AGRAME: Any-Granularity Ranking with Multi-Vector Embeddings

Revanth Gangi Reddy[1*] **Omar Attia**[2*] **Yunyao Li**[3†] **Heng Ji**[1] **Saloni Potdar**[2]
[1]University of Illinois at Urbana-Champaign [2]Apple [3]Adobe
{revanth3,hengji}@illinois.edu
{oattia,s_potdar}@apple.com yunyaol@adobe.com

Abstract

Ranking is a fundamental problem in search, however, existing ranking algorithms usually restrict the granularity of ranking to full passages or require a specific dense index for each desired level of granularity. Such lack of flexibility in granularity negatively affects many applications that can benefit from more granular ranking, such as sentence-level ranking for open-domain QA, or proposition-level ranking for attribution. In this work, we introduce the idea of *any-granularity ranking*[1] which leverages multi-vector embeddings to rank at varying levels of granularity while maintaining encoding at a single (coarser) level of granularity. We propose a multi-granular contrastive loss for training multi-vector approaches and validate its utility with both sentences and propositions as ranking units. Finally, we demonstrate the application of proposition-level ranking to post-hoc citation addition in retrieval-augmented generation, surpassing the performance of prompt-driven citation generation.

1 Introduction

Dense Retrieval methods employ dual-encoder models to obtain vector representations for queries and passages. Usually, single-vector methods (Gautier et al., 2022; Karpukhin et al., 2020) produce one embedding per query and passage, utilizing dot product to determine relevance scores. Conversely, multi-vector methods (Khattab and Zaharia, 2020; Santhanam et al., 2022b) capture more fine-grained interactions when computing query-passage relevance score, resulting in better ranking accuracy. An advantage of multi-vector approaches is the use of token-level embeddings paired with a MaxSim operation (Khattab and Zaharia, 2020) which allows for a granular scoring mechanism through dot products between individual query and passage token embeddings. The token-level scores are then aggregated for final relevance score computation.

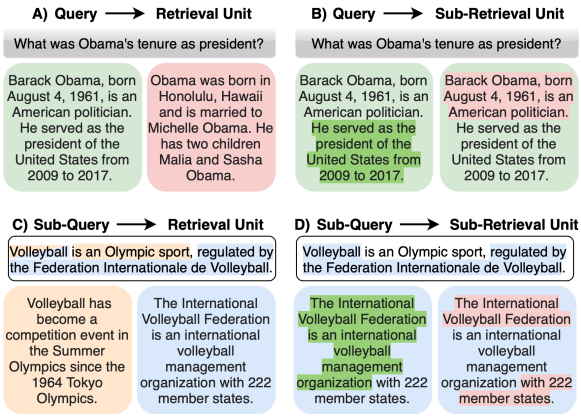

Figure 1: Ranking at different levels of granularity. $X \rightarrow Y$ is used to denote that X represents the query granularity used for ranking, with entire query encoded, and Y indicates the granularity of the retrieval unit being ranked, with entire retrieval unit encoded. In addition to the typical ranking setting (A), our proposed approach enables ranking finer retrieval units (B and D) or using finer query units for ranking (C and D).

We make an important observation that token-level embeddings in multi-vector approaches enable discriminative scoring of sub-components within a retrieval unit. We argue that finer-granularity scoring used by multi-vector approaches cannot be extended to single-vector approaches since the whole passage is represented by a single embedding, thereby not allowing for sub-unit scoring. Such granular scoring enables multi-granularity ranking, which is beneficial for many applications. In open-domain question answering (Lee et al., 2019; Karpukhin et al., 2020), ranking sentences within passages can more precisely locate answers. Similarly, in attribution (Rashkin et al., 2023; Chen et al., 2023a), atomic facts within sentences can be used as queries to retrieve evidence supporting factual claims.

To achieve this, we introduce AGRAME

[*]Equal Contribution. Revanth is an external collaborator.
[†]Work done during position at Apple.
[1]Code is available at https://github.com/apple/ml-any-granularity-ranking

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 8630–8641
November 12-16, 2024

(**A**ny-**G**ranularity **Ra**nking with **M**ulti-vector **E**mbeddings), a method that permits ranking at different levels of granularity while maintaining encoding at a single, coarser level. Our approach enables i) ranking at a finer level than the encoding (or retrieval) unit, and ii) ranking using fragments of the query, as demonstrated in Figure 1. We hypothesize that encoding at a coarser level–such as the entire retrieval unit or query–can provide additional context for the sub-retrieval units being ranked or sub-parts of the query used for ranking. In contrast, achieving such granularity with single-vector approaches requires the use of specialized encoders, such as a sub-sentence encoder (Chen et al., 2023b), or necessitates a separate encoding at the desired ranking granularity (Chen et al., 2023c).

Firstly, how well do multi-vector approaches perform when used for ranking at a finer granularity? We investigate this by conducting an experiment (§2) using ColBERTv2 (Santhanam et al., 2022b). We observe that the performance of sentence ranking is notably inferior when the encoding is at the passage-level. To improve the model's ability to rank at finer granularity, we propose a multi-granular contrastive loss during training (outlined in §3.3). Our experiments in §4.1 confirm a significant boost in sentence-level ranking while maintaining passage-level performance. While AGRAME is generally applicable to arbitrary granularity, we explore the effectiveness for proposition-level ranking, crucial for applications requiring fine-grained attribution (Rashkin et al., 2023). Our results in §4.2 indicate that incorporating a sentence-level contrastive loss further improves proposition-level ranking. Additionally, we propose PROPCITE, which utilizes propositions from generated text as queries to rank input context passages and select relevant citations. In §4.3, PROPCITE shows superior performance over traditional methods that prompt models to include citations in RAG.

