
6. Reconcile the RIR counts: §3.2 (5.8M), Table 5 (5,676,877 as 95%, implying about 5.98M), Table 6 (145,562 queries), and Table 7 (10,000).
7. §4.2, Documents Retrieval paragraph. Rephrase "GeAR achieves superior performance" to reflect that (a) the w/o L_{LM} row in Table 1 is equal to or better in nine of ten columns, (b) BGE_{RT} is ahead on NQ R@5, and (c) GTE is ahead on NQ and TriviaQA.

8. Say how SBERT_{RT} and BGE_{RT} were trained: epochs, batch, negatives. If the queue and momentum distillation are what make the difference, a retrained BGE with the same queue would show it.
9. Define gold units for SQuAD, NQ, and TriviaQA in the localisation task, and give formulas for R@k and M@k (this also explains why M@1 > R@1 in every QA column of Table 2).
10. §3.5: describe exactly how cross-attention weights are aggregated into sentence scores.
11. Footnote 4 and Figure 4(c): choose the layer on a validation split, and say so.
12. §4.3, "Correlation of Generation and Localization": report the coefficient and remove "demonstrates". An across-epoch correlation cannot show that one objective causes the other; item 3 can.
13. Report variance over at least three seeds for GeAR and the retrained baselines on one QAR and one RIR set, or bootstrap confidence intervals over queries for Tables 1 and 2.
14. Abstract and §5: limit "no computational burden" to the retrieval stage and report the per-document latency of localisation and generation. Qualify "interpretation of retrieval results", since the explanation comes from different parameters than the ranking score.
15. Appendix A: give the Llama decoding parameters, the Wikipedia dump date, the sentence splitter and pronoun detector, and the BGE checkpoint used for filtering.
16. §4.1 Baselines: list exact checkpoint names for E5, BGE, and GTE and whether instruction prefixes were used; give the FAISS index type.
17. Table 6: explain why SQuAD has 5,928 queries and NQ 2,889, and qualify "held out" given that PAQ, SQuAD, NQ, and TriviaQA all come from Wikipedia.
18. Release the code, synthetic RIR data, and checkpoints, or at least the RIR test split, with this version, not after review.
19. Small things. Table numbering follows the order of the captions, not the order of appearance (Table 3 is printed before Tables 1 and 2 on Qeios). Typos: "cross attnetion" (§4.3), "The Affect of" (§4.3 heading, should be "Effect"), "Visulization" (Figure 5), "mininum" (Appendix C), "Document retrieval serve" (§1).

5. Recommendation

Major revisions. The architecture is a sensible transfer of BLIP-style joint training to text retrieval. The retrieval results are real, and the paper is honest about the retrieval trade-off. The main claim, though, is that generation gives better fine-grained localisation, and it rests on a Table 2 that has no ablation of the

LM loss, no context-aware baseline, one impossible cell that happens to be the only baseline win, and an RIR split whose document overlap is not stated. All of these can be fixed with experiments the authors are well placed to run on existing checkpoints. If the localisation advantage survives item 3 and item 4 of my list, the paper will be considerably stronger than it is now.

Declarations

Potential competing interests: No potential competing interests to declare.