Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 219-240
DOI: https://doi.org/10.51903/jtie.v5i1.504

# Explainable Multi-Hop Question Answering for QA Assistants: Two-Hop Evidence Retrieval, Sentence-Level Supporting Facts, and Explicit Reasoning Paths

Xiaofei Luo*[1]
Email: xiaofeiluo01@gmail.com
[1]*Information Science, University of Illinois at Urbana-Champaign, IL, US*
*Corresponding Author

**Abstract**
*Multi-hop question answering (QA) for customer-facing assistants requires not only accurate answers but also auditable evidence that explains how the system arrived at each answer. This study investigates, in a controlled and fully reproducible setting, how much explanation quality can be obtained from a deterministic multi-hop QA pipeline that separates retrieval, sentence-level evidence selection, and rule-based answer extraction. Rather than introducing a new learned model, the contribution is an interpretable integration of these modules and an empirical analysis of how each module affects evidence quality. The retriever ranks candidate paragraphs using lexical IDF-weighted token overlap; the evidence selector chooses a small set of high-scoring sentences; and the reasoner extracts an answer using transparent heuristic rules. We evaluate three variants on the complete development splits of HotpotQA (7,405 questions, distractor setting) and 2WikiMultihopQA (12,576 questions). On HotpotQA, sentence-level evidence selection improves Supporting Fact F1 from 0.334 to 0.419, and adding an explicit two-hop retrieval path further increases Supporting Fact F1 to 0.426 and paragraph recall@2 to 0.603, with Answer F1 increasing from 0.084 to 0.088. On 2WikiMultihopQA, evidence selection improves Supporting Fact F1 from 0.328 to 0.429 and Answer F1 from 0.071 to 0.075, while the fixed two-hop path does not provide an additional gain. These results show that most explainability gains come from sentence selection, whereas explicit path constraints help mainly when the task has a strong two-hop structure. The study therefore provides a simple baseline and clearly defines its applicability and limitations for explanation-critical QA assistants.*

**Keywords**: *Multi-Hop Question Answering, Explainable QA, Evidence Retrieval, Supporting Facts, Reasoning Paths.*

## I. INTRODUCTION

Question–answering (QA) assistants are increasingly used as "knowledge customer service agents" that must answer user questions using documents such as help-center articles, wikis, and policy pages. In these settings, users and auditors care about more than the final answer string. They also require supporting evidence: the system must show which facts justify the answer so that a human can verify it, correct it, or trace it back to an authoritative source. This need for traceability is especially acute in customer support, where incorrect answers can violate policy, create financial risk, or erode trust in the product and the organization (Miller, 2019). Consequently, QA assistants are increasingly evaluated not only on answer correctness but also on evidence quality and reasoning transparency.

A large fraction of real user requests are multi-hop questions. Unlike single-hop questions, where one sentence often contains the answer, multi-hop questions require combining information from multiple documents or multiple parts of a document. A system may need to identify an

*Received on February, 2026; Revised on February, 2026; Accepted on March, 2026; Published on April, 2026*

intermediate entity (hop 1) and then use that entity to retrieve another document or fact (hop 2) before it can answer the original query. Examples include: "What nationality is the author of the book that inspired this film?", or "Are these two products made in the same country?". In practice, multi-hop failures often manifest as "answering the wrong thing": the system retrieves a partially relevant passage, extracts a plausible but incorrect span, and produces an answer that is difficult to audit because the evidence trail is missing or incomplete.

## II. LITERATURE REVIEW

Retrieval-augmented QA has become a standard approach for knowledge-intensive NLP (Gario et al., 2026; Hao & Liu, 2025; Sriasih et al., 2025). Early retriever-reader pipelines combined sparse retrieval with neural reading (Chen, Fisch, Weston, & Bordes, 2017), while more recent systems learn dense retrieval representations to improve recall (Karpukhin et al., 2020) and integrate retrieval into pretraining (Guu et al., 2020). Efficient neural retrieval architectures such as ColBERT make dense retrieval practical at scale (Khattab & Zaharia, 2020). Generative QA architectures integrate retrieval with sequence generation, as in retrieval-augmented generation (Lewis et al., 2020), and fusion-in-decoder models aggregate evidence from multiple passages in a single generative step (Izacard & Grave, 2021). These approaches can be accurate but are hard to attribute to a small set of verifiable facts, complicating explanation-critical deployments.