The main contributions are as follows: (1) We introduce AGRAME, that leverages multi-vector embeddings for ranking at various granularities while using the same encoding-level. (2) We introduce a multi-granular contrastive loss for training multi-vector approaches, which we show improves sentence-level ranking even when encoding at passage-level. (3) We demonstrate superior proposition-level ranking using AGRAME, surpassing existing state-of-the-art methods. (4) We leverage proposition-level ranking to formulate a post-hoc citation addition approach for retrieval-augmented generation, that outperforms prompt-driven citation generation.

Model	Encoding Level	Ranking Level			
		Sentence		Passage	
		P@1	R@5	P@1	R@5
Contriever (Single Vec.)	Sentence	19.3	45.6	32.4	62.8
	Passage	-	-	37.8	65.1
ColBERTv2 (Multi Vec.)	Sentence	**31.6**	**56.3**	40.2	66.8
	Passage	27.4	48.8	**43.4**	**69.1**

Table 1: Precision@1 (P@1) and Recall@5 (R@5) results on the Natural Questions (Kwiatkowski et al., 2019) dev set. We show numbers both at sentence-level and passage-level ranking granularities for when sentences and passages are encoded individually.

2 Motivating Experiment

Here, we investigate the effectiveness of ColBERTv2 (Santhanam et al., 2022b), a multi-vector approach, in ranking at a finer granularity than the encoding level. Specifically, when encoding is at the passage-level, we measure the sentence-level and passage-level ranking performance.A MaxSim operation is applied between query token vectors and token vectors corresponding to the sentence to get a sentence-level score, which is then added to the passage-level score to get the final query-sentence relevance score for ranking. When encoding is at the sentence-level, the usual MaxSim score gives query-sentence relevance. On the other hand, the query-passage relevance score for ranking using sentence-level encoding is obtained as the maximum of the corresponding passage's query-sentence relevance scores.

We also include Contriever (Gautier et al., 2022), a single-vector approach, for comparison. When encoding is at the passage-level, Contriever does not support sentence-level ranking, which is an inherent limitation of single-vector approaches. Our evaluation uses the Natural Questions dev set (Kwiatkowski et al., 2019) and a 22M passage corpus (Gautier et al., 2022) from Wikipedia 2018 for retrieval. To keep the retrieval index size manageable, Contriever is used to index and retrieve 100 passages, which are then ranked by ColBERTv2. When ranking at sentence-level, only the sentences in these top 100 passages are considered. Eval metrics are Precision@1 and Recall@5, based on string exact match (Rajpurkar et al., 2016).

Table 1 demonstrates a significant reduction in sentence-level ranking with passage-level encod-

Q: How does climate change affect marine ecosystems

Passage: *S1:* Climate change is leading to many changes in marine life. *S2:* Coral reefs are vulnerable to the effects of climate change. *S3:* Warming waters can lead to coral bleaching, stronger hurricanes can destroy reefs, and sea level rise can cause corals to be smothered by sediment.

Sent. ID	Sentence-level Enc.	Passage-level Enc.
S1 ✗	Rank:**1**, Score:**23.31**	Rank:**1**, Score:**23.92**
S2 ✗	Rank:2, Score:17.47	Rank:2, Score:20.12
S3 ✓	Rank:3, Score:16.63	Rank:3, Score:16.96

Table 2: Sentence-level ColBERTv2 scores for different sentences in the same passage, when encoding is at sentence-level and passage-level. We see that the most relevant sentence S3 has the lowest score. Token-wise MaxSim score heatmap is also shown, with tokens in S1 & S2 having higher scores than in S3.

ing, and vice versa. We notice that the sentence-level performance decreased despite the richer contextual information provided by passage-level encoding. This contextual information is particularly useful when sentences that directly address the query lack overlapping terms (semantic or lexical), as illustrated in Table 2. Here, S1 and S2 receive higher rankings due to strong lexical ties with the query, whereas S3, which pertains to climate change effects but exhibits weak semantic overlap, scores lower, as evidenced by token-wise MaxSim score heatmap. High scores for S1 and S2 are beneficial for identifying relevant passages in a large corpus but may hinder the selection of the most pertinent sentence in those passages. This suggests that the model should be capable of adjusting relevance criteria based on the ranking task granularity. In section 3.2 & 4 we discuss our approach AGRAME and how it improves sentence-level ranking, even with a passage-level encoding.

3 Method

3.1 ColBERTv2 Preliminaries

ColBERTv2 (Santhanam et al., 2022b) is a multi-vector retrieval model, that uses token-level dense embeddings of the query and passage. Given a query q containing n tokens t^q_i and passage p containing m tokens t^p_i, additional query and passage marker tokens m_q and m_p are prepended to the query and passage respectively before encoding, to provide an additional signal to the encoder. The query-passage relevance score $S_{CB}(q,p)$ is obtained as below using the *MaxSim* operator introduced in Khattab and Zaharia (2020):

$$[\vec{Q}_{t^q_1}, \vec{Q}_{t^q_1}, ..., \vec{Q}_{t^q_n}] = E(cat(m_q, t^q_1, ..., t^q_n))$$

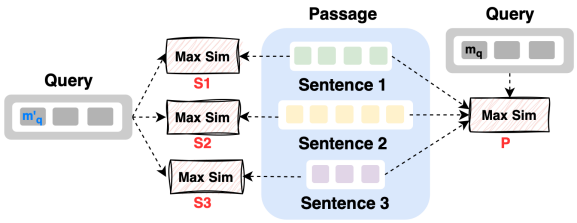

Figure 2: Figure demonstrating our sentence-level scoring methodology using multi-vector representations with encoding at passage-level. Query marker m_q is used while getting passage-level score P, while marker m'_q is used for getting sentence-level scores $S1$, $S2$, $S3$.

$$[\vec{P}_{t^p_1}, \vec{P}_{t^p_2}, ..., \vec{P}_{t^p_m}] = E(cat(m_p, t^p_1, ..., t^p_m))$$

$$S_{CB}(q,p) = \sum_{i=1}^{n} \max_{1 \le j \le m} \vec{Q}_{t^q_i}{}^T \vec{P}_{t^p_j}$$

The training process for neural retrievers typically involves a contrastive loss over the <query q, postitive p^+, negative p^-> triples. ColBERTv2 instead incorporates a distillation-based training strategy wherein k negative passages are sampled from the retrieval corpus, to form a $(k+1)$-way passage set $[p] = \{p^+, p^-_1, ..., p^-_k\}$ for each query. The relevance supervision is in the form of soft scores $S_{CE}(.)$ from a cross-encoder reranker. For training we use KL-Divergence loss $\mathcal{L}_{psg}$ between the cross-encoder and ColBERT passage scoring distributions, $D_{CE}(q,[p])$ and $D_{CB}(q,[p])$.

$$\mathcal{L}_{psg}(q,[p]) = KL(D_{CE}(q,[p]) || D_{CB}(q,[p]))$$

3.2 AGRAME: Any-Granularity Ranking with Multi-Vector Embeddings

We introduce our approach for scoring sub-units within the retrieval unit. We do this by accessing the token-level embeddings in multi-vector approaches. While AGRAME can rank at any granularity, we will consider sentences as the sub-units for simplicity. With the entire passage input to the encoder, only the output embeddings corresponding to tokens within a given sentence are used during the *MaxSim* operation for scoring that sentence.

Let $t^{p_i}_{jr}$ correspond to the j^{th} token of sentence $s^{p_i}_j$ from passage p_i that is passed as input to encoder E. To signal the model to score discriminatively *within the passage* for sentence-level ranking, we prepend a new query marker token m'_q, different from m_q used when ranking at passage-level. The *in-passage* query-sentence relevance score $S_{CB}(q, s^{p_i}_j)$ is computed as follows:

$$[\vec{Q'}_{t^q_1}, \vec{Q'}_{t^q_1}, ..., \vec{Q'}_{t^q_n}] = E(cat(m'_q, t^q_1, ..., t^q_n))$$

$$S_{CB}(q, s_j^{p_i}) = \sum_{i=1}^{n} \max_{1 \le r \le |s_j^{p_i}|} \vec{Q'_{t_i^q}}^T \vec{P_{t_{jr}^{p_i}}}$$

Note that the passage encoding is the same as before, meaning the same multi-vector index can be used for both passage-level and sentence-level ranking. As we demonstrate in §4.1, encoding at passage-level provides more context to the token embeddings to benefit sentence-level ranking.

We note that our proposed sentence-level loss (described in §3.3) teaches the model to rank sentences discriminatively *within a passage*, and not *across passages*. Hence, at inference to get a final sentence-level relevance score $Score(q, s_j^{p_i})$ to rank sentences across passages, we combine the in-passage sentence relevance score $S_{CB}(q, s_j^{p_i})$ with the usual passage-level relevance score $S_{CB}(q, p_i)$:

$$Score(q, s_j^{p_i}) = S_{CB}(q, s_j^{p_i}) + \alpha S_{CB}(q, p_i)$$

3.3 Multi-Granular Contrastive Training

As discussed in §3.1, given a query q and a passage set $[p]$, the ColBERTv2 training process aims to teach the model to identify the most relevant passage within $[p]$. To enable the model to discriminatively select sub-units within the passage, we propose to incorporate a more finer-level of training supervision, by teaching to further identify the most relevant sentence within each passage.

Since ColBERTv2 uses passage-level cross-encoder scores as teacher supervision, we train a different cross-encoder model CE' to provide in-passage sentence-level relevance supervision. Specifically, CE' takes a passage p_i as input, with a given sentence $s_j^{p_i}$ marked with delimiters \$, to give a relevance score $S_{CE'}(q, s_j^{p_i})$ for the sentence. CE' is trained using question answering data in the form <query, passage, answer> triples. A binary cross-entropy loss is used while training CE', wherein any sentence within the passage that contains the answer is marked as a positive, with the other sentences marked as negatives.

The cross encoder CE' provides soft scores for sentence-level relevance superivision when training our model. For each passage p_i, we compute a KL-divergence loss $\mathcal{L}_s(q, p_i)$ between the CE' and ColBERTv2 sentence-level scoring distributions, $D_{CE'}(q, [s^{p_i}])$ and $D_{CB}(q, [s^{p_i}])$ respectively.

$$\mathcal{L}_s(q, p_i) = KL(D_{CE'}(q, [s^{p_i}]) || D_{CB}(q, [s^{p_i}])))$$

We then aggregate each passage's sentence-level scoring loss $\mathcal{L}_s(q, p_i)$, by weighting with the corresponding passage's relevance supervision score $S_{CE}(q, p_i)$, to get a single loss $L_{sent.}(q, [p])$. The passage score weight ensures that the model is penalized higher on sentence-level losses for passages that are more relevant. The sentence-level loss $L_{sent.}(q, [p])$ is finally added to original passage-level loss $L_{psg}(q, [p])$ to get the training loss $\mathcal{L}$.

$$\mathcal{L}_{sent.}(q, [p]) = \sum_{i=1}^{k+1} \sigma(S_{CE}(q, p_i)) \mathcal{L}_s(q, p_i)$$

$$\mathcal{L}(q, [p]) = \mathcal{L}_{psg}(q, [p]) + \mathcal{L}_{sent.}(q, [p])$$

4 Experiments

AGRAME can rank at different granularities, as shown in Figure 1, which involves ranking sub-parts of the retrieval unit or ranking using sub-parts of the query. In our experiments, we aim to investigate two research questions: **RQ1:** Can the training approach proposed in §3.3 improve ranking at a finer granularity than the level of encoding, i.e. *Query→Sub-Retrieval Unit*? In §4.1, we show the improvements at sentence-level ranking from our proposed multi-granular contrastive loss, while maintaining performance at passage-level, i.e. *Query→Retrieval Unit*; **RQ2:** Can multi-vector embeddings be used to rank with sub-parts of the query? In §4.2, we demonstrate the application of multi-vector approaches in *Sub-Query→Sub-Retrieval Unit* ranking for proposition-level attribution. Here, a given proposition within a sentence is used as the query to rank and identify relevant propositions in a corpus of sentences. Further, in §4.3, we introduce PROPCITE, a post-hoc citation addition approach based on *Sub-Query→Retrieval Unit* ranking. PROPCITE scores input context passages based on propositions in the generated text to add citations in retrieval-augmented generation.

4.1 Query→Sub-Retrieval Unit Ranking for Open-Domain QA

In §2, we saw that with a multi-vector approach, sentence ranking performance drops when changing the encoding from sentence-level to passage-level. We addressed this in two ways: a) AGRAME introduces a new query marker (in §3.2) for sentence-level scoring; (b) our multi-granular contrastive loss (in §3.3) providing sentence-level relevance supervision during training. Here, we empirically demonstrate the benefits of our proposed approach for sentence-level (sub-retrieval unit) ranking performance when encoding is at passage-level.

Model	Encoding Level	Natural Questions				TriviaQA				Web Questions				Entity Questions			
		Sentence		Passage		Sentence		Passage		Sentence		Passage		Sentence		Passage	
		P@1	R@5	P@1	R@5	P@1	R@5	P@1	R@5	P@1	R@5	P@1	R@5	P@1	R@5	P@1	R@5
Contriever	Sentence	20.6	48.9	35.0	65.4	31.0	58.8	48.5	72.1	14.5	39.1	28.8	57.9	14.7	42.7	39.8	64.9
	Passage	-	-	40.3	66.0	-	-	50.1	71.5	-	-	36.9	63.6	-	-	36.9	63.6
ColBERTv2	Sentence	32.7	58.8	42.0	68.8	43.2	66.1	55.6	74.7	29.0	51.9	38.8	63.7	38.1	59.4	50.9	68.1
	Passage	27.9	51.1	43.2	**70.0**	43.5	65.6	57.5	**75.6**	27.6	50.7	41.0	65.1	39.2	55.3	53.9	69.2
Ours	Passage	**36.8**	**60.5**	**44.0**	69.9	**48.9**	**68.1**	**57.9**	**75.6**	**33.2**	**55.6**	**41.2**	**65.4**	**43.8**	**61.5**	**54.2**	**69.5**

Table 3: Precision@1 (P@1) and Recall@5 (R@5) results on various open-domain QA datasets. We show numbers both at sentence-level and passage-level ranking for when sentences and passages are encoded individually.

Model	Encoding Level	Finance		Recreation		Lifestyle		Science		Technology		Writing		Biomedical		Average	
		Sent.	Psg.	Sent.	Psg.	Sent.	Psg.	Sent.	Psg.	Sent.	Psg.	Sent.	Psg.	Sent.	Psg.	Sent.	Psg.
Contriever	Sentence	13.8	22.2	17.9	29.4	19.7	32.7	10.9	18.8	11.3	18.3	23.0	36.1	10.7	16.6	15.3	24.9
	Passage	-	27.2	-	34.7	-	40.4	-	17.5	-	21.4	-	39.6	-	4.6	-	26.5
ColBERTv2	Sentence	15.8	23.7	24.0	33.6	22.6	34.2	17.6	25.0	15.5	23.4	33.4	46.6	12.8	17.3	20.2	29.1
	Passage	17.1	**29.8**	25.5	**40.7**	23.9	41.9	18.4	**28.7**	16.7	**27.1**	34.7	**51.3**	13.1	16.9	21.4	**33.8**
Ours	Passage	**19.5**	**29.8**	**29.2**	40.4	**30.0**	**42.6**	**20.5**	28.1	**18.4**	26.4	**36.7**	50.2	**15.4**	**17.5**	**24.2**	33.6

Table 4: Precision@1 results on various domains from the RobustQA dataset (Han et al., 2023). We show numbers at sentence-level and passage-level ranking when sentences and passages are encoded individually.

4.1.1 Setup

Datasets We first evaluate on different popular open-domain QA datasets: Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), Web Questions (Berant et al., 2013) and Entity Questions (Sciavolino et al., 2021). For the retrieval corpus, we use the 2018 Wikipedia dump released by Lee et al. (2019). For cross-domain evaluation, we consider the RobustQA (Han et al., 2023) dataset, a large-scale OpenQA benchmark specifically designed for evaluating cross-domain generalization capabilities.

Baselines We use Contriever (Gautier et al., 2022) as the single-vector baseline, and ColBERTv2 (Santhanam et al., 2022b) as the multi-vector baseline. All models use MS MARCO (Nguyen et al., 2016) as the training dataset. Due to storage constraints, we create a single-vector index with Contriever and rank the top-100 retrieval results from Contriever using the multi-vector approaches to report numbers.

4.1.2 Results

Table 3 shows ranking results on various open-domain QA datasets. It is evident that for both Contriever and ColBERTv2, passage-level ranking is best with passage-level encoding. Our proposed approach not only improves sentence-level ranking with passage-level encoding but also surpasses its performance at sentence-level encoding. This result supports our notion that passage-level encoding aids sentence-level ranking by providing additional context. Furthermore, our method ensures that passage-level ranking remains on par with that of ColBERTv2. Table 4 shows sentence-level and passage-level ranking results for the cross-domain RobustQA benchmark. We observe that our approach is robust and extends to cross-domain settings, with consistent improvements in sentence-level ranking across the board, while maintaining passage-level ranking performance.

4.1.3 Analysis

We do an ablation study to examine the impact of substituting the default query marker m_q with a new query marker m'_q on sentence-level scoring. Note that the markers m_q and m'_q at inference only affect the query token embeddings. his analysis involved three experimental settings: A1) employing m'_q for both training and inference, representing our proposed method; A2) using m'_q during training with m_q at inference; and A3) utilizing m_q throughout training and inference. Additionally, we compared these settings against the baseline ColBERTv2, which lacks sentence-level training supervision and uses m_q at inference. Based on the results in Table 5, A1 outperforms A3 in the majority of the cases, indicating the effectiveness of the new marker. Furthermore, the A2 setting, wherein m_q is used at inference, demonstrates improvements over the ColBERTv2 baseline, suggesting that training with m'_q enables the model to encode passage tokens to be better at discriminatively scoring sentences. More analysis in Appendix A.1.

Setting	NQ	TQA	WebQ	EntQ
ColBERTv2	27.9	43.5	27.6	39.2
A1) Train→m'_q, Rank→m'_q	**36.8**	**48.9**	**33.2**	43.8
A2) Train→m'_q, Rank→m_q	29.1	44.8	29.4	40.8
A3) Train→m_q, Rank→m_q	35.9	47.6	32.9	**44.1**

Table 5: Precision@1 of sentence-level ranking performance for different variations of using a query marker. ColBERTv2, trained with a passage-level loss, uses marker m_q. The latter three variants are represented with the query marker while training with sentence-level loss and that used for ranking at inference.

Model	Proposition		Sentence	
	P@1	R@5	P@1	R@5
GTR	21.9	52.5	49.4	77.0
ST5	26.2	57.7	50.6	79.4
SUBENCODER (GTR)	40.8	72.9	42.9	82.3
SUBENCODER (ST5)	41.0	72.2	43.5	81.4
ColBERTv2	46.9	74.2	54.7	**87.8**
Ours	**47.7**	**74.7**	**55.0**	87.4

Table 6: Evaluation results on the *Atomic Fact Retrieval* task in PROPSEGMENT (Chen et al., 2023a). The encoding level is individual sentences, with each sentence consisting of multiple propositions. All models are based on encoders with 110M parameters. Numbers for GTR, ST5, SUBENCODER are from Chen et al. (2023b).

4.2 Sub-Query→Sub-Retrieval Unit Ranking for Fine-Grained Attribution

Attributing model-generated text with citations from established sources is an emerging research topic (Gao et al., 2023a; Liu et al., 2023). Each sentence in the generation can have multiple atomic facts (or propositions) (Min et al., 2023) for which evidence needs to be obtained. We explore ranking at proposition-level, wherein given a sentence, fine-grained attributions (Rashkin et al., 2023) need to be obtained for a specific sub-part of the sentence. Our study focuses on the *Atomic Fact Retrieval* task, which requires identifying and sourcing evidence for specific propositions within a sentence.

We consider this task to demonstrate that multi-vector embeddings can be leveraged to natively rank at the sub-sentence level, and compare against specialized models (Chen et al., 2023b) trained to encode propositions. We note that the encoding here is at the sentence-level, unlike §4.1 where encoding is at the passage-level. Since the marker m'_q in our multi-granular training loss is for sentence-level ranking with passage-level encoding, we use the default marker m_q when ranking at proposition-level with sentence-level encoding.

4.2.1 Setup

Dataset For evaluating proposition-level ranking, we use the PROPSEGMENT (Chen et al., 2023a), which involves 8.8k propositions as sub-queries for which evidence needs to be obtained from a corpus of 45k human-labeled atomic propositions.

Baselines We consider SUBENCODER (Chen et al., 2023b) as the primary baseline, a state-of-the-art sub-sentence encoder for proposition-level ranking. SUBENCODER has been specifically trained to produce contextual embeddings for atomic propositions in a sentence. SUBENCODER produces a single sub-sentence embedding for each atomic proposition in the sentence. We include other sentence-level embedding approaches, such as GTR (Ni et al., 2022b), Sentence-T5 (Ni et al., 2022a) as baselines from Chen et al. (2023a).

4.2.2 Results

Table 6 shows results from the Atomic Fact Retrieval task. The baseline ColBERTv2 already outperforms the state-of-the-art SUBENCODER at proposition-level (sub-sentence) ranking. Although our proposed approach adds a sentence-level constrastive loss at passage-level encoding, we see some improvements even when ranking at proposition-level. However, we hypothesize that better proposition-level ranking can be expected by further training with a proposition-level loss in §3.3, which we leave for future work to explore. Given the superior performance of multi-vector methods in proposition-level ranking, we introduce next in §4.3 a practical application that leverages this capability to add citations to machine-generated text.

4.3 Sub-Query→Retrieval Unit Ranking for Citation Addition

Retrieval-augmented generation (RAG) (Lewis et al., 2020) produces a long-form answer to a query using a set of relevant input passages. We investigate the use of multi-vector methods for citation addition in RAG. Specifically, given K passages and the generated long-form answer, the task is to add citations to one or more of the input passages for each sentence of the generated response.

We present PROPCITE, a post-hoc methodology that adds citations to the input context supporting propositions (atomic facts) in the generated text. Figure 3 illustrates PROPCITE, which makes use

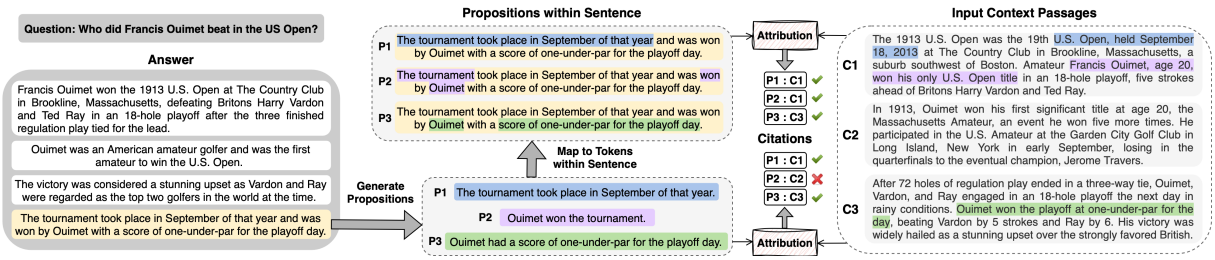

Figure 3: PROPCITE, our proposed approach for post-hoc addition of citations to long-form answers. PROPCITE encodes sentences and uses the propositions *within* them as queries for attribution. Propositions are highlighted within the current sentence (in yellow), and corresponding supporting evidence is highlighted in the input passages. PROPCITE correctly attributes *P2* to *C1*, while directly encoding and querying using *P2* incorrectly attributes to *C2*.

of propositions tagged[2] *within* the generated sentences. These sentence sub-units are used to score the input passages and identify the ones to cite. Our approach is 'post-hoc' with citations added *after* the text is generated, unlike the typical approach of generating text with citations by directly prompting the generation model (Gao et al., 2023c).

4.3.1 Setup

Datasets and Metrics We consider two long-form QA datasets: ASQA (Stelmakh et al., 2022) and ELI5 (Fan et al., 2019). Our RAG setup uses either K=5 or K=10 passages as input to the language model to generate answers. Attribution quality is evaluated using citation precision and recall metrics from Gao et al. (2023c). Citation recall assesses if the output is completely supported by cited passages, while citation precision identifies irrelevant citations. These metrics are calculated with TRUE (Honovich et al., 2022), an 11B-parameter model trained on a collection of natural language inference datasets, and widely used (Bohnet et al., 2022; Gao et al., 2023b) for evaluating whether cited passages entail the claims in the sentence.

Baselines We compare PROPCITE against the commonly used instruction-driven citation generation (Gao et al., 2023c), which we call *Generate*, where the generation model is prompted to output text with citations. We use the same few-shot prompt as in Gao et al. (2023c) to instruct the model to add citations *while* generating the answer. We consider instruction-tuned variants of two LLMs: 4B Qwen1.5 (Bai et al., 2023) and 7B Mistral (Jiang et al., 2023). Additionally, we evaluate against Self-RAG (Asai et al., 2023), which employs a self-reflective generation framework to adaptively pick passages to generate from and cite.

[2]More details on identifying propositions in Appendix B

Generation Model	Psg.	Citation Method	ASQA P	ASQA R	ELI5 P	ELI5 R
Qwen1.5 4B	5	*Generate*	26.9	21.3	11.0	8.6
		PROPCITE	**48.9**	**54.5**	**19.5**	**23.4**
	10	*Generate*	14.8	11.7	5.7	4.7
		PROPCITE	**45.3**	**52.0**	**18.3**	**22.9**
Mistral 7B	5	*Generate*	64.9	69.5	40.5	49.0
		PROPCITE	**65.7**	**74.2**	**43.0**	**51.9**
	10	*Generate*	60.2	66.7	38.0	48.8
		PROPCITE	**61.6**	**71.9**	**41.9**	**53.0**
Self-RAG 7B	5	*Generate*	67.9	67.1	-	-
		PROPCITE	**68.5**	**68.4**	-	-
Self-RAG 13B		*Generate*	71.4	70.5	-	-
		PROPCITE	**71.6**	**71.5**	-	-

Table 7: Table showing precision (P) and recall (R) for different citation addition approaches on the ASQA and ELI5 datasets. For Self-RAG, we directly use the generation outputs from Asai et al. (2023).

4.3.2 Results

Table 7 compares citation precision and recall between the *Generate* approach vs our post-hoc PROPCITE. Text generation models with strong instruction-following capabilities, such as Mistral 7B, outperform weaker models like Qwen1.5 4B in generating text with citations. Additionally, the quality of post-hoc citations is contingent upon the quality of the generated text; weaker models often lead to lower-quality citations due to inaccuracies or hallucinated text. PROPCITE enhances citation quality across both 4B and 7B model outputs, and shows improvements even for Self-RAG models, which are specially tuned for citation generation by incorporating reflection tokens. Importantly, as a post-hoc solution, PROPCITE can be integrated with any RAG framework without modifications to the generation model. PROPCITE is lightweight and can add citations to sentences as they are generated one-by-one in a streaming setting.

Setting	Precision	Recall
Generate	64.9	69.5
PROPCITE	65.7	**74.2**
+ *Thresholding*	**69.2**	71.1
(i) Propositions as query	63.5	73.9
(ii) Sentence as query (top 1)	69.0	67.5
(iii) Sentence as query (top 2)	51.2	72.6

Table 8: Analysis of citation precision and recall performance on ASQA for Mistral 7B when using top-5 passages as input. We consider different settings, wherein the generated propositions or the sentence itself are used as the query when searching for relevant citations.

4.3.3 Analysis

Table 8 shows results for an ablation study examining different methods of post-hoc citation addition, highlighting the effectiveness of using propositions *within* sentences as queries. We introduce a high-precision variant of PROPCITE that employs thresholding to mitigate false positives, only adding citations if the top-scored passage exceeds a relevance score margin of at least 1.0. We also compare against variants that directly encode the proposition (i) or query using the entire sentence (ii, iii). Our results show that encoding propositions directly leads to lower precision compared to encoding entire sentences, supporting our hypothesis that sentence-level encoding provides a richer context for proposition-level queries. Figure 3 illustrates this with an example from ASQA. Here, PROPCITE correctly attributes *P2* to *C1*. However, directly encoding *P2* incorrectly links it to passage *C2*, referencing a different tournament won by Ouiment.

Additionally, we evaluate an alternative method using the entire sentence as a single query, which yields high precision but low recall when tagging only the top-scored passage as the citation (ii), and higher recall but reduced precision when using the top two scored passages (iii). Overall, PROPCITE results in 66% of sentences with one citation, 30% with two, and 4% with more than two citations.

5 Related Work

The phrase 'multi-granularity' can have different meanings depending on the domain of usage. In the field of image retrieval, it corresponds to representing different regions of the image separately (Wang et al.; Zhang et al., 2022). For representation learning (Kusupati et al., 2022; Li et al., 2024), it refers to encoding information at different output embedding dimensions, to adapt to the computational constraints of downstream tasks. Our definition of granularity in text ranking corresponds to the level of the ranked sub-units with a given retrieval unit.

Multi-vector approaches (Luan et al., 2021; Khattab and Zaharia, 2020; Santhanam et al., 2022b) have primarily been used for ranking at the same granularity as the encoding level, which is typically at passage-level. Single vector approaches, on the other hand, inherently do not support ranking at a finer granularity than the encoding level, thereby needing a separate dense index for each granularity (Chen et al., 2023c). Hence, specialized models for single-vector embeddings have been introduced for embedding phrases (Lee et al., 2021), propositions (Chen et al., 2023b), sentences (Reimers and Gurevych, 2019) or passages (Karpukhin et al., 2020). Our approach overcomes this limitation by leveraging multi-vector approaches for ranking at different granularities, while still encoding at a single coarser granularity.

Prior approaches that rank at different granularities have used custom scoring functions or incorporate separate embeddings. Chang et al. (2023) proposes a multi-granularity matching model that uses a convolutional filter for scoring, instead of cosine similarity, meaning it cannot be scaled to a retrieval-scale corpus due to the matching function. Hierarchical ranking approaches (Liu et al., 2019; Chu et al., 2022; Ma et al., 2024) consider multi-granular ranking but require using a separate embeddings for each ranking granularity. Further, our approach uses multi-vector embeddings with a dot product for scoring at all levels of granularity, meaning the same pre-computed dense index can be used for ranking at any granularity.

6 Conclusion

In this work, we introduce AGRAME, which leverages multi-vector embeddings to rank at finer granularities, while maintaining encoding at a single, coarser level. Our proposed multi-granular contrastive loss for training multi-vector approaches improves sentence ranking performance even with encoding at passage-level. We demonstrate that AGRAME can rank at any granularity, even by using sub-parts of the query for ranking. Leveraging multi-vector approaches' superior performance at proposition-level ranking, our post-hoc attribution approach, PROPCITE, outperforms the conventional approach of prompt-driven citation in retrieval-augmented generation.

Limitations

While training AGRAME, we incorporate sentence-level relevance supervision in addition to the usual passage-level supervision, and introduce a corresponding new query marker m'_q for sentence-level ranking granularity. While this does improve ranking even at proposition-level ranking, as shown in §4.2, we expect more improvements from additionally providing proposition-level supervision during training, along with a separate query marker for proposition-level ranking granularity.

While our proposed PROPCITE approach is lightweight, we do not explicitly measure latency when used for post-hoc citation addition in a practical application such as streaming text generation. Moreover, since the citation precision and recall metrics are automatic, there is a possibility of inaccurate judgement from the evaluation model, although it has been shown in Gao et al. (2023c) to highly correlate with human judgements.

Acknowledgement

We would like to thank Omar Khattab and members of the Blender NLP group for helpful comments and feedback. We are also grateful to members of the Apple Knowledge Platform team, especially Mostafa Arefiyan, Ihab Ilyas, Theodoros Rekatsinas and Benjamin Han for early discussions. This research is based upon work supported DARPA ITM Program No. FA8650-23-C-7316 and the Agriculture and Food Research Initiative (AFRI) grant no. 2020-67021-32799/project accession no.1024178 from the USDA National Institute of Food and Agriculture. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Bernd Bohnet, Vinh Q Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, et al. 2022. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*.

Guanghui Chang, Weihan Wang, and Shiyang Hu. 2023. Matchacnn: A multi-granularity deep matching model. *Neural Processing Letters*, 55(4):4419–4438.

Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, Dan Roth, and Tal Schuster. 2023a. Propsegment: A large-scale corpus for proposition-level segmentation and entailment recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8874–8893.

Sihao Chen, Hongming Zhang, Tong Chen, Ben Zhou, Wenhao Yu, Dian Yu, Baolin Peng, Hongwei Wang, Dan Roth, and Dong Yu. 2023b. Sub-sentence encoder: Contrastive learning of propositional semantic representations. *arXiv preprint arXiv:2311.04335*.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Dong Yu, and Hongming Zhang. 2023c. Dense x retrieval: What retrieval granularity should we use? *arXiv preprint arXiv:2312.06648*.

Xiaokai Chu, Jiashu Zhao, Lixin Zou, and Dawei Yin. 2022. H-ernie: A multi-granularity pre-trained language model for web search. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*, pages 1478–1489.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023a. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan,

et al. 2023b. Rarr: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023c. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488.

Izacard Gautier, Caron Mathilde, Hosseini Lucas, Riedel Sebastian, Bojanowski Piotr, Joulin Armand, and Grave Edouard. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603.

Rujun Han, Peng Qi, Yuhao Zhang, Lan Liu, Juliette Burger, William Yang Wang, Zhiheng Huang, Bing Xiang, and Dan Roth. 2023. Robustqa: Benchmarking the robustness of domain adaptation for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4294–4311.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. 2022. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning dense representations of phrases at scale. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6634–6647.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Xianming Li, Zongxi Li, Jing Li, Haoran Xie, and Qing Li. 2024. 2d matryoshka sentence embeddings. *arXiv preprint arXiv:2402.14776*.

Nelson F Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7001–7025.

Wei Liu, Lei Zhang, Longxuan Ma, Pengfei Wang, and Feng Zhang. 2019. Hierarchical multi-dimensional attention model for answer selection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Kai Ma, Junyuan Deng, Miao Tian, Liufeng Tao, Junjie Liu, Zhong Xie, Hua Huang, and Qinjun Qiu. 2024. Multi-granularity retrieval of mineral resource geological reports based on multi-feature association. *Ore Geology Reviews*, page 105889.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPs*.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022a. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, et al. 2022b. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2023. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022a. Plaid: an efficient engine for late interaction retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1747–1756.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022b. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 6138–6148. Association for Computational Linguistics (ACL).

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288.

Chengji Wang, Zhiming Luo, Yaojin Lin, and Shaozi Li. Text-based person search via multi-granularity embedding learning.

Miriam Wanner, Seth Ebner, Zhengping Jiang, Mark Dredze, and Benjamin Van Durme. 2024. A closer look at claim decomposition. *arXiv preprint arXiv:2403.11903*.

Jiacheng Zhang, Xiangru Lin, Minyue Jiang, Yue Yu, Chenting Gong, Wei Zhang, Xiao Tan, Yingying Li, Errui Ding, and Guanbin Li. 2022. A multi-granularity retrieval system for natural language-based vehicle retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3216–3225.

A Analysis

A.1 Query→Sub-Retrieval Unit Ranking for Open-Domain QA

We show the training loss curves in Figure 4 when the same query marker (m_q) vs different query markers (m'_q and m_q) are used for sentence-level and passage-level loss respectively. We can see that the model converges faster at sentence-level loss when new marker m'_q is used. Further, Table 9 shows the sentence-wise scores for the example in Table 2 from using m'_q vs m_q for sentence-level scoring. We observe that sentence-level ranking changes when m'_q is used, with the most relevant sentence (S3) ranked best.

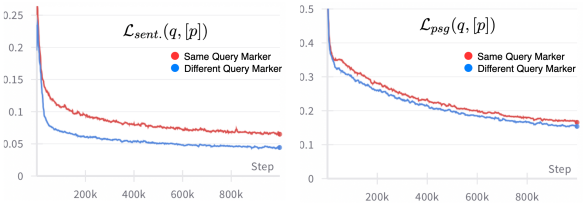

Figure 4: Comparison of training curves for sentence-level and passage-level loss, when a different query marker is used. The model converges faster at sentence-level with a different query marker, while passage-level loss is mostly similar for the two.

Sent. ID	Query Marker m'_q	Query Marker m_q
S1 ✗	Rank:2, Score:14.32	Rank:**1**, Score:**24.04**
S2 ✗	Rank:3, Score:14.16	Rank:2, Score:21.07
S3 ✓	Rank:**1**, Score:**15.92**	Rank:3, Score:16.81

Table 9: Sentence-level scores from our model at passage-level encoding for the example in Table 2, when different query markers are used. The most relevant sentence (S3) is ranked best when new marker m'_q is used.

A.2 Time and Storage Consumption Analysis

AGRaME uses the same token embeddings from a single coarser level of encoding for ranking at finer granularities. Hence, the storage consumptions are the same as the baseline ColBERTv2 approach. AGRaME uses different query embeddings depending on the ranking granularity, i.e. with marker m_q for passage ranking and m'_q for sentence ranking. Thus, compared to the baseline ColBERTv2, AGRaME involves an extra query embedding step that involves using a different query marker m'_q and then max-sim scoring with passage tokens. As per PLAID (Santhanam et al., 2022a), an efficient inference engine for ColBERTv2, the query embedding step contributes to 20% (roughly 12ms) of the overall retrieval, with majority of the latency corresponding to the passage retrieval steps, while the final max-sim scoring step has negligible latency. We hypothesize that this additional query embedding can be run concurrently with the passage retrieval process, thereby seeing no increase in the overall latency.

B Identifying Propositions in Sentences

We employ the approach from Chen et al. (2023b), which uses a T5 model (Raffel et al., 2020) to segment sentences into propositions, that are then converted into token masks by aligning the tokens in each proposition to the sentence. While we use a T5 model to explicitly generate propositions, faster approaches relying on syntactic dependency parsing (Goyal and Durrett, 2020; Wanner et al., 2024) can be a cheaper alternative to get the substructures with a sentence that represents the propositions or atomic claims.

C ColBERTv2 Training

Our training data to finetune ColBERTv2 for AGRAME comprises the 20M examples from Santhanam et al. (2022b). Hyperparameters for the training run are provided below.

Hyperparameter	Value
Max Steps	1,000,000
Warmup	40,000
Batch Size	16
Learning Rate	1e-05
Output Embed Dim	128
N-Way Negatives	63
In-Batch Negatives	True
Max Doc Len	128
Max Query Len	32

Table 10: Hyperparameters for training ColBERTv2