JCTA Journal of Computing Theories and Applications ISSN:3024-9104

Research Article

Benchmark-Defined Adaptive Candidate Reranking for Indonesian Multi-Hop Question Answering

Iqbal Abdul Rahman * and Antoni Wibowo

Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia; e-mail: iqbal.rahman001@binus.ac.id; antoni.wibowo@binus.edu
* Corresponding Author : Iqbal Abdul Rahman

Abstract: Multi-hop question answering (MHQA) requires recovering a complete set of supporting documents, yet most retrieval-augmented pipelines apply one evidence budget and one selection strategy to every question. This paper studies benchmark-defined adaptive candidate reranking for Indonesian MHQA in a closed setting in which each question is paired with ten candidate documents. A routing classifier assigns each question to a two-gold-document branch (HotpotQA), processed by two-stage sequential reranking, or to a four-gold-document branch (a bridge-comparison subset of 2WikiMultihopQA), processed by parallel ranking-based reranking, using lightweight XLM-RoBERTa cross-encoders. All components are trained without stemming and selected on a held-out validation split. On 8,000 Indonesian-translated evaluation questions, the framework recovers the complete supporting-document set for 95.40% of questions with 3.00 documents per question and 97.69% document precision. In evidence recovery, it significantly outperforms a single-model fixed-K=4 reranker (+2.60 points; McNemar's exact test $p < 10^{-17}$); relative to top-4 selection with the same branch-specific scorers, it trades 0.89 points of containment for 23.55 points higher document precision and one document fewer per question. Ablations attribute the largest effect to sequential conditioning (+14.00 points at an equal budget of two documents) and show that routing errors cost 0.03 points within the benchmark, that transfer initialization mainly accelerates convergence, and that Sastrawi stemming degrades the scorers. Class-wise probes and a cross-source test show that the routing classifier relies on a film-topic shortcut and misroutes 10.78% of unseen two-document questions; it is therefore a benchmark-defined controller, not a detector of reasoning complexity, and the results do not establish generalization beyond the evaluated benchmark. With GPT-5-mini as the generator, adaptive evidence uses 68.5% fewer context tokens than all ten candidates and 26.8% fewer than fixed-K=4 selection at 9 to 20 ms of additional selection latency per question, but no improvement in answer accuracy was found: answer accuracy remains statistically indistinguishable across evidence contexts and within the run-to-run variance of the generator.

Received: September, 16th 2026
Revised: October, 20th 2026
Accepted: October, 21st 2026
Published: November, 1st 2026

Keywords: Adaptive Candidate Reranking; Benchmark-Defined Routing; Closed Candidate Reranking; Cross-Encoder Reranking; Indonesian Natural Language Processing; Multi-Hop Question Answering; Retrieval-Augmented Generation; Supporting-Document Recovery.

Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) licenses (https://creativecommons.org/licenses/by/4.0/)

1. Introduction

Multi-hop question answering (MHQA) requires a system to gather and integrate evidence from multiple documents before producing an answer, in contrast to single-hop QA, in which a single passage is sufficient [1], [2]. Retrieval-Augmented Generation (RAG) has become a standard paradigm for grounding such answers in retrieved evidence [3], combining a retrieval or reranking stage with a neural generator. However, most RAG pipelines still apply a uniform candidate budget or selection mechanism to every question, regardless of how many supporting documents that question actually requires. This design is suboptimal: questions with short reasoning chains may be served well by a compact, focused selection, while questions with broader evidence requirements need a different selection strategy altogether. Selecting too few candidates risks omitting a required supporting document, while

selecting too many introduces distractor noise and longer contexts into the downstream generator [4]–[6].

This problem is compounded for Indonesian, a language that remains underrepresented in retrieval-oriented question answering research despite its large number of speakers, and for which no natively authored multi-hop benchmark with supporting-document annotations is available; this study therefore relies on machine-translated versions of established English benchmarks [7].

Existing work addresses evidence selection from three directions, each with a limitation that motivates this study. First, retrieve-then-rerank pipelines score candidates with neural rerankers such as monoBERT [8] and pass a fixed top-K to the generator [3], so a question needing two documents receives the same budget as one needing four. Second, adaptive and iterative RAG methods decide whether, when, or how often to retrieve, by routing on query complexity [9], interleaving retrieval with chain-of-thought reasoning [10], or triggering retrieval from generation-time signals [11]–[13]; they operate over open corpora with repeated LLM calls, and each step still returns a fixed top-K. Third, supporting-document selectors for the HotpotQA distractor setting, such as SAE [14], FE2H [15], and Beam Retrieval [16], are highly accurate in English but apply one selection procedure to every question, and recent dynamic-budget rerankers [17]–[19] adapt the number or set of passages with a single LLM-based mechanism. The gap addressed here is therefore specific: whether a lightweight encoder-based reranker can adapt both the evidence budget and the selection structure, sequential for chained evidence and parallel for multi-chain evidence, within a closed pool, and what each component contributes once isolated.

We study this problem under a closed candidate reranking setting, in which each question is paired with a fixed pool of candidate documents and the task is supporting-document selection rather than open-domain retrieval (Section 3.1). Within this setting, we distinguish two benchmark-defined branches, a two-gold-document branch drawn from HotpotQA and a four-gold-document branch drawn from a filtered bridge-comparison subset of 2WikiMultihopQA, and route each question to a branch-specific reranking strategy using a learned routing classifier.

We deliberately avoid framing this routing mechanism as a general-purpose reasoning complexity detector. The number of gold documents a benchmark assigns to a question, two versus four, is a structural property of how that benchmark was constructed rather than an independently validated measure of how difficult the underlying reasoning is. To test this, Section 4.1 reports two diagnostics: class-wise probes based on shallow lexical and stylistic classifiers with no access to any learned semantic representation, and a cross-source test on unseen two-document 2WikiMultihopQA questions that separates question source from evidence cardinality. We report these diagnostics in the main text, rather than only in a limitations section, because they bear directly on how the routing accuracy should be interpreted.

The contributions of this paper are as follows:

- A benchmark-defined adaptive candidate reranking framework that adapts both the evidence budget and the selection structure, two-stage sequential reranking for chained two-document evidence and parallel ranking-based reranking for multi-chain four-document evidence, using lightweight encoder-based cross-encoders within a closed candidate pool;
- A controlled evaluation protocol in which every checkpoint and configuration is selected on a held-out validation split, and two families of fixed-K baselines separate the effects of adapting the budget and selection structure from that of branch-specific scoring, with evidence recovery, selection precision, ranking quality, context length, selection latency, and answer quality reported as distinct dimensions;
- Component-level ablations that isolate routing errors, two-stage sequential reranking at an equal evidence budget, transfer initialization separately from learning rate and loss function, Indonesian stemming, and the evidence context supplied to a fixed generator, including the run-to-run variance of the generator itself; and
- A diagnostic analysis of the routing classifier, combining class-wise shallow-feature probes with a cross-source test, that examines whether the routing classifier acts as a benchmark-defined controller relying on a topic shortcut rather than on evidence structure, together with a public release of the code, configurations, and data-preparation scripts.

The remainder of this paper is organized as follows. Section 2 reviews related work on multi-hop question answering, adaptive and corrective retrieval-augmented generation, passage reranking and supporting-document selection, dynamic evidence budgets, and Indonesian language processing. Section 3 explains the design rationale and presents the proposed method, including the dataset and preprocessing, the routing classifier, the two branch-specific reranking strategies, the answer generation module, the evaluation protocol, and the training configuration and model-selection procedure. Section 4 reports the routing diagnostics, supporting-document recovery and baseline comparisons, ablation studies, efficiency, and answer generation with error analysis. Section 5 compares these results quantitatively with previously published supporting-document selectors and adaptive RAG methods, and Section 6 summarizes the conclusions, limitations, and directions for future work.

2. Related Work

2.1. Multi-Hop Question Answering and Retrieval-Augmented Generation

Multi-hop question answering benchmarks such as HotpotQA [1] and 2WikiMultihopQA [2] explicitly assess a system's ability to integrate evidence from multiple documents rather than a single passage, and they structure this requirement differently: HotpotQA is built primarily around two-document bridge and comparison questions, while 2WikiMultihopQA includes questions requiring longer reasoning chains, including four-document bridge-comparison instances. Reading Wikipedia to answer open-domain questions established an early large-scale retriever-reader pipeline for open-domain QA [20], later extended to multi-hop dense retrieval over open-domain corpora [21], and Retrieval-Augmented Generation subsequently formalized the combination of a retrieval component with a neural generator for knowledge-intensive NLP tasks more broadly [3]. In multi-hop QA specifically, answer quality depends strongly on whether the system recovers the complete supporting evidence chain before generation, since a generator conditioned on an incomplete evidence set cannot be expected to reconstruct the missing reasoning step from the query alone. This sensitivity to incomplete or conflicting input signals is not unique to multi-hop QA; comparable robustness challenges arise wherever a model must fuse evidence from multiple, potentially unreliable sources, as in multimodal sentiment analysis under heterogeneous observability [22]. Multi-hop benchmarks are also known to contain artifacts that allow models to answer questions, or to recognize their type, from surface cues without the intended multi-hop reasoning [23], a concern that applies directly to the routing label studied here.

2.2. Adaptive and Corrective Retrieval-Augmented Generation

Recent studies have moved beyond static retrieve-then-generate pipelines by making some part of the pipeline conditional on the query. Adaptive-RAG selects among no-retrieval, single-step, and multi-step retrieval strategies based on an estimate of query complexity [9]. Self-RAG trains a language model to retrieve passages on demand and to critique both retrieved passages and its own generated output using learned reflection tokens [11]. Corrective RAG introduces a retrieval evaluator that estimates the quality of retrieved documents and triggers corrective actions, such as web search or knowledge refinement, when retrieval quality is judged insufficient [24]. Iterative methods such as IRCoT interleave retrieval with chain-of-thought reasoning so that each reasoning step determines the next query [10], while FLARE [12] and DRAGIN [13] trigger retrieval during generation from the model's uncertainty or real-time information needs. Table 1 summarizes how these methods relate to the present study. Each of these methods primarily adapts whether or how much retrieval occurs; by contrast, the present work assumes retrieval is always required, because every question is paired with a fixed candidate pool, and instead adapts the structure of the reranking process itself to the benchmark-defined evidence requirement, a distinction that matters specifically in the closed-candidate setting studied here. Separately, work on context length has shown that supplying a generator with more retrieved context does not monotonically improve answer quality, and can hurt it even when the correct evidence is present among the retrieved set [4], a pattern echoed by studies showing that noisy or irrelevant passages within a retrieved set can actively degrade RAG performance [5], which together motivate recovering a precise rather than a merely large evidence set.

Table 1. Comparison with adaptive, iterative, and dynamic-budget RAG methods

Method	Main adaptation	Difference from this work
Adaptive-RAG [9]	No, single-step, or multi-step retrieval chosen from predicted query complexity	Open corpus, fixed top-K per step; does not adapt selection within a candidate pool.
IRCoT [10]	Retrieval interleaved with chain-of-thought steps	Open corpus, several LLM calls per question; fixed top-K per step
Self-RAG [11]	On-demand retrieval with critique of passages and outputs	Generation-time critique; requires a fine-tuned generator
FLARE [12], DRAGIN [13]	Retrieval triggered by generation-time uncertainty or information needs	Decides when to retrieve, not which candidates form the evidence set
CRAG [24]	Evaluation and correction of retrieved documents	Corrects retrieval quality with web search; no structure-aware selection
Adaptive-k [17], DynamicRAG [18], SetR [19]	Number or set of passages passed to the generator	One selection mechanism for all questions; DynamicRAG and SetR use LLM rerankers
Proposed	Evidence budget and selection structure (sequential vs. parallel)	Closed pool of 10 candidates; encoder-based rerankers; routing by benchmark-defined cardinality

2.3. Passage Reranking, Evidence Budgets, and Supporting-Document Selection

Neural passage reranking improves retrieval precision over lexical matching; monoBERT treats reranking as pointwise relevance classification [8], ColBERT preserves much of this effectiveness through late interaction [25], and large language models can rerank passages zero-shot in a listwise manner [26], at the cost of one or more LLM calls per question. In retrieve-then-rerank pipelines, however, the reranked list is truncated to a fixed top-K before generation [3], which creates a trade-off in multi-hop QA: a small K omits evidence for questions with many supporting documents, whereas a large K adds distractors, and both noisy passages [5] and longer contexts [4], [6] can degrade generation. For supporting-document selection in the HotpotQA distractor setting, SAE selects relevant documents before answering [14], the easy-to-hard FE2H selector identifies an anchor passage and uses it to select the next one [15], and Beam Retrieval keeps several partial evidence chains per hop, reaching 97.52% retrieval exact match on the English development set [16]; all apply one procedure to every question. This paper adopts the easy-to-hard principle of FE2H for its two-gold-document branch and evaluates it against controlled fixed-K baselines. The four-document reranker is initialized from the trained first-stage scorer of the two-gold-document branch, following the observation that transfer learning can improve sample efficiency when tasks share structure [27]; Section 4.3.3 tests whether this initialization is necessary.

A recent line of work adapts the evidence budget itself: Adaptive-k selects the number of passages from the distribution of similarity scores [17], DynamicRAG trains an LLM-based reranker with reinforcement learning to adjust the order and number of passages [18], and SetR selects passage sets for collective coverage [19]. These methods show that a fixed K is not optimal, but each applies a single mechanism to all questions and relies on a large language model. The present study examines a lightweight alternative in which both the budget and the selection structure follow the benchmark-defined evidence cardinality.

2.4. Indonesian Language Processing

Indonesian has productive affixation through prefixes, suffixes, and confixes, and rule-based stemming such as the Nazief-Adriani confix-stripping algorithm has been used to reduce vocabulary fragmentation in Indonesian information retrieval [28]–[30]. Multilingual pretrained encoders such as multilingual BERT transfer cross-lingual representations reasonably well without explicit cross-lingual supervision [31], and XLM-RoBERTa extends this line of work with larger-scale cross-lingual pretraining [32], which motivates their use as encoders for the routing classifier and the rerankers in this study. Supporting-document selection for multi-hop QA in Indonesian, however, has received little attention, largely because no natively authored Indonesian multi-hop benchmark with supporting-document annotations is available; this study therefore evaluates on machine-translated versions of HotpotQA and 2WikiMultihopQA. Whether stemming still helps a neural cross-encoder whose subword

tokenizer already shares representations across inflected forms has, to our knowledge, not been tested for Indonesian supporting-document selection; Section 4.3.5 examines it.

2.5. Research Gap

Taken together, the reviewed literature leaves three limitations. Adaptive and iterative RAG methods adapt retrieval decisions over open corpora and still pass a fixed top-K at each step [9]–[13], [24]; supporting-document selectors reach high accuracy in English but apply one selection procedure to every question [14]–[16]; and dynamic-budget rerankers adapt the number of passages but not the structure of selection, and depend on LLM-based rerankers [17]–[19]. To our knowledge, no prior study has examined, for Indonesian MHQA, an encoder-based reranker that adapts both the evidence budget and the selection structure within a closed candidate pool, together with isolated ablations of each component and an explicit diagnostic analysis of whether the routing reflects evidence structure or benchmark-origin confounding. Section 3 explains how each design choice of the proposed method responds to these limitations.

3. Proposed Method

Fig. 1 summarizes the end-to-end workflow; each box is labeled with the subsection that describes it.

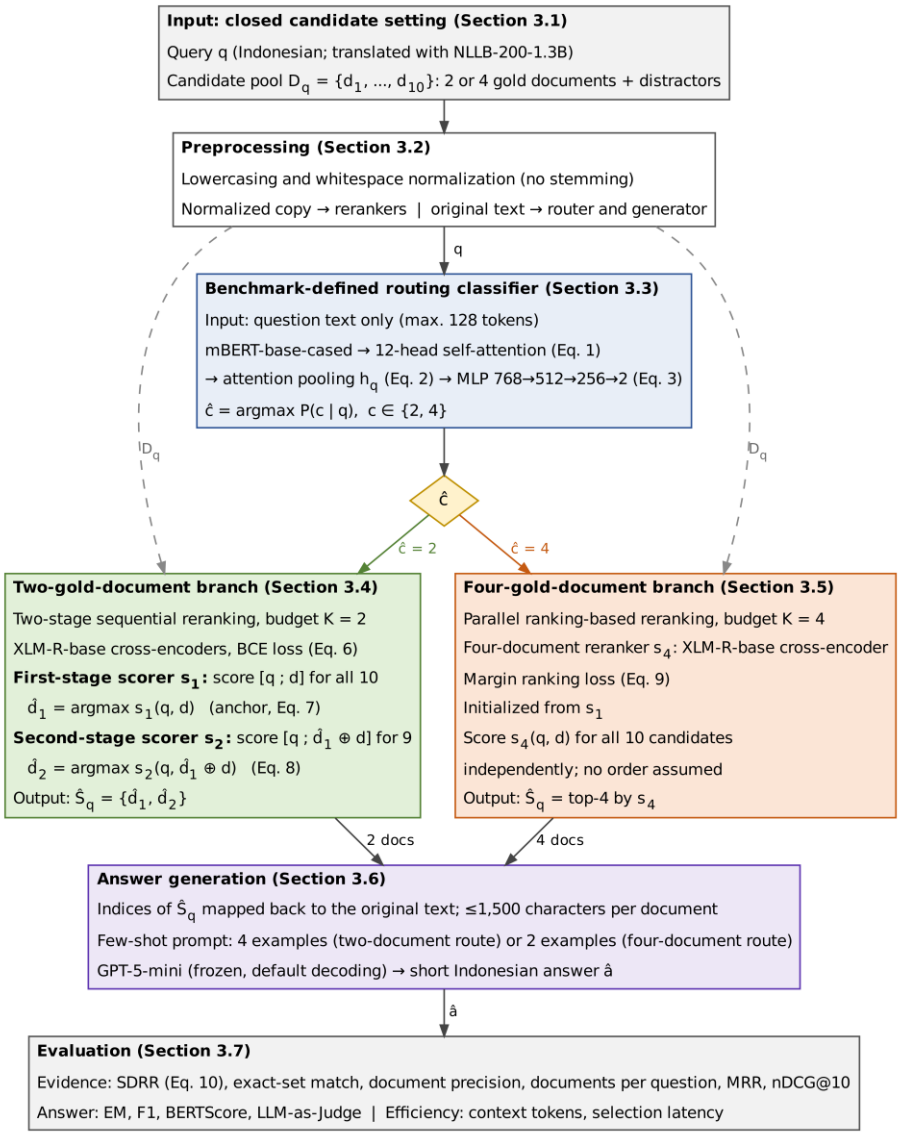


Figure 1. Overview of the proposed system workflow

Each question arrives with a closed pool of ten candidate documents (Section 3.1); the normalized text is passed to the rerankers, while the original text is kept for the routing classifier and for generation (Section 3.2); the routing classifier reads only the question and predicts the two-gold-document or four-gold-document branch (Section 3.3); the predicted branch determines both the reranking strategy and the evidence budget, two-stage sequential reranking of two documents (Section 3.4) or parallel ranking-based reranking of four documents (Section 3.5); and the selected documents are combined with the query in a few-shot prompt for the LLM-based generator (Section 3.6). Section 3.7 defines the metrics and statistical tests, and Section 3.8 lists the configuration of every model and the model-selection protocol.

The design responds to the limitations identified in Section 2.5 in four ways. First, because a fixed top-K either omits evidence or adds distractors (Section 2.3), the evidence budget is tied to the benchmark-defined number of gold documents instead of being fixed globally. Second, because a two-document bridge question usually exposes the second document only through an entity mentioned in the first, a single-pass scorer that judges every candidate against the question alone is poorly suited to it; the two-gold-document branch therefore adopts the easy-to-hard principle of FE2H [15] and conditions the second selection on the first. Third, because a four-document bridge-comparison question consists of two independent chains whose documents have no natural order, a four-step sequential selector would impose an arbitrary order and compound errors; the four-gold-document branch therefore scores all candidates in parallel with a ranking objective that directly optimizes the separation between gold and non-gold candidates, which is what top-K selection requires. Fourth, because adaptive, iterative, and LLM-based reranking methods require one or more LLM calls per question [9], [10], [18], and [26], all routing and reranking decisions are made by base-size encoders, so that adaptation adds milliseconds rather than LLM calls (Section 4.4).

As illustrated in Figure 1, the proposed system begins with a closed candidate pool containing ten documents per question, followed by text preprocessing. The routing classifier then predicts whether a question belongs to the two-gold-document or four-gold-document branch. Based on this prediction, the system applies either two-stage sequential reranking using the first-stage scorer s_1 and second-stage scorer s_2, or parallel ranking-based reranking using the four-document reranker s_4. The selected supporting documents are subsequently combined with the query in a few-shot prompt for answer generation. Finally, the system is evaluated using the metrics and statistical tests described in the following methodology sections.

3.1. Problem Formulation and Closed Candidate Reranking Setting

This study is conducted under a closed candidate reranking setting: each question q is paired with a fixed pool of ten candidate documents $D_q = \{d_1, \dots, d_{10}\}$ (fewer for a small number of incomplete pools, Section 3.2), consisting of the gold supporting documents together with distractor documents drawn from the underlying benchmark's distractor set; each candidate d carries a binary gold label $y_d \in \{0,1\}$ from the benchmark annotation, and the gold set of each question is $S_q = \{d \mid d \in D_q,\ y_d = 1\}$, the annotated supporting documents present in the pool. The task is therefore supporting-document selection from a known pool rather than open-domain retrieval over a large corpus, and reported recovery rates should be interpreted accordingly. Each question additionally carries a benchmark-defined cardinality label $c \in \{2,4\}$, the number of supporting documents annotated by its source benchmark, indicating whether it originates from the two-gold-document branch or the four-gold-document branch; this label is used to train the routing classifier described in Section 3.3, and its predicted value $\hat{c}$ determines which reranking strategy processes a given question at evaluation time. In tables, the two branches are abbreviated as two-doc and four-doc.

The two cardinalities were chosen because they represent structurally different evidence patterns rather than difficulty levels. In a two-gold-document question, typically a HotpotQA bridge question, the second document is usually reachable only through an entity in the first, so the evidence forms one dependent chain, which sequential selection is designed to exploit. A four-gold-document bridge-comparison question in 2WikiMultihopQA consists of two independent two-hop chains, for example, film A to its director and film B to its director, joined by a comparison; the four documents have no single order, so a four-step sequential selector

would impose a non-existent order and compound errors, and parallel ranking is the natural alternative. Two and four are also the smallest and largest gold-set sizes available in distractor-style benchmarks with complete annotation and a fixed ten-document pool, since bridge-comparison is the only 2WikiMultihopQA type with more than two supporting documents [2]; the fixed pool size keeps the branches comparable, and the two sizes expose the fixed-K trade-off most clearly ($K = 2$ cannot cover four documents, whereas $K = 4$ adds two distractors to every two-document question). Because the cardinalities coincide with the source datasets, Section 4.1 examines whether routing reflects evidence structure or dataset origin.

3.2. Dataset and Preprocessing

The two-gold-document branch is built from HotpotQA [1]; the four-gold-document branch is built from a filtered subset of 2WikiMultihopQA restricted to bridge-comparison instances with exactly four unique gold supporting-document titles, rather than the full corpus [2]. All questions, answers, document titles, and document contents were translated into Indonesian using NLLB-200-1.3B [7]. After dataset selection, 40,000 instances were obtained. The 16,000 training instances of each branch were split at random (seed 42) into 14,400 training and 1,600 validation instances, giving 28,800 training, 3,200 validation, and 8,000 evaluation instances (Table 2); no evaluation questions occur in the training or validation data. Each instance is paired with a fixed candidate pool of ten documents; in the two-gold-document branch, 147 training, 15 validation, and 35 evaluation instances have incomplete pools (fewer than ten candidates or not exactly two gold documents in the pool), which are excluded from reranker training but retained in every evaluation, where their gold set consists of the annotated supporting documents present in the pool (Section 3.1); these 35 evaluation questions can change any two-gold-document figure by at most 0.88 points. The text used for candidate scoring is lowercased and whitespace-normalized only, whereas the routing classifier and the answer generator receive the original text. Sastrawi stemming [28], [30], which was part of an earlier version of this pipeline, is not applied because Section 4.3.5 shows that it degrades the scorers. Translation quality, including corpus-level and back-translation-based metrics, is not evaluated in this article; it is examined in separate, not yet published work on the benchmark construction pipeline.

Table 2. Dataset composition by benchmark-defined branch

Split	HotpotQA (two-gold-document)	2WikiMultihopQA bridge comparison (four-gold-document)	Total
Training	14,400	14,400	28,800
Validation	1,600	1,600	3,200
Evaluation	4,000	4,000	8,000
Total	20,000	20,000	40,000

3.3. Benchmark-Defined Routing Classifier

The routing classifier (hereafter, the router) predicts the cardinality label $\hat{c} \in \{2,4\}$ of a question q from the question text only. It consists of a bert-base-multilingual-cased encoder [33] (maximum sequence length 128 tokens) and an attention-based classification head [34]. The encoder produces the token representations $H \in \mathbb{R}^{T\times 768}$ of the T question tokens, to which the head applies a 12-head scaled dot-product self-attention layer without an output projection:

$$H = \mathrm{mBERT}(q) \in \mathbb{R}^{T\times 768}, \qquad A = [a_1, \dots, a_T] = \mathrm{MHA}\big(HW_Q, HW_K, HW_V\big). \tag{1}$$

where W_Q, W_K, and W_V are learned projections, and padding positions are masked. The attended token vectors are pooled into a single question representation h_q with attention weights α_t computed from learned scores e_t,

$$e_t = u^{\top} \tanh(Ua_t + b_u), \qquad \alpha = \mathrm{softmax}(e), \qquad h_q = \sum_{t=1}^{T} \alpha_t \, a_t. \tag{2}$$

where U, b_u, and u are learned parameters (a 768→256→1 network with tanh activation). Thus, h_q denotes the pooled representation after the self-attention layer and attention pooling, not the raw encoder output. A multilayer perceptron (768→512→256→2, ReLU activations, dropout 0.3 and 0.2) maps h_q to these class logits z, and the probability of class c is the corresponding component of the softmax output, the two components of z being assigned to $c = 2$ and $c = 4$, so that $\hat{c} = \arg\max P(c \mid q)$.

$$z = \mathrm{MLP}(h_q), \qquad P(c \mid q) = [\mathrm{softmax}(z)]_c, \qquad \hat{c} = \arg\max_{c \in \{2,4\}} P(c \mid q). \tag{3}$$

During training, the embedding layer and the first eight BERT layers are frozen, while the remaining four layers, the attention mechanism, and the classification head are fine-tuned with AdamW (learning rate 2e-5, weight decay 0.01, batch size 32), gradient clipping at a maximum norm of 1.0, and a linear warmup-then-decay learning-rate schedule with 10% warmup. The model is trained for five epochs by minimizing the cross-entropy loss over the two classes.

$$L_{\mathrm{cls}} = -\frac{1}{M}\sum_{m=1}^{M} \log P(c_m \mid q_m) \tag{4}$$

where $M = 28{,}800$ is the number of training questions and c_m the benchmark-defined label of question q_m. The checkpoint with the highest accuracy on the 3,200 validation questions is used; the evaluation set is not used for model selection (Section 3.8).

3.4. Two-Stage Sequential Reranking for Two-Gold-Document Questions

The two-stage sequential reranking of the two-gold-document branch uses two scorers, the first-stage scorer s_1 and the second-stage scorer s_2, and the parallel ranking-based reranking of the four-gold-document branch (Section 3.5) uses the four-document reranker s_4. All three are XLM-RoBERTa-Base cross-encoders [32] with separate parameters; each encodes the concatenation $[q; x]$ of the query and a candidate text x and maps its [CLS] representation h_k to a scalar relevance score:

$$s_k(q, x) = w_k^{\top} h_k([q; x]) + b_k, \qquad k \in \{1,2,4\}. \tag{5}$$

where w_k and b_k are learned parameters. For s_1 and s_4 the candidate text is a single document, $x = d_i$; for s_2 it is the concatenation of the selected anchor document and a candidate, $x = \hat{d}_1 \oplus d_i$. At inference, the raw score is used for ranking. The scorers s_1 and s_2 are trained as pointwise classifiers with the binary cross-entropy loss over relevance labels $y_j \in \{0,1\}$:

$$\mathcal{L}_{\mathrm{BCE}} = -\frac{1}{|B|}\sum_{j \in B}\left[y_j \log \sigma\left(o_j\right) + \left(1 - y_j\right)\log\left(1 - \sigma(o_j)\right)\right]. \tag{6}$$

where B is a mini-batch of query-candidate pairs (q_j, x_j), $o_j = s_k(q_j, x_j)$ is the score of pair j under the scorer being trained $k \in \{1,2\}$, and σ is the sigmoid function.

Two-stage sequential reranking then proceeds as follows. In the first stage, every candidate is scored independently by s_1, and the highest-scoring document is selected as the anchor:

$$\hat{d}_1 = \underset{d_i \in D_q}{\mathrm{argmax}}\, s_1(q, d_i) \tag{7}$$

In the second stage, the remaining nine candidates are scored by s_2, whose input contains both the anchor and the candidate, so that the score of a candidate depends on the evidence already selected:

$$\hat{d}_2 = \arg\max_{d_i \in D_q \setminus \{\hat{d}_1\}} s_2(q, \hat{d}_1 \oplus d_i). \tag{8}$$

The selected set of the two-gold-document branch is $\hat{S}_q = \{\hat{d}_1, \hat{d}_2\}$ (Algorithm 1). The first-stage scorer s_1 is trained on all ten candidates of each training question (two positives and eight negatives; learning rate 2e-5, three epochs, batch size 16, maximum input length

384 tokens during training and 512 tokens at inference). The second-stage scorer s_2 is trained with teacher forcing: each of the two gold documents serves in turn as the anchor $\hat{d}_1$, the other gold document is the positive candidate, and four distractors resampled in every epoch are the negatives (learning rate 2e-5, two epochs, batch size 16, maximum input length 512 tokens). At inference, the anchor is the one chosen by s_1, so first-stage errors propagate to the second stage. The epoch of each scorer is selected on the validation split, using top-1 accuracy for s_1 and the SDRR of the complete two-stage procedure for s_2.

Algorithm 1. Two-Stage Sequential Reranking (Two-Gold-Document Branch)

INPUT: Query q, candidate pool D_q of size 10, first-stage scorer s_1 and second-stage scorer s_2 (Eq. (5))

OUTPUT: Selected supporting-document set $\hat{S}_q$

1: For each candidate document d_i in D_q, compute the first-stage score $s_1(q, d_i)$.
2: Select the anchor document $\hat{d}_1$ as the candidate with the highest score.
3: Remove $\hat{d}_1$ from the candidate pool.
4: For each remaining candidate d_i, compute the second-stage score $s_2(q, \hat{d}_1 \oplus d_i)$.
5: Select $\hat{d}_2$ as the remaining candidate with the highest second-stage score.
6: Return $\hat{S}_q = \{\hat{d}_1, \hat{d}_2\}$

3.5. Parallel Ranking-Based Reranking for Four-Gold-Document Questions

In the four-gold-document branch, all ten candidates are scored independently by the four-document reranker s_4 (Eq. (5)), and the selected set $\hat{S}_q$ consists of the four highest-scoring candidates, without any order among them. Because only membership in the top four matters, s_4 is trained with a pairwise margin ranking loss, which requires every gold candidate of a question to score at least a margin m above every non-gold candidate of the same question, rather than with a calibrated pointwise objective (Section 4.3.3 compares both):

$$\mathcal{L}_{\text{rank}} = \frac{1}{|\Omega|} \sum_{q \in \Omega} \frac{1}{|P_q||N_q|} \sum_{p \in P_q} \sum_{n \in N_q} \max\big(0,\, m - s_4(q,p) + s_4(q,n)\big). \tag{9}$$

where Ω is the set of training questions in a mini-batch, $P_q = S_q$ and $N_q = D_q \setminus S_q$ are the gold and non-gold candidates of question q ($|P_q| = 4$ and $|N_q| = 6$ for every training question), and $m = 0.5$. The configuration of s_4 was selected on the 1,600 four-gold-document validation questions from a grid that crosses initialization, either from the trained first-stage scorer s_1 of Section 3.4 (transfer) or from the original XLM-RoBERTa weights, with the learning rate (5e-6 or 2e-5) under the ranking loss, plus a transfer variant trained with the binary cross-entropy of Eq. (6); each configuration was trained for two epochs (batch size 4 with gradient accumulation over 4 steps, 5% warmup, maximum input length 384 tokens during training and 512 tokens at inference). The selected configuration, transfer initialization with the ranking loss and learning rate 2e-5 after two epochs, is used in the pipeline; Section 4.3.3 reports the full grid.

3.6. Answer Generation

After reranking, the indices of the selected documents are mapped back to the original text, and the documents are concatenated with the query into a few-shot prompt [35] (four examples for the two-gold-document branch and two for the four-gold-document branch) instructing the model to answer concisely in Indonesian, with each document truncated to 1,500 characters. The prompt template accepts any number of documents, so the same template is used for every evidence context in Section 4.3.4. Answer generation uses GPT-5-mini through the OpenAI API (October 2026); the API exposes a fixed default sampling configuration, so decoding parameters were not tuned, and Section 4.3.4 measures the resulting run-to-run variance by repeating generation. Generated answers are evaluated with exact match and token-level F1 after answer normalization, with BERTScore F1 [36] computed with bert-base-multilingual-cased without baseline rescaling, and with an LLM-as-Judge [37] (gpt-4o-mini, temperature 0) that decides whether the generated answer has the same meaning as the reference answer, ignoring differences in spelling, date or number format, and full versus

short names. Answer generation is evaluated separately from supporting-document selection, since reranking errors and generation errors affect final QA performance differently.

3.7. Evaluation Metrics and Statistical Testing

The primary evaluation metric is the Supporting-Document Recovery Rate (SDRR), the proportion of questions whose complete gold supporting-document set is contained in the selected set, so that partial recovery is not counted as success:

$$\text{SDRR} = \frac{1}{|E|}\sum_{q\in E}\mathbb{1}\left(S_q \subseteq \hat{S}_q\right) \tag{10}$$

where E is the set of evaluation questions, S_q is the gold supporting-document set of question q (Section 3.1), $\hat{S}_q$ is the set selected by the method under evaluation (Eqs. (7) and (8) or the top four candidates under s_4 for the proposed method, chosen by the predicted branch; the top-K candidates for a fixed-K baseline), and $\mathbb{1}$ is the indicator function. Because SDRR measures containment and is non-decreasing in the number of selected documents, four complementary set-level measures are reported: exact-set match $\hat{S}_q = S_q$, document precision $\left(|S_q \cap \hat{S}_q|\right)/|\hat{S}_q|$ and recall $\left(|S_q \cap \hat{S}_q|\right)/|S_q|$ averaged over questions, and the mean number of documents passed to the generator.

For the proposed method, the number of selected documents equals the predicted cardinality, i.e., $|\hat{S}_q| = \hat{c}$. SDRR and exact-set match therefore differ only on misrouted questions. Ranking quality is measured on the complete ordering of the ten candidates using the mean reciprocal rank (MRR) of the first gold document, mean average precision (MAP), and nDCG@10 with binary relevance, together with Recall@K ($K = 2$ and 4), the fraction of gold documents among the top K positions of the ordering, averaged over questions. Context efficiency is measured as the number of context tokens passed to the generator (using the o200k_base tokenizer, after truncation) and as the wall-clock latency of routing and reranking. Routing is evaluated with accuracy and class-wise precision, recall, and F1-score. Paired comparisons use McNemar's exact test on per-instance correctness and paired bootstrap 95% confidence intervals over 10,000 resamples [38]; when several comparisons are made on the same data, p-values are adjusted with the Holm-Bonferroni procedure.

3.8. Training Configuration and Model Selection

Table 3. Configuration of the trained components (final pipeline). Values in parentheses give the epoch selected on the validation split

Setting	Routing classifier	First-stage scorer s_1	Second-stage scorer s_2	Four-document reranker s_4
Encoder	mBERT-base-cased (embeddings and 8 layers frozen)	XLM-R-base	XLM-R-base	XLM-R-base
Initialization	Pretrained	Pretrained	Pretrained	Trained s_1
Input	Question (original text)	$[q\,;\,d]$	$[q\,;\,\hat{d}_1 \oplus d]$	$[q\,;\,d]$
Max. length (train/inference)	128 / 128	384 / 512	512 / 512	384 / 512
Loss	Cross-entropy, Eq. (4)	BCE, Eq. (6)	BCE, Eq. (6)	Margin ranking, m = 0.5, Eq. (9)
Training data	28,800 questions	14,253 questions × 10 candidates	14,253 questions × 2 anchors × (1 positive + 4 negatives)	14,400 questions × 10 candidates
Learning rate	2e-5	2e-5	2e-5	2e-5 (grid: 5e-6, 2e-5)
Batch size	32	16	16	4 × 4 accumulation
Epochs (selected)	5 (1)	3 (3)	2 (2)	2 (2)
Warmup	10%	10%	10%	5%
Selection criterion (validation)	Accuracy	Top-1 accuracy	Two-stage SDRR	SDRR at top 4

Table 3 lists the configuration of every trained component. All components share one protocol: training uses only the training split, every checkpoint or configuration choice (the epoch and, for the four-document reranker, also the initialization, learning rate, and loss) is made on the validation split, and the evaluation set is used once, after all choices have been fixed. All models use AdamW with weight decay 0.01, gradient clipping at norm 1.0, a linear warmup-then-decay schedule, and random seed 42; the rerankers are trained with fp16 mixed precision on a single NVIDIA A100 40 GB GPU. Each configuration was trained with a single seed, so seed variance is not reported (Section 6). The source code, configurations, and data-preparation scripts are publicly available (see the Data Availability Statement).

4. Results and Discussion

4.1. Routing Classifier: Training and Diagnostics

The routing classifier reached 100.00% validation accuracy after the first epoch (training loss 0.1114), and its validation accuracy stayed between 99.97% and 100.00% over the remaining four epochs while the training loss fell to 0.0009; the first epoch was therefore selected. On the evaluation set, it reaches 99.94% accuracy: 5 of 8,000 questions are misrouted (3 two-gold-document questions sent to the four-gold-document branch and 2 in the opposite direction), and class-wise precision, recall, and F1-score lie between 99.92% and 99.95% (Table 4). Routing these five questions through their correct branch raises SDRR by only 0.03 points (Section 4.3.1).

Saturated in-distribution accuracy is, however, not evidence that the router detects reasoning structure. Because the two-gold-document class is drawn entirely from HotpotQA and the four-gold-document class entirely from the filtered 2WikiMultihopQA subset, the routing label is confounded with the dataset of origin, and a classifier could reach this accuracy from benchmark-specific writing conventions alone [23], [39]. Two diagnostics test this. First, four shallow probes with no access to any learned semantic representation were trained on the same 28,800 training questions (14,400 per class) and evaluated on the same balanced evaluation set (4,000 per class), with hyperparameters fixed in advance: logistic regression on word unigram and bigram TF-IDF features, on character 2-5-gram TF-IDF features, on word TF-IDF features after masking capitalized words, quoted titles, and numbers, and on eleven length and punctuation features with no lexical content. Table 4 reports the accuracy, class-wise precision, recall, and F1-score, and the confusion matrix of each probe and of the router.

Table 4. Routing-label diagnostics on the evaluation set (4,000 questions per class). P, R, and F1 are class-wise percentages; the last two columns give the confusion matrix (correct / misrouted questions per true class)

Classifier	Accuracy	Two-doc P / R / F1	Four-doc P / R / F1	Two-doc: correct/wrong	Four-doc: correct / wrong
Word TF-IDF + LR	99.58	99.87 / 99.28 / 99.57	99.28 / 99.88 / 99.58	3,971 / 29	3,995 / 5
Character n-gram TF-IDF + LR	99.76	99.92 / 99.60 / 99.76	99.60 / 99.92 / 99.76	3,984 / 16	3,997 / 3
Entity-masked TF-IDF + LR	99.72	99.65 / 99.80 / 99.73	99.80 / 99.65 / 99.72	3,992 / 8	3,986 / 14
Length/style features + LR	77.35	72.91 / 87.05 / 79.35	83.93 / 67.65 / 74.92	3,482 / 518	2,706 / 1,294
Proposed router (mBERT)	99.94	99.95 / 99.92 / 99.94	99.93 / 99.95 / 99.94	3,997 / 3	3,998 / 2

The three lexical probes separate the classes almost as well as the router, and they do so for both classes: their class-wise F1-scores differ by at most 0.01 points, so the high accuracy is not an artifact of one dominant class. The weights of the word-level probe identify the source of this predictability. The n-grams most indicative of the four-gold-document class are film, sutradara (director), kedua film (both films), and yang sama (the same), whereas those of the two-gold-document class are general question words and topics such as siapa (who), tahun (year), band, and penulis (writer). The probes therefore separate topic and question

template rather than evidence cardinality; 99.80% of the four-gold-document training questions contain the word film, against 13.35% of the two-gold-document questions. The length/style probe, which has no lexical access, reaches 77.35%, showing that surface form alone is also informative. Accordingly, the routing classifier is interpreted strictly as a benchmark-defined controller, not as a general-purpose complexity detector.

The probes cannot show which cue the router uses when evidence cardinality and dataset origin are separated. Second, the router was applied to 2WikiMultihopQA questions it had never seen: 900 comparison, compositional, and inference questions from the validation split of the original benchmark, which have two supporting documents but share the template-based style of the four-gold-document branch, and, as a control, 300 bridge-comparison questions with four supporting documents drawn from unused training-split instances. Questions were translated with the same NLLB-200-1.3B model, and identifiers used anywhere in the benchmark were excluded. Table 5 reports the results.

Table 5. Cross-source routing test on unseen 2WikiMultihopQA questions

Question type	Gold documents	n	Routed to two-doc branch	Routed to four-doc branch
Comparison	2	300	86.67%	13.33%
mentioning a film	2	111	63.96%	36.04%
not mentioning a film	2	189	100.00%	0.00%
Compositional	2	300	81.00%	19.00%
mentioning a film	2	111	48.65%	51.35%
not mentioning a film	2	189	100.00%	0.00%
Inference	2	300	100.00%	0.00%
All two-document types	2	900	89.22%	10.78%
Bridge-comparison (control)	4	300	0.00%	100.00%

The router sends every control question to the four-gold-document branch and 89.22% of the two-document questions to the two-gold-document branch, so it does not simply detect the source dataset. Its errors are, however, entirely topical: all 97 misrouted questions mention a film, with misrouting rates of 36.04% and 51.35% for comparison and compositional questions about films (for example, *Di mana sutradara* film Please *lahir*?, 'Where was the director of the film Please born?') and 0% for the 378 comparison and compositional questions that do not mention a film. This matches the training data, because 2WikiMultihopQA builds bridge-comparison questions from film-director relations [2]. The router thus combines a film-director topic with a template-like question pattern, a shortcut [39] rather than a measure of reasoning structure. The router of the earlier version of this study, selected on the evaluation set after three epochs, misrouted 5.67% of these questions, whereas the validation-selected router misroutes 10.78%: with validation accuracy saturated from the first epoch, in-distribution validation cannot distinguish checkpoints that generalize differently out of distribution. Misrouting in this direction adds two non-supporting documents to the context, and it implies that the four-gold-document results characterize film-director questions specifically.

4.2. Supporting-Document Recovery and Comparison with Baselines

Under the full routed pipeline, the framework recovers the complete supporting-document set for 95.40% of the 8,000 evaluation questions: 92.58% in the two-gold-document branch and 98.22% in the four-gold-document branch (Table 6). It selects 3.00 documents per question (2.00 and 4.00 per branch) with 97.69% document precision, and its exact-set match is 95.36%, 0.04 points below its SDRR because of misrouted questions. These figures combine routing, branch-specific reranking, and their interaction.

The framework is compared with two families of supervised fixed-K rerankers in the same closed setting, for $K \in \{2,3,4\}$; SDRR is non-decreasing in K, so the sweep stops at $K = 4$, the largest gold-set size. The single-model baseline applies the four-document reranker to every question and thus represents a non-adaptive system without routing or branch-specific models. The branch-matched baseline uses the same router and the same scorers as the proposed framework: the first-stage scorer for questions routed to the two-

gold-document branch and the four-document reranker otherwise, but selects a fixed top-K for every question; it isolates the effect of adapting the budget and selection structure from that of branch-specific scoring. Five unsupervised or zero-shot retrievers are listed separately as context. Table 6 reports SDRR together with exact-set match, evidence budget, and document precision.

Table 6. Supporting-document recovery on the evaluation set (8,000 questions). The upper block contains the proposed method and controlled baselines that share its data, candidate pools, and preprocessing; the lower block contains contextual baselines that are not controlled

Method	Docs per question	SDRR overall	SDRR two-doc	SDRR four-doc	Exact set match	Document precision
Proposed adaptive	3.00	95.40%	92.58%	98.22%	95.36%	97.69%
Branch-matched, $K = 2$	2.00	39.27%	78.55%	0.00%	39.27%	94.39%
Branch-matched, $K = 3$	3.00	44.85%	89.70%	0.00%	0.15%	81.59%
Branch-matched, $K = 4$	4.00	96.29%	94.35%	98.22%	49.26%	74.14%
Single-model, $K = 2$	2.00	33.41%	66.83%	0.00%	33.41%	91.13%
Single-model, $K = 3$	3.00	39.92%	79.85%	0.00%	0.15%	79.83%
Single-model, $K = 4$	4.00	92.80%	87.33%	98.28%	49.29%	73.22%
Contextual baselines (not controlled)						
TF-IDF, $K = 5$	5.00	38.91%	67.42%	10.40%	–	–
BM25, $K = 5$	5.00	43.14%	70.25%	16.02%	–	–
DPR, $K = 5$	5.00	42.51%	63.60%	21.43%	–	–
Multilingual-E5-Large, $K = 5$	5.00	52.90%	82.03%	23.77%	–	–
BGE-M3, $K = 5$	5.00	56.25%	77.20%	35.30%	–	–

Two comparisons must be distinguished. Against the single-model baseline, the framework is better on every evidence-selection measure (recovery, precision, and budget): SDRR is 2.60 points higher (95.40% vs. 92.80%; paired bootstrap $95\%\ \mathrm{CI}\ [2.02, 3.20]$; McNemar's exact test with 386 vs. 178 discordant questions, $p = 1.2 \times 10^{-18}$), entirely through the two-gold-document branch (92.58% vs. 87.33%; in the four-gold-document branch the two methods differ on two questions, $p = 0.50$), with 24.47 points higher document precision and one document fewer per question. Against the branch-matched baseline, which uses the same router and scorers, the result is a trade-off rather than an improvement. Fixed top-4 selection attains 0.89 points higher containment (96.29% vs. 95.40%; CI [0.39, 1.35]; 225 vs. 154 discordant questions, $p = 3.1 \times 10^{-4}$), all of it in the two-gold-document branch (94.35% vs. 92.58%), whereas the framework passes 25% fewer documents (3.00 vs. 4.00) with 23.55 points higher precision (97.69% vs. 74.14%; CI [22.99, 24.11]); on two-gold-document questions, top-4 selection passes about two non-supporting documents per question (precision 48.74% vs. 95.81%; this value is computed on the full routed pipeline and therefore includes the three misrouted two-gold-document questions, whereas the 95.85% in Tables 7 and 8 refers to the two-stage branch alone). In the four-gold-document branch, both methods reach the same SDRR (98.22%) and select the same documents except for the two misrouted questions. The advantage over the single-model baseline therefore comes from branch-specific scoring and two-stage sequential reranking, and the framework is not an unqualified improvement over fixed-K selection: it trades a small loss in containment for a compact, high-precision evidence set. Whether this trade-off matters downstream depends on the generator (Section 4.3.4).

The five unsupervised and zero-shot retrievers, each selecting $K = 5$ documents, were evaluated on the same evaluation questions in the earlier version of this study and are listed only as context; they differ from the proposed method in supervision and were not retrained. The multilingual embedding models perform best among them (52.90% and 56.25% SDRR) but remain far below the supervised rerankers.

SDRR and precision are set-level measures. Table 7 reports ranking-sensitive metrics over the full ordering of the ten candidates, which show where two-stage sequential reranking acts. For two-gold-document questions, the first document is chosen by the same first-stage

scorer in both strategies, so MRR is essentially unchanged (97.88% vs. 97.86%); conditioning on the anchor improves the placement of the second gold document, raising MAP from 94.04% to 97.05%, nDCG@10 from 96.61% to 98.08%, and Recall@2 from 88.81% to 95.85%. Applying the four-document reranker to two-gold-document questions gives the lowest ranking quality (nDCG@10 93.99%), which explains the weaker two-gold-document branch of the single-model baseline. In the four-gold-document branch, ranking is nearly perfect (MAP 99.88%, nDCG@10 99.96%).

Table 7. Ranking quality over the complete ordering of the ten candidates (evaluation set; binary relevance; percentages)

Branch and ordering	MRR	MAP	nDCG@10	Recall@2	Recall@4
Two-doc: first-stage scorer (single-stage)	97.88	94.04	96.61	88.81	97.15
Two-doc: two-stage (anchor, then second-stage scores)	97.86	97.05	98.08	95.85	98.42
Two-doc: four-document reranker	96.00	89.61	93.99	82.27	93.44
Four-doc: four-document reranker	100.00	99.88	99.96	49.99	99.57

4.3. Ablation Studies

Each ablation changes one component while keeping the data, the remaining components, and the metrics fixed. Every variant trained in this revision was selected on the validation split before being evaluated, and variants are compared with the pipeline configuration using McNemar's exact test on the evaluation set unless stated otherwise; results taken from the earlier version of this study (the stemming ablation and the zero-shot transfer figure) are marked as such.

4.3.1. Routing

Re-processing the five misrouted evaluation questions through their correct branch (oracle routing) changes SDRR from 95.40% to 95.43% (two questions gained and none lost; $p = 0.50$), with the two-gold-document branch unchanged (92.58%) and the four-gold-document branch moving from 98.22% to 98.28%. Routing errors are therefore negligible on this benchmark; their consequences outside it are examined in Section 4.1.

4.3.2. Selection Strategy in the Two-Gold-Document Branch

To isolate the selection strategy, two-stage sequential reranking is compared with single-stage top-K selection by the same first-stage scorer on the 4,000 two-gold-document questions (Table 8). At an equal budget of two documents, conditioning the second selection on the first raises SDRR from 78.57% to 92.58%, a gain of 14.00 points (paired bootstrap 95% CI [12.78, 15.20]; 634 vs. 74 discordant questions, $p = 7.8\times10^{-112}$), the largest single effect in this study. Single-stage selection needs a budget of four documents to exceed two-stage sequential reranking (94.35% vs. 92.58%), at which point about half of the selected documents are non-supporting (precision 48.74% vs. 95.85%).

Table 8. Selection strategy in the two-gold-document branch (4,000 questions)

Strategy	Docs per question	SDRR	Document precision
Single-stage top-2	2.00	78.57%	88.81%
Single-stage top-3	3.00	89.72%	63.28%
Single-stage top-4	3.99	94.35%	48.74%
Two-stage sequential (proposed)	2.00	92.58%	95.85%

Single-stage top-4 averages 3.99 documents because a few candidate pools contain fewer than four documents (Section 3.2); Table 6 reports the average over all 8,000 questions (4.00).

A component-level evaluation on the 3,965 questions with complete pools gives the same picture: the first-stage scorer ranks a gold document first for 96.09% of questions and places at least one gold document in its top two for 99.04%, but both gold documents for only 78.41%, so the second supporting document is the difficult one.

4.3.3. Training Regime of the Four-Gold-Document Branch

Table 9 reports the grid from which the four-document reranker was selected, with validation and evaluation SDRR at the top four for every configuration and epoch. Validation and evaluation results rank the ten configuration-epoch pairs almost identically (Spearman's $\rho = 0.985$), so selection on the validation split is reliable. Because the grid crosses initialization with learning rate under the ranking loss, it separates two factors that were confounded in the earlier version of this study. At the smaller learning rate (5e-6), transfer initialization is clearly beneficial (97.25% vs. 93.83% after two epochs), because training from the original XLM-RoBERTa weights converges poorly and even degrades in the second epoch. At the larger learning rate (2e-5), transfer initialization is ahead after the first epoch (97.82% vs. 97.15%) but not significantly after the second (98.28% vs. 97.97%; 41 vs. 29 discordant questions, $p = 0.19$). Transfer learning therefore accelerates convergence and makes training robust to the learning rate, but it does not significantly improve the final reranker once the learning rate is tuned. Fine-tuning itself is essential: in the earlier version of this study, a first-stage scorer applied zero-shot recovered all four gold documents for only 25.32% of questions. Under the same initialization and learning rate (5e-6), binary cross-entropy outperforms the ranking loss (97.82% vs. 97.25% after two epochs), but the loss was varied only at this learning rate; the configuration selected on validation, transfer initialization with the ranking loss at 2e-5, is also significantly better than the configuration used in the earlier pipeline (transfer, ranking loss, 5e-6, one epoch; 98.28% vs. 96.85%, $p = 4.6\times10^{-9}$).

Table 9. Training regime of the four-document reranker: SDRR at top four on the validation split (1,600 questions) and the evaluation set (4,000 questions). The asterisk marks the configuration selected on validation and used in the pipeline

Initialization	Loss	Learning rate	Validation, epoch 1	Validation, epoch 2	Evaluation, epoch 1	Evaluation, epoch 2
Transfer	Ranking	5e-6	96.50%	97.12%	96.85%	97.25%
Transfer	Ranking	2e-5	97.94%	98.38%*	97.82%	98.28%*
Scratch	Ranking	5e-6	95.38%	93.94%	95.60%	93.83%
Scratch	Ranking	2e-5	96.94%	98.12%	97.15%	97.97%
Transfer	BCE	5e-6	97.00%	97.81%	97.32%	97.82%

4.3.4. Evidence Context and Answer Quality

To test whether better evidence selection translates into better answers, the generator, prompt template, and few-shot examples were held fixed and only the evidence context was varied, on a stratified random subset of 1,000 evaluation questions (500 per branch, seed 42). Four contexts were compared: all ten candidates without reranking, branch-matched fixed-K=4 selection, the proposed adaptive selection, and the gold supporting documents as an upper bound. To separate context effects from the sampling variability of the generator, every context was generated twice in independent runs. Table 10 reports exact match with 95% bootstrap confidence intervals for both runs, token-level F1 for the first run, and the share of answers that are identical across the two runs.

Table 10. Answer quality under different evidence contexts (GPT-5-mini; 1,000 questions; two independent generation runs; 95% bootstrap confidence intervals in brackets)

Evidence context	Docs per question	EM, run 1	EM, run 2	F1, run 1	Identical answers
All ten candidates	9.99	62.60% [59.6, 65.6]	61.80% [58.7, 64.8]	75.45%	90.5%
Fixed-K=4 (branch-matched)	4.00	62.70% [59.6, 65.7]	61.70% [58.6, 64.7]	75.06%	90.8%
Adaptive (proposed)	3.00	62.20% [59.1, 65.2]	62.20% [59.2, 65.1]	74.93%	89.6%
Gold documents (upper bound)	3.00	63.30% [60.3, 66.3]	63.80% [60.8, 66.7]	76.41%	90.6%

No evidence context differs significantly from another. Adaptive evidence is 0.4 points below all ten candidates in the first run and 0.4 points above them in the second, and 0.5 points below and above fixed-K=4 selection; for each run, the nine comparisons of adaptive evidence with the other contexts (three contexts, overall and per branch) have Holm-adjusted p-values of at least 0.34. The largest gap, 2.6 to 2.8 points below the gold documents on two-gold-document questions (unadjusted $p \approx 0.04$), is not significant after adjustment. The variability of the generator itself is of the same size: re-running the same prompts changes about 10% of the answers and moves EM by up to 1.0 point, with no run-to-run difference significant. Within this resolution, neither the 0.89-point containment advantage of fixed-K=4 selection nor the 23.55-point precision advantage of adaptive selection produces a detectable change in answer accuracy, and supplying all ten candidates does not yield a reliable gain once multiple comparisons and generator variance are taken into account, which corrects the conclusion drawn from a single run in the earlier version of this study. For a strong instruction-tuned generator and about ten short paragraphs, distractors appear to be largely ignored, so the benefit of adaptive reranking lies in a much shorter context (Section 4.4) and in evidence that identifies the supporting documents; the experiments do not establish an improvement in answer accuracy. These conclusions are limited to one generator and a 1,000-question subset, whose confidence intervals are about ±3 points wide. Even the gold documents yield only 63.30% to 63.80% exact match, which suggests that the gap between SDRR and answer exact match is dominated by answer-form and translation effects (Section 4.5).

4.3.5. Indonesian Stemming and Model-Selection Protocol

The earlier version of this study stemmed all scorer inputs with Sastrawi and selected checkpoints on the evaluation set. To evaluate stemming in isolation, two first-stage scorers were trained from the original XLM-RoBERTa weights with identical settings (binary cross-entropy, learning rate 2e-5, three epochs, seed 42) in a separate run from the earlier version of this study, on all 16,000 training questions of the two-gold-document branch (before the 1,600 validation questions of the current split were held out; no checkpoint was selected, and results are reported after every epoch), one on stemmed and one on unstemmed text, and evaluated on the 3,965 two-gold-document questions with complete pools. Because these two scorers are separate from the first-stage scorer of the final pipeline, their absolute values differ from those in Section 4.3.2 (96.09% and 78.41%); only the paired comparison between them is interpreted. Stemming lowers top-1 accuracy from 96.39% to 95.59% ($p = 0.014$) and the proportion of questions with both gold documents in the top two from 80.10% to 75.08% ($p = 3.8\times10^{-17}$), and the gap is present after every epoch (73.59% vs. 69.56%, 78.74% vs. 73.97%, and 80.10% vs. 75.08% after epochs 1 to 3). The cause is visible in the stemmed text: confix stripping removes meaning-bearing affixes, so that ditemukan (found) becomes temu, berikut (following) becomes ikut (to join), and meninggal (to die) becomes tinggal (to stay), which matters for questions that compare death dates, whereas the subword tokenizer of XLM-RoBERTa already shares representations across inflected forms. The pipeline was therefore retrained without stemming under the validation protocol of Section 3.8, which also adopts the four-gold-document configuration selected in Section 4.3.3. Relative to the earlier pipeline, SDRR rises from 93.16% to 95.40% (+2.24 points, CI [1.74, 2.76]; 311 vs. 132 discordant questions, $p = 9.8\times10^{-18}$), with gains in both branches (from 89.60% to 92.58% and from 96.73% to 98.22%). This comparison verifies directly that the revised configuration is better, but because it changes stemming, the data split, and the four-gold-document configuration together, it does not attribute the gain to a single factor; the isolated effect of stemming is the one reported above.

4.4. Efficiency: Context Length and Selection Latency

Efficiency is reported only for the quantities that were measured. Table 11 lists the number of context tokens passed to the generator on the 1,000-question subset and the wall-clock latency of routing and reranking, measured on 500 evaluation questions (250 per branch) processed one at a time on an NVIDIA A100 40 GB GPU (fp16 rerankers, fp32 router, 20 warmup questions). Adaptive evidence averages 355 context tokens, 68.5% fewer than all ten candidates (1,128) and 26.8% fewer than fixed-K=4 selection (485), and is as short as the gold documents themselves (355). The selection stage, however, is the slowest of the three supervised selectors: 39.6 ms per question on average, against 30.5 ms for branch-

matched and 19.4 ms for single-model fixed-K=4 selection. Routing accounts for about 11 ms, the difference between the branch-matched and single-model baselines, and the second-stage scorer adds about 20 ms on two-gold-document questions (49.9 vs. 30.4 ms), because it scores nine additional query-anchor-candidate pairs. The framework therefore trades 9 ms (relative to branch-matched) to 20 ms (relative to single-model fixed-K=4 selection) of selection latency per question for a context that is 26.8% to 68.5% shorter. The latency and monetary cost of the LLM call itself were not measured, so no end-to-end efficiency claim is made; because the prompt template is identical across contexts, the token reduction applies directly to the evidence part of the generator input.

Table 11. Context length passed to the generator and latency of the selection stage. Context tokens: o200k_base tokenizer, 1,000-question subset. Latency: routing and reranking per question in milliseconds, 500 questions, NVIDIA A100 40 GB

Selector	Docs per question	Context tokens	Mean latency	Median	p95	Mean two-doc / four-doc	Cross-encoder pairs per question
All ten candidates (no reranking)	9.99	1,127.5	–	–	–	–	–
Fixed-K=4, single model	4.00	–	19.4	19.5	22.0	19.6 / 19.2	10.0
Fixed-K=4, branch-matched	4.00	484.8	30.5	30.4	33.2	30.4 / 30.5	10.0
Adaptive (proposed)	3.00	355.0	39.6	39.7	52.4	49.9 / 29.4	14.4
Gold documents (upper bound)	3.00	354.9	–	–	–	–	–

4.5. Answer Generation Performance and Error Analysis

Using the evidence selected by the pipeline, GPT-5-mini generates a concise Indonesian answer for each of the 8,000 evaluation questions. Table 12 reports the lexical and semantic metrics together, overall and by branch, with 95% bootstrap confidence intervals.

Table 12. Answer generation performance on the evaluation set (8,000 questions; 95% bootstrap confidence intervals in brackets)

Scope	Exact match	F1-score	BERTScore F1	LLM-as-Judge
Overall	63.71% [62.6, 64.8]	76.17% [75.3, 77.0]	91.65% [91.4, 91.9]	86.56% [85.8, 87.3]
Two-gold-document branch	54.25% [52.7, 55.8]	72.00% [70.9, 73.1]	89.37% [89.0, 89.8]	83.90% [82.8, 85.0]
Four-gold-document branch	73.17% [71.8, 74.6]	80.33% [79.2, 81.5]	93.93% [93.6, 94.2]	89.22% [88.3, 90.2]

The four metrics agree on the ordering of the branches but not on the size of the gap. The two-gold-document branch is 18.92 points below the four-gold-document branch in exact match but only 5.32 points below it according to the LLM judge, and the judge accepts 86.56% of all answers, against 63.71% exact match. The gap between strict lexical matching and semantic acceptance is consistent with residual machine-translation noise, the distribution of answer types, and full-name versus short-name variation rather than with failures of evidence selection, since even the gold documents yield only about 63% exact match (Section 4.3.4).

To examine whether strict exact match penalizes particular answer forms, gold answers were grouped by rule into four types: Boolean (ya/tidak and equivalents), date or number (containing a digit or a month name), named entity or title (containing a capitalized word), and common noun or attribute (the remainder). Table 13 reports the number and share of each type with its metrics. The distribution is strongly imbalanced, and the two smallest types (766 and 291 questions) have wide confidence intervals, so differences between them should be interpreted with caution.

Table 13. Answer generation performance by answer type (8,000 questions; 95% bootstrap confidence interval of exact match in brackets)

Answer type	n	Share	Exact match	F1-score	LLM-as-Judge
Named entity/title	5,550	69.38%	68.47% [67.2, 69.7]	80.45%	87.21%
Boolean	1,393	17.41%	65.47% [63.0, 68.0]	65.64%	86.50%
Date/number	766	9.57%	36.03% [32.6, 39.4]	70.98%	87.60%
Common noun/attribute	291	3.64%	37.46% [32.0, 43.0]	58.52%	71.82%

Date and number answers show the lowest exact match (36.03%) but a much higher F1-score (70.98%) and the highest judge acceptance (87.60%), indicating formatting differences such as date order or numeral format introduced during translation rather than wrong answers. Common noun and attribute answers are the only type with markedly lower judge acceptance (71.82%), but with 291 questions, their estimate is the least precise.

Manual inspection of individual errors, carried out on the generation outputs of the earlier version of this study, shows three recurring categories. In lexical mismatches, a correct answer is phrased differently from the reference, for example "keduanya publik" ("both are public") for the reference "publik" (EM 0, F1 0.667). In entity-name variations, the same entity is rendered differently, for example "Cid Corman" for "Cid (Sidney) Corman" (EM 0, F1 0.800), consistent with machine translation of proper names. In the third category, the supporting documents are recovered correctly, but the generator selects the wrong entity, for example "Jacinto Benavente y Martínez" instead of "Rohinton Mistry" for a question about the more recent Nobel laureate. Together with the answer-type breakdown, these cases indicate that much of the gap between SDRR and answer exact match originates downstream of supporting-document selection.

5. Comparison with Previous Studies

Table 14 places the results of this study alongside published results of supporting-document selectors and adaptive RAG methods on the same source benchmarks. The upper part reports supporting-document selection in the ten-candidate distractor setting, the setting closest to this study; the lower part reports answer exact match of adaptive and iterative RAG methods, which operate in an open-domain setting.

Table 14. Published results on HotpotQA and 2WikiMultihopQA and the results of this study

Method	Data (language)	Setting and model	Metric	Result
SAE [14] †	HotpotQA dev (EN)	Distractor, 10 candidates	Retrieval EM	91.98%
FE2H [15] †	HotpotQA dev (EN)	Distractor, 10 candidates	Retrieval EM	96.32%
Beam Retrieval [16]	HotpotQA dev (EN)	Distractor, 10 candidates; DeBERTa-large	Retrieval EM	97.52%
Beam Retrieval [16]	2WikiMultihopQA dev, all types (EN)	Distractor, 10 candidates; DeBERTa-base	Retrieval EM	99.93%
This work	HotpotQA, 4,000 (ID, translated)	Closed, 10 candidates; XLM-R-base	Exact-set match	92.50%
This work	2WikiMultihopQA bridge-comparison, 4,000 (ID, translated)	Closed, 10 candidates; XLM-R-base	Exact-set match	98.22%
Single-step retrieval [9]‡	HotpotQA / 2WikiMultihopQA, 500 each (EN)	Open-domain, BM25; FLAN-T5-XL	Answer EM	34.40%/41.60%
Adaptive-RAG [9] ‡	HotpotQA / 2WikiMultihopQA, 500 each (EN)	Open-domain, BM25; FLAN-T5-XL	Answer EM	42.00%/40.60%
Multi-step, IRCoT [9], [10] ‡	HotpotQA / 2WikiMultihopQA, 500 each (EN)	Open-domain, BM25; FLAN-T5-XL	Answer EM	44.60%/49.60%
This work	HotpotQA / 2WikiMultihopQA-BC, 4,000 each (ID)	Closed, 10 candidates; GPT-5-mini	Answer EM	54.25%/73.17%

† As reported in [16], ‡ As reported in [9].

These comparisons are not controlled, and the differences are stated to keep the claims scoped. For supporting-document selection, pool size and source benchmarks match, but this

study uses machine-translated Indonesian text, a base-size encoder, and its own 4,000-question subsets, whereas the published selectors use the official English development sets and, for Beam Retrieval on HotpotQA, a large encoder; the published 2WikiMultihopQA figure also covers all question types rather than the bridge-comparison subset. Under these differences, the two-gold-document exact-set match of this study (92.50%) is about 4 to 5 points below English state-of-the-art selectors (96.32% to 97.52%) and slightly above the figure reported for SAE (91.98%), and the four-gold-document exact-set match (98.22%) is about 1.7 points below Beam Retrieval on 2WikiMultihopQA. The contribution of this study is therefore not a new state of the art but a controlled analysis of adaptive selection in an Indonesian setting; the effects of translation, encoder size, and evaluation subset on this gap have not been isolated. For answer quality, adaptive and iterative RAG methods retrieve from an open Wikipedia corpus with BM25, use different generators, and are evaluated on 500 English questions per dataset, whereas this study selects from a closed pool that always contains the gold documents, so its higher answer EM must not be read as superiority. Conceptually, the approaches are complementary: Adaptive-RAG, IRCoT, FLARE, DRAGIN, and CRAG decide whether, when, and how often to retrieve, whereas the proposed reranker decides how many and which retrieved candidates form the evidence set and could follow their retrieval step. Section 4.3.4 further shows that, for a strong generator, compact evidence yields no significant difference in answer quality.

6. Conclusions and Limitations

This study asked whether a lightweight encoder-based reranker that adapts both the evidence budget and the selection structure to a benchmark-defined evidence cardinality improves supporting-document selection for Indonesian multi-hop question answering in a closed candidate setting, and what each of its components contributes. The answer is qualified. Under a protocol in which every model choice was made on a validation split, the framework significantly outperforms a non-adaptive single-model reranker in evidence recovery and yields a compact, high-precision evidence set, but compared with fixed top-4 selection by the same scorers it represents a trade-off between containment and precision rather than an overall improvement. The ablations indicate that conditioning the second selection on the first is the component that matters most for chained evidence, that parallel ranking suffices when the evidence forms independent chains, and that two commonly assumed aids, transfer initialization and Sastrawi stemming, contribute little or even harm once the training regime is controlled for XLM-RoBERTa scorers. Compact evidence also shortens the generator context substantially at the price of 9 to 20 ms of additional selection time per question, yet with a strong LLM generator it did not improve answer accuracy, so its benefit lies in efficiency and in an explicit supporting-document set rather than in accuracy. All of these findings hold for the benchmark-defined setting; because the router relies on a topic shortcut outside the benchmark, they do not establish general reasoning-complexity detection or generalization to other question distributions. The contributions of this study are therefore the adaptive budget-and-structure design, a controlled evaluation protocol that keeps evidence recovery, precision, ranking quality, efficiency, and answer quality apart, and diagnostic evidence on what the router actually learns.

The main limitations follow from the setting. First, the routing label coincides with the source dataset, and outside the benchmark the router relies on a film-director topic shortcut, so it is a benchmark-defined controller rather than a detector of reasoning structure, and the four-gold-document results characterize film-director questions. Second, only two evidence cardinalities and a closed pool that always contains the gold documents were studied. Third, the benchmark is machine-translated rather than natively authored. Fourth, each model was trained with a single seed; the context ablation uses a single generator and a 1,000-question subset; the generator and judge are API-based models whose behavior may change over time, and the latency and cost of the LLM call were not measured.

Future work should decouple topic from evidence cardinality when training the router, for example by adding two-document film questions, or predict the evidence budget directly; extend the framework to variable cardinalities, such as those of MuSiQue [40], and to open-domain retrieval, where it could follow an adaptive retrieval step; evaluate on natively authored Indonesian benchmarks; and report several seeds, several generators, and end-to-end latency and cost.

Author Contributions: Conceptualization, I.A.R. and A.W.; methodology, I.A.R.; software, I.A.R.; validation, I.A.R. and A.W.; formal analysis, I.A.R.; investigation, I.A.R.; data curation, I.A.R.; writing-original draft preparation, I.A.R.; writing-review and editing, A.W.; supervision, A.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The source code, experiment configurations, and data-preparation scripts supporting the results of this study, including the scripts that build the training, validation, and evaluation splits and the candidate pools, are openly available at https://github.com/iqbalabd1201/adaptive-indonesian-multihop-rag. The Indonesian-translated data are derived from the publicly available HotpotQA (https://hotpotqa.github.io) and 2WikiMultihopQA (https://github.com/Alab-NII/2wikimultihop) benchmarks; the translation procedure is described in Section 3.2 and implemented in the released data-preparation scripts.

Acknowledgments: The authors acknowledge the use of an AI writing assistant to help prepare and format portions of this manuscript from the authors' underlying research materials and results; the authors have reviewed all AI-assisted content, take full responsibility for its accuracy, and confirm that all reported data and findings originate from their own research.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] Z. Yang *et al.*, "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2369–2380. doi: 10.18653/v1/D18-1259.

[2] X. Ho, A.-K. Duong Nguyen, S. Sugawara, and A. Aizawa, "Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 6609–6625. doi: 10.18653/v1/2020.coling-main.580.

[3] P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *arXiv*. Apr. 12, 2021. [Online]. Available: http://arxiv.org/abs/2005.11401

[4] Y. Du *et al.*, "Context Length Alone Hurts LLM Performance Despite Perfect Retrieval," in *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025, pp. 23281–23298. doi: 10.18653/v1/2025.findings-emnlp.1264.

[5] F. Cuconasu *et al.*, "The Power of Noise: Redefining Retrieval for RAG Systems," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2024, pp. 719–729. doi: 10.1145/3626772.3657834.

[6] N. F. Liu *et al.*, "Lost in the Middle: How Language Models Use Long Contexts," *Trans. Assoc. Comput. Linguist.*, vol. 12, pp. 157–173, Feb. 2024, doi: 10.1162/tacl_a_00638.

[7] NLLB Team *et al.*, "No Language Left Behind: Scaling Human-Centered Machine Translation," *Arxiv preprint*. Aug. 25, 2022. [Online]. Available: http://arxiv.org/abs/2207.04672

[8] R. Nogueira and K. Cho, "Passage Re-ranking with BERT," *Arxiv preprint*. Apr. 14, 2020. [Online]. Available: http://arxiv.org/abs/1901.04085

[9] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. Park, "Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024, pp. 7036–7050. doi: 10.18653/v1/2024.naacl-long.389.

[10] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 10014–10037. doi: 10.18653/v1/2023.acl-long.557.

[11] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," in *arXiv*, Oct. 2023. [Online]. Available: http://arxiv.org/abs/2310.11511

[12] Z. Jiang *et al.*, "Active Retrieval Augmented Generation," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 7969–7992. doi: 10.18653/v1/2023.emnlp-main.495.

[13] W. Su, Y. Tang, Q. Ai, Z. Wu, and Y. Liu, "DRAGIN: Dynamic Retrieval Augmented Generation based on the Real-time Information Needs of Large Language Models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 12991–13013. doi: 10.18653/v1/2024.acl-long.702.

[14] M. Tu, K. Huang, G. Wang, J. Huang, X. He, and B. Zhou, "Select, Answer and Explain: Interpretable Multi-Hop Reading Comprehension over Multiple Documents," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 05, pp. 9073–9080, Apr. 2020, doi: 10.1609/aaai.v34i05.6441.

[15] X.-Y. Li, W.-J. Lei, and Y.-B. Yang, "From Easy to Hard: Two-stage Selector and Reader for Multi-hop Question Answering," *Arxiv preprint*. May 24, 2022. [Online]. Available: http://arxiv.org/abs/2205.11729

[16] J. Zhang, H. Zhang, D. Zhang, L. Yong, and S. Huang, "End-to-End Beam Retrieval for Multi-Hop Question Answering," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024, pp. 1718–1731. doi: 10.18653/v1/2024.naacl-long.96.

[17] C. Taguchi, S. Maekawa, and N. Bhutani, "Efficient Context Selection for Long-Context QA: No Tuning, No Iteration, Just Adaptive-k," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025, pp. 20116–20141. doi: 10.18653/v1/2025.emnlp-main.1017.

[18] J. Sun, X. Zhong, S. Zhou, and J. Han, "DynamicRAG: Leveraging Outputs of Large Language Model as Feedback for Dynamic Reranking in Retrieval-Augmented Generation," *Arxiv preprint*. May 16, 2025. [Online]. Available: http://arxiv.org/abs/2505.07233

[19] D. Lee, Y. Jo, H. Park, and M. Lee, "Shifting from Ranking to Set Selection for Retrieval Augmented Generation," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 17606–17619. doi: 10.18653/v1/2025.acl-long.861.

[20] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to Answer Open-Domain Questions," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1870–1879. doi: 10.18653/v1/P17-1171.

[21] W. Xiong *et al.*, "Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval," *Arxiv preprint*. Feb. 19, 2021. [Online]. Available: http://arxiv.org/abs/2009.12756

[22] K. Chourasiya, P. Saxena, and P. K. Jain, "CAMMRF: A Conflict-Aware Missing-Modality Robust Fusion Framework for Multimodal Sentiment Analysis under Heterogeneous Observability," *J. Comput. Theor. Appl.*, vol. 4, no. 2, pp. 443–456, Sep. 2026, doi: 10.62411/jcta.16538.

[23] S. Min, E. Wallace, S. Singh, M. Gardner, H. Hajishirzi, and L. Zettlemoyer, "Compositional Questions Do Not Necessitate Multi-hop Reasoning," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 4249–4257. doi: 10.18653/v1/P19-1416.

[24] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling, "Corrective Retrieval Augmented Generation," *Arxiv preprint*. Oct. 07, 2024. [Online]. Available: http://arxiv.org/abs/2401.15884

[25] O. Khattab and M. Zaharia, "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*, Jul. 2020, pp. 39–48. doi: 10.1145/3397271.3401075.

[26] W. Sun *et al.*, "Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 14918–14937. doi: 10.18653/v1/2023.emnlp-main.923.

[27] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.

[28] B. A. A. Nazief and M. Adriani, "Confix stripping: Approach to stemming algorithm for Bahasa Indonesia," 1996.

[29] M. Adriani, J. Asian, B. Nazief, S. M. M. Tahaghoghi, and H. E. Williams, "Stemming Indonesian: A confix-stripping approach," *ACM Trans. Asian Lang. Inf. Process.*, vol. 6, no. 4, pp. 1–33, Dec. 2007, doi: 10.1145/1316457.1316459.

[30] F. Z. Tala, "A study of stemming effects on information retrieval in Bahasa Indonesia," University of Amsterdam, 2003. [Online]. Available: https://eprints.illc.uva.nl/id/eprint/740/1/MoL-2003-02.text.pdf

[31] T. Pires, E. Schlinger, and D. Garrette, "How Multilingual is Multilingual BERT?," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 4996–5001. doi: 10.18653/v1/P19-1493.

[32] A. Conneau *et al.*, "Unsupervised Cross-lingual Representation Learning at Scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8440–8451. doi: 10.18653/v1/2020.acl-main.747.

[33] J. Devlin, M.-W. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Oct. 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[34] A. Vaswani *et al.*, "Attention Is All You Need," Aug. 2023, [Online]. Available: http://arxiv.org/abs/1706.03762

[35] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," in *arXiv*, Jul. 2020. [Online]. Available: http://arxiv.org/abs/2005.14165

[36] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," *Arxiv preprint*. Feb. 24, 2020. [Online]. Available: http://arxiv.org/abs/1904.09675

[37] L. Zheng *et al.*, "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," *arXiv*. Dec. 24, 2023. [Online]. Available: http://arxiv.org/abs/2306.05685

[38] R. Dror, G. Baumer, S. Shlomov, and R. Reichart, "The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1383–1392. doi: 10.18653/v1/P18-1128.

[39] R. Geirhos *et al.*, "Shortcut learning in deep neural networks," *Nat. Mach. Intell.*, vol. 2, no. 11, pp. 665–673, Nov. 2020, doi: 10.1038/s42256-020-00257-z.

[40] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "♫ MuSiQue: Multihop Questions via Single-hop Question Composition," *Trans. Assoc. Comput. Linguist.*, vol. 10, pp. 539–554, May 2022, doi: 10.1162/tacl_a_00475.