For multi-hop QA, the most operational form of explainability is evidence-based rather than purely post-hoc: the system should retrieve the relevant pages, identify supporting sentences, and expose the evidence chain used for answering. HotpotQA directly evaluates this requirement through sentence-level supporting facts (Yang et al., 2018), and ERASER provides a general rationale-evaluation framework for evidence alignment and faithfulness (DeYoung et al., 2020). More focused multi-hop studies have also emphasized explicit retrieval structure, including reasoning-path retrieval over Wikipedia graphs (Asai et al., 2020), joint paragraph/sentence/answer modeling with hierarchical graphs (Fang et al., 2020), and structured explanations for multi-hop reasoning (Li & Du, 2023). These studies suggest that explanations are most useful when they are grounded in retrievable evidence units rather than free-form rationales.

Datasets for explainable multi-hop QA make this evidence-based requirement measurable. HotpotQA provides multi-hop questions paired with sentence-level supporting facts and multiple Wikipedia paragraphs, enabling evaluation of both answers and evidence selection (Yang et al., 2018). 2WikiMultihopQA extends this idea by designing questions that require reasoning over multiple Wikipedia pages and by providing explicit evidence annotations intended to evaluate reasoning steps (Ho, Sugawara, & Aizawa, 2020). These datasets match the needs of enterprise

QA: they contain questions that require combining multiple sources, and they provide supervision signals for the evidence chain that is critical for trust.

Beyond HotpotQA and 2WikiMultihopQA, several benchmarks highlight complementary aspects of multi-hop evidence use. WikiHop evaluates multi-document multi-hop reading comprehension where evidence must be aggregated across documents (Welbl, Stenetorp, & Riedel, 2018). FEVER frames evidence retrieval as fact extraction and verification, requiring systems to retrieve and cite evidence sentences to support or refute a claim (Thorne et al., 2018). MuSiQue constructs multi-hop questions by composing single-hop questions, creating longer evidence chains and making reasoning depth a controllable factor (Trivedi et al., 2022). Taken together, these resources reinforce a key lesson for QA assistants: reliable answering requires both high evidence recall and the ability to present a compact, faithful rationale.

A second line of research proposes explicit reasoning structure as a mechanism for both accuracy and interpretability. Question decomposition methods rewrite a complex question into a sequence of simpler sub-questions and then combine the intermediate answers (Min et al., 2019). Graph-based models propagate information across entities and documents (De Cao et al., 2019), and hierarchical graph networks jointly optimize paragraph selection, supporting fact extraction, and answer prediction for multi-hop QA (Fang et al., 2020). While these approaches often use learned components, they share a common design principle with our pipeline: separating retrieval, evidence selection, and reasoning clarifies failure modes. This separation is valuable even when deploying large generative models, because evidence selection and path inspection remain necessary for auditing and for reducing unsupported answers.

Recent multi-hop QA systems have also moved toward more explicit retrieval–reasoning interaction, for example through recurrent reasoning-path retrieval over Wikipedia graphs (Asai et al., 2020), multi-hop dense retrieval that conditions later hops on earlier evidence (Xiong et al., 2021), and end-to-end beam retrieval that preserves multiple partial hypotheses across hops (Zhang et al., 2024). These systems are stronger accuracy-oriented baselines, but for production deployments interpretable deterministic baselines remain valuable because they offer predictable latency, exact reproducibility, and auditable intermediate outputs.

Despite progress in learned multi-hop QA, two issues remain important for explanation-critical assistants. First, many strong systems optimize end-task accuracy, but the separate effects of retrieval structure and sentence-level evidence selection on explanation quality are not always isolated. Second, when the reasoning trace is implicit, it becomes difficult to diagnose whether an error originates from retrieval, evidence selection, or answer extraction. The specific gap addressed in this paper is therefore not a missing neural architecture, but the lack of a transparent

baseline that cleanly exposes these stages and quantifies what each stage contributes to evidence quality.

This study investigates whether a deterministic Retriever → Evidence Selector → Reasoner pipeline can serve as such a baseline. The objectives are to determine: (1) how much sentence-level evidence selection improves supporting-fact quality over retrieval-only explanations; (2) whether an explicit ordered two-hop path improves paragraph coverage and supporting-fact quality; and (3) under which data conditions these gains hold or weaken. Accordingly, the contribution of the paper is an interpretable integration and controlled empirical analysis, rather than a claim of algorithmic novelty in the individual components. Concretely, we contribute (1) a modular, fully inspectable pipeline that outputs answers, supporting facts, and ordered two-hop paths; (2) controlled evaluations on the full development splits of HotpotQA and 2WikiMultihopQA; and (3) ablation and scenario analyses that show where evidence selection is consistently helpful and where strict two-hop retrieval is beneficial or limiting.

## III. RESEARCH METHOD

This section describes the task formulation, datasets, the proposed modular pipeline, and the evaluation protocol. To keep the comparison fair, all system variants use the same preprocessing, lexical scoring backbone, paragraph budget (top-2), and rule-based reasoner unless the tested module explicitly changes that stage; thus, differences among R, R+ES, and R+ES+Path can be attributed to the added evidence-selection or path module rather than to separate retuning. All experiments are conducted on the full development splits of the specified datasets, without subsampling. Because the goal is explainability and traceability, every system variant outputs (i) an answer string, (ii) a set of sentence-level supporting facts, and (iii) an explicit two-hop evidence path when the path module is enabled. As a simple robustness check, we also report results across two datasets and across scenario slices defined by hop proxy, paragraph coverage, and answer type.

### *A. Task definition*

Each example consists of a question $q$and a small candidate context $\mathcal{C} = \{P_i\}_{i=1}^{m}$, where each paragraph $P_i$is represented as $\left(\text{title}_i, [s_{i,1}, \ldots, s_{i,n_i}]\right)$. The system must produce an answer $\hat{a}$and an evidence set $\hat{\mathcal{S}}$consisting of sentence identifiers $(\text{title}_i, j)$. For datasets that provide supporting facts $\mathcal{S}^*$, we treat $\mathcal{S}^*$as the gold evidence set and evaluate how well $\hat{\mathcal{S}}$matches $\mathcal{S}^*$. In addition, for path-based variants we output an ordered pair of paragraph titles $(p_1 \rightarrow p_2)$that constitutes an explicit two-hop evidence path. We make four scope assumptions explicit. First, the provided candidate context contains most of the evidence needed for answering, so the study evaluates

*Journal of Technology Informatics and Engineering (JTIE)*
**Vol. 5 No. 1 April 2026**
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 219-240
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

controlled multi-hop selection rather than open-web retrieval. Second, lexical overlap is treated as a reasonable first-stage retrieval signal, favoring settings in which key entities or titles are surface-visible. Third, the explicit path module assumes that many questions can be approximated by one dominant two-hop chain. Fourth, the rule-based reasoner is best suited to date, number, restricted yes/no comparison, and short span-like answers. When these assumptions do not hold, especially in variable-hop or paraphrase-heavy cases, performance should be expected to degrade.

### *B. Datasets*

We evaluate on HotpotQA in the distractor setting and on 2WikiMultihopQA. HotpotQA was designed to require reasoning over two Wikipedia articles and includes sentence-level supporting facts (Yang et al., 2018). 2WikiMultihopQA similarly emphasizes multi-hop reasoning and provides evidence annotations intended to assess reasoning steps (Ho et al., 2020). Table 1 summarizes key properties of the two development splits used in this study, including the number of questions, the size of the candidate context, the average number of labeled supporting sentences, and the distribution of gold supporting-fact counts.

**Table 1. Dataset Statistics for the Evaluated Development Splits**

| Dataset (split) | N | Avg ctx paras | Avg ctx sents | Avg gold SF sents | Avg gold titles | Yes/No answers | SF dist (2/3/4/5+)% |
|---|---|---|---|---|---|---|---|
| HotpotQA (dev distractor) | 7405 | 9.953 | 41.389 | 2.431 | 2.000 | 6.185% | 67.4/24.0/7.3/1.4 |
| 2WikiMultihopQA (dev) | 12576 | 10.000 | 31.878 | 2.440 | 2.438 | 10.297% | 78.0/0.2/21.8/0.1 |

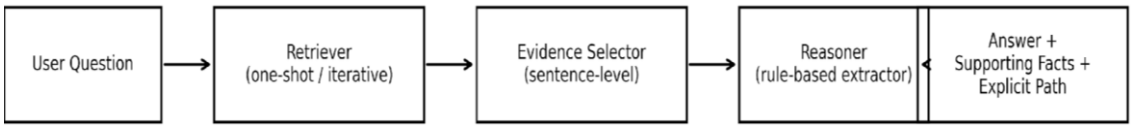



**Figure 1. Modular Architecture (Retriever → Evidence Selector → Reasoner) and the Produced Explanation Artifacts**

### *C. Explainability output*

In a deployed assistant, explanations are most useful when they are both concise and faithful. We therefore output two complementary forms of explanation: (1) a compact set of supporting sentences (supporting facts) that can be rendered as citations in a user interface, and (2) an explicit ordered path over paragraphs (hop 1 → hop 2) that exposes the intermediate evidence chain. Figure 1 illustrates the overall architecture and the produced artifacts.

### *D. Overview of system variants*

We evaluate three system configurations that progressively add explainability-oriented modules while holding the rest of the pipeline fixed. The first configuration (R) uses only paragraph

retrieval and produces an answer by combining all sentences from the top-retrieved paragraphs. The second (R+ES) retains the same retriever and reasoner but adds sentence-level evidence selection, producing a small set of supporting facts. The third (R+ES+Path) retains the same evidence selector and reasoner and changes only the retrieval stage by adding an explicit two-hop path via iterative retrieval with query expansion. This design makes the comparisons module-specific: R→R+ES isolates the effect of sentence selection, while R+ES→R+ES+Path isolates the effect of ordered two-hop retrieval. Table 2 summarizes the differences among these systems.

**Table 2. System Variants and Their Components**

| System | Retriever | Evidence Selector | Reasoner | Explicit path output |
|---|---|---|---|---|
| R (One-shot retrieval) | LexIDF overlap, top-2 paragraphs | None (use all sentences in top-2) | Weighted phrase extraction (rule-based) | Implicit (top-2 ranked set) |
| R+ES (Sentence selection) | LexIDF overlap, top-2 paragraphs | Top-4 sentences, ≤2 per paragraph | Weighted phrase extraction (rule-based) | Implicit (top-2 ranked set) |
| R+ES+Path (2-hop iterative) | Hop1 by LexIDF; Hop2 by query expansion with hop1 title | Top-4 sentences, ≤2 per paragraph | Weighted phrase extraction (rule-based) | Explicit ordered (hop1 → hop2) |

### *E. LexIDF Retriever (one-shot)*

The retriever ranks candidate paragraphs using lexical IDF-weighted token overlap, which is a transparent approximation to classical sparse retrieval methods such as BM25 (Robertson & Zaragoza, 2009). We tokenize text with a simple regex tokenizer and remove stopwords. For each dataset, we compute the document frequency df(t) across all candidate paragraphs in the development split and define the idf(t) = log((N+1)/(df(t)+1)) + 1, where N is the number of paragraphs. For a question q, the score of paragraph P is the sum of idf weights of query tokens that appear in P, plus an additional title-overlap bonus weighted by $\alpha = 1.5$. The one-shot retriever selects the top-2 paragraphs.

### *F. Two-hop path retriever (iterative)*

Multi-hop questions frequently benefit from retrieving an intermediate paragraph before retrieving a second paragraph that contains the final answer. To explicitly expose this structure, the R+ES+Path variant performs two sequential retrieval steps. Hop 1 selects the top paragraph under the one-shot scoring function. Hop 2 expands the query with hop 1's title tokens (q ∪ title(hop1)) and then selects the best remaining paragraph under the same LexIDF scoring. The resulting ordered titles define an explicit path hop1 → hop2. This design is deterministic and directly inspectable: auditors can see which intermediate title influenced hop 2 retrieval.

### *G. Evidence Selector (sentence-level)*

*Journal of Technology Informatics and Engineering (JTIE)*
**Vol. 5 No. 1 April 2026**
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 219-240
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

The Evidence Selector reduces the retrieved content to a compact set of supporting facts. Given the two retrieved paragraphs, we score each sentence by IDF-weighted overlap with an expanded query consisting of the question tokens unioned with the retrieved title tokens. We then select the top 4 sentences across both paragraphs, with a maximum of 2 sentences per paragraph. This yields a rationale that is short enough to display and evaluate, while still allowing two sentences per hop for cases where the supporting evidence is split across sentences.

**Table 3. Hyperparameters and Implementation Details Used in All Experiments**

| Category | Parameter | Value |
|---|---|---|
| Text processing | Tokenizer | Regex [A-Za-z0-9]+, lowercase |
| Text processing | Stopword list | scikit-learn ENGLISH_STOP_WORDS |
| Retriever | IDF | idf(t)=log((N+1)/(df(t)+1))+1 |
| Retriever | Title weight α | 1.5 |
| Retriever | Top-K paragraphs | 2 |
| Path retriever | Hop2 query expansion | question tokens ∪ hop1 title tokens |
| Evidence selector | Top-K sentences | 4 |
| Evidence selector | Max sentences per paragraph | 2 |
| Reasoner | Answer extraction | Weighted candidate phrase scoring by sentence relevance |
| Reasoner | Special cases | When→year/date; How many→number; same-country yes/no→country tail compare |
| Compute | Runtime (sec/example) HotpotQA | R=0.0008, R+ES=0.0006, R+ES+Path=0.0007 |
| Compute | Runtime (sec/example) 2Wiki | R=0.0006, R+ES=0.0005, R+ES+Path=0.0006 |

### *H. Reasoner (weighted phrase extractor)*

The Reasoner converts selected evidence sentences into an answer. For “When” questions, the reasoner returns the first year/date-like pattern found in the evidence. For “How many” questions, it returns the first number. For a restricted class of yes/no questions that ask whether two entities share the same country or nationality, it extracts country-like tail phrases from the first sentence of each retrieved paragraph and returns “yes” if they match and “no” otherwise. For all other questions, it extracts candidate proper-noun phrases from evidence sentences and scores each candidate by the relevance score of the sentence in which it appears (IDF overlap with the question). The predicted answer is the candidate with the highest total score. This scoring is transparent: for any prediction, the system can report which evidence sentences contributed to the winning candidate.

### *I. Supporting-fact prediction sets*

The three system variants differ in the size of the predicted supporting-fact set $\hat{S}$. In R (retrieval-only), we treat every sentence in the two retrieved paragraphs as a supporting fact. This design

maximizes recall but often produces long rationales, which is undesirable in a QA assistant interface. In R+ES and R+ES+Path, we cap $\hat{S}$ to the top-4 selected sentences (with at most two per paragraph), matching the goal of producing a concise explanation. Because supporting-fact evaluation is set-based, precision and recall respond directly to this cap: a smaller $\hat{S}$ typically increases precision but may reduce recall. Reporting precision and recall alongside F1 (Tables 4–5) therefore provides a more complete view of explanation quality than F1 alone.

### *J. Computational complexity*

For each question, the LexIDF retriever scores all candidate paragraphs in the provided context, a process that requires token set construction and overlap computation. Given m paragraphs and n total sentences, scoring runs in O(m + n) time with the cached token sets used in our implementation. Sentence selection then sorts sentence scores within the retrieved paragraphs, which is O(n log n) in the worst case but small in practice because only two paragraphs are scored for selection. The overall runtime, as shown in Table 3, is under a millisecond per example across all variants, making the approach suitable as a low-latency evidence layer for interactive QA assistants.

### *K. Evaluation metrics*

We report Answer Exact Match (EM) and token-level F1 scores using the standard normalization protocol for extractive QA benchmarks (Rajpurkar et al., 2016; Yang et al., 2018). For supporting facts, we compute precision, recall, and F1 between predicted and gold (title, sentence-index) pairs, as well as Supporting Fact EM (whether the predicted set exactly matches the gold set). We also report paragraph recall@2, defined as the fraction of distinct gold evidence titles that appear in the top-2 retrieved titles. Paragraph recall@2 measures whether the retriever recovers the correct evidence pages, while Supporting Fact F1 measures sentence-level explanation quality.

### *L. Hop bucketing*

To analyze how performance varies with reasoning depth, we compute a hop proxy based on the number of gold supporting-fact sentences $|S^*|$. We assign an example to the 2-hop bucket if $|S^*| \leq 2$ and to the 3-hop bucket if $|S^*| \geq 3$. The datasets' annotation structure supports this proxy: questions requiring more reasoning steps typically annotate more supporting sentences. We use this bucketing consistently for both datasets and report bucketed metrics in Table 7 and Figure 4 as a simple robustness check across question complexity. Because it is only a proxy, we interpret these results as scenario-based stress tests rather than exact counts of logical inference steps.

### *M. Implementation details and reproducibility*

*Journal of Technology Informatics and Engineering (JTIE)*
**Vol. 5 No. 1 April 2026**
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 219-240
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

Table 3 lists all hyperparameters and preprocessing choices. The values were fixed once and used consistently across both datasets: top-K paragraphs=2 matches the intended two-page evidence budget of the pipeline; top-K sentences=4 with at most two per paragraph provides a compact citation set while still allowing both hops to contribute more than one sentence; and title weight $\alpha$=1.5 is used as a conservative bonus so that title overlap can help retrieval without overwhelming body-text overlap.

No variant receives dataset-specific tuning beyond computing IDF statistics from the respective development context. The pipeline is deterministic and requires no training, so there are no random initializations, optimization seeds, or stochastic decoding steps; repeated runs on the same split produce identical outputs. All reported results are computed by running the full development sets end-to-end with the same tokenization, stopword list, and evaluation script, and runtime is reported as seconds per example on the evaluation machine, illustrating that the system is suitable for latency-sensitive QA assistant deployments.

## IV. RESULT

This section reports empirical results and analyzes how each module affects answer accuracy and explanation quality. The presentation follows the three objectives stated in the Introduction: first, we measure the effect of sentence-level evidence selection over retrieval-only explanations; second, we test whether an explicit ordered two-hop path improves retrieval coverage and supporting-fact quality; third, we examine where these effects are stable or unstable through scenario analyses by hop proxy, paragraph coverage, and answer type. Because the pipeline is deterministic and uses only the provided candidate context, the reported numbers directly reflect the behavior of the retriever, evidence selector, and reasoner without any learned parameters.

**Table 4. HotpotQA (Dev) Results for Answer Accuracy and Supporting-Fact Explainability**

| System | Answer EM | Answer F1 | SF F1 | SF Precision | SF Recall | Para Recall@2 |
|---|---|---|---|---|---|---|
| R (One-shot retrieval) | 0.033 | 0.084 | 0.334 | 0.244 | 0.602 | 0.595 |
| R+ES (Sentence selection) | 0.033 | 0.087 | 0.419 | 0.351 | 0.544 | 0.595 |
| R+ES+Path (2-hop iterative) | 0.034 | 0.088 | 0.426 | 0.357 | 0.552 | 0.603 |

### *A. Main results on HotpotQA*

Table 4 reports Answer EM/F1, Supporting Fact metrics, and paragraph recall@2 for the three system variants on the full HotpotQA development split (N=7,405). The one-shot retriever baseline (R) achieves Answer F1=0.084 and Supporting Fact F1=0.334. Adding sentence-level evidence selection (R+ES) improves Supporting Fact F1 to 0.419, indicating that selecting a small

set of relevant sentences substantially increases the faithfulness and compactness of explanations. The full system with an explicit two-hop path (R+ES+Path) further improves Supporting Fact F1 to 0.426 and slightly increases Answer F1 to 0.088. Figure 2 visualizes the trends of Answer F1 and Supporting Fact F1 across the three systems.

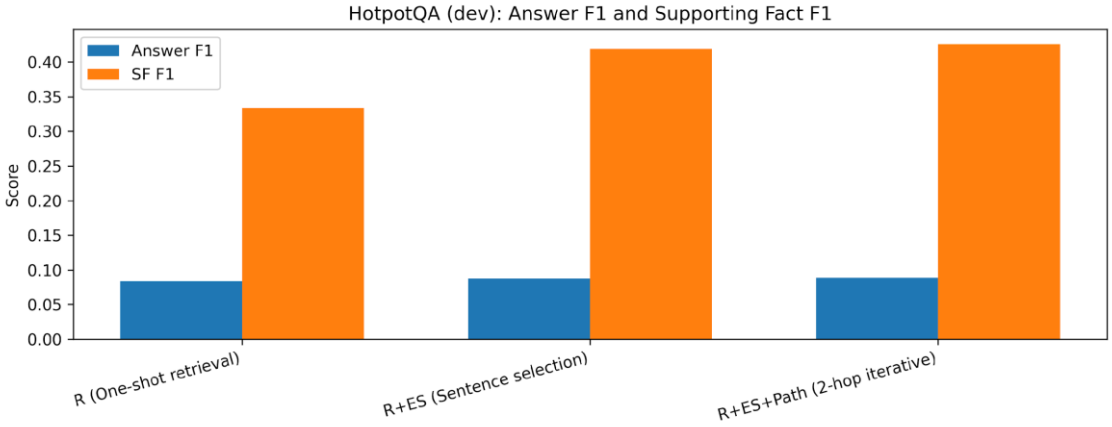



**Figure 2. HotpotQA (dev): Answer F1 and Supporting Fact F1 Across System Variants**

### *B. Interpretation of HotpotQA results*

HotpotQA's supporting-fact supervision directly rewards systems that identify the correct sentences, even when the final answer span is difficult to extract. The large gain in Supporting Fact F1 from R to R+ES and R+ES+Path arises mainly because the baseline R outputs every sentence in the top-2 paragraphs, which inflates recall but introduces many irrelevant sentences; once the selector restricts the explanation to the top-4 sentences, precision increases substantially with only a moderate recall drop. The additional gain from the explicit path module is smaller but consistent on HotpotQA because many questions are organized around two key Wikipedia pages and the second page is often easier to retrieve after conditioning on the first-hop title. In other words, the path module helps most when HotpotQA's intended two-page structure aligns with the ordered hop1 → hop2 retrieval assumption. The comparatively small Answer F1 gains indicate that evidence quality improves faster than answer extraction, which is expected because the rule-based reasoner remains the main bottleneck after the correct evidence is found.

### *C. Supporting-fact precision–recall trade-off*

The Supporting Fact metrics in Tables 4–5 reveal a consistent pattern. On HotpotQA, retrieval-only explanations (R) achieve Supporting Fact recall=0.602 but low precision=0.244, because the system outputs every sentence in the retrieved paragraphs. Adding sentence selection (R+ES) increases precision to 0.351 while keeping recall relatively stable at 0.544, producing a substantial F1 gain. With the path module (R+ES+Path), precision further increases to 0.357 and recall to

*Journal of Technology Informatics and Engineering (JTIE)*
**Vol. 5 No. 1 April 2026**
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 219-240
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

0.552. This behavior aligns with the intended use case: a QA assistant benefits more from a short list of high-quality citations than from a long list containing many irrelevant sentences.

**Table 5. 2WikiMultihopQA (dev) Results for Answer Accuracy and Supporting-Fact Explainability**

| System | Answer EM | Answer F1 | SF F1 | SF Precision | SF Recall | Para Recall@2 |
|---|---|---|---|---|---|---|
| R (One-shot retrieval) | 0.038 | 0.071 | 0.328 | 0.252 | 0.643 | 0.643 |
| R+ES (Sentence selection) | 0.038 | 0.075 | 0.429 | 0.368 | 0.551 | 0.643 |
| R+ES+Path (2-hop iterative) | 0.038 | 0.075 | 0.428 | 0.368 | 0.550 | 0.642 |

### *D. Main results on 2WikiMultihopQA*

Table 5 reports the same metrics on the full 2WikiMultihopQA development split (N=12,576). The baseline R system obtains Answer F1=0.071 and Supporting Fact F1=0.328. Adding sentence selection (R+ES) increases Supporting Fact F1 to 0.429 and Answer F1 to 0.075. The explicit two-hop path variant (R+ES+Path) yields a Supporting Fact F1 of 0.428, which is slightly lower than that of R+ES. Figure 3 visualizes the Answer F1 and Supporting Fact F1 across the three variants.

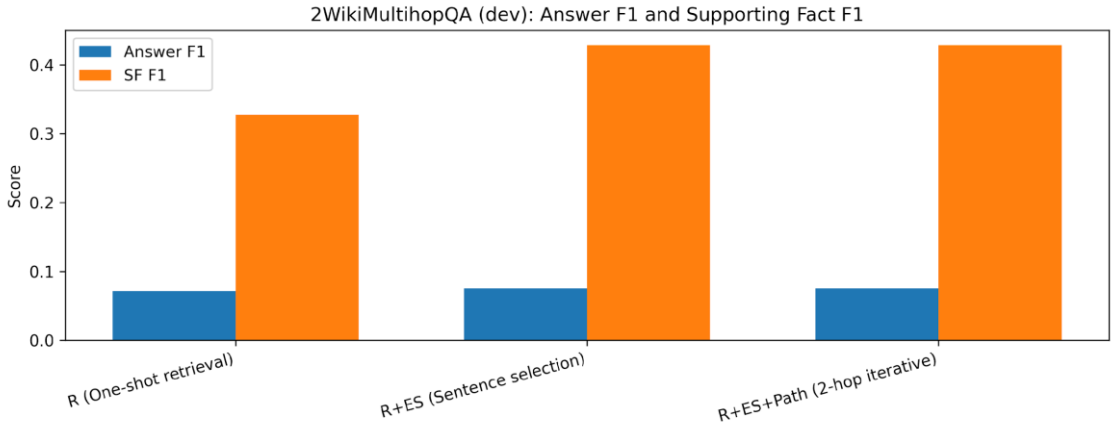



**Figure 3. 2WikiMultihopQA (dev): Answer F1 and Supporting Fact F1 Across System Variants**

### *E. Interpretation of 2WikiMultihopQA results*

Compared with HotpotQA, the 2Wiki setting in our evaluation split contains a larger subset of questions whose gold evidence spans more than two supporting sentences and, on average, more than two titles (Table 1). Under these conditions, the sentence selector still helps by removing many non-supporting sentences from the retrieved paragraphs and concentrating the explanation budget on the most question-relevant evidence. However, the explicit two-hop path does not provide an additional gain because conditioning hop 2 only on hop 1's title can be too restrictive when the evidence structure is less cleanly two-step or when multiple intermediate entities are

useful. This explains why R+ES and R+ES+Path perform almost identically on 2WikiMultihopQA: the benefit of explicit ordering is offset by the loss of flexibility. The result also clarifies a practical boundary of applicability—strict ordered paths are better suited to domains with stable two-step evidence chains than to domains with variable-hop or broader evidence requirements.

### *F. 2Wiki supporting-fact behavior*

The same trade-off appears in 2WikiMultihopQA, but with a stronger precision jump and a larger recall drop under selection. R outputs many sentences, yielding recall=0.643 and precision=0.252. R+ES sharply increases precision to 0.368 while reducing recall to 0.551, which still increases F1 from 0.328 to 0.429. The recall drop reflects the fixed top-4 budget interacting with questions whose gold evidence contains more than four labeled supporting sentences. This suggests that in domains with long evidence chains, a production assistant may either increase the citation budget or present a hierarchical explanation that groups evidence by hop.

### *G. Supporting Fact EM as a strict criterion*

In addition to Supporting Fact F1, we also compute Supporting Fact EM, which requires the predicted supporting-fact set $\hat{S}$ to exactly match the gold set $S^*$. This metric is intentionally strict and therefore low across all variants: on HotpotQA, it increases from 0.013 (R) to 0.026 (R+ES+Path), and on 2WikiMultihopQA, it reaches 0.010 (R+ES). Two factors explain the gap between F1 and EM. First, many examples contain more labeled supporting sentences than the system's top-4 citation budget, which makes perfect set matching impossible even when the retrieved evidence is correct. Second, minor sentence-index mismatches (e.g., selecting an adjacent sentence in the same paragraph) are penalized as completely wrong under EM. For QA assistants, F1 and precision/recall are typically more informative because they capture partial credit and align better with the user experience of seeing a few correct citations.

### *H. Using explainable baselines as diagnostic tools*

Although the proposed system does not aim to maximize answer accuracy, its modular outputs make error sources measurable. When paragraph recall@2 is low, the failure is attributable to retrieval and can be addressed with improved indexing, synonym expansion, or hybrid dense retrieval. When paragraph recall@2 is high, but Answer F1 remains low, the failure is attributable to the reader/reasoner and motivates adding a learned extractive model or constrained generation over the selected sentences. This diagnostic separation is valuable for enterprise teams: it supports targeted iteration. It enables enforcement of policies such as "answers must be supported by at

*Journal of Technology Informatics and Engineering (JTIE)*
**Vol. 5 No. 1 April 2026**
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 219-240
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

least one selected supporting sentence," thereby reducing unsupported generations in downstream models.

**Table 6. Module Ablations: Gains (Δ) Relative to One-Shot Retrieval (R)**

| Dataset | System | Δ Answer F1 | Δ SF F1 | Δ ParaRecall@2 |
|---|---|---|---|---|
| HotpotQA (dev) | R (One-shot retrieval) | 0.000 | 0.000 | 0.000 |
| HotpotQA (dev) | R+ES (Sentence selection) | 0.003 | 0.086 | 0.000 |
| HotpotQA (dev) | R+ES+Path (2-hop iterative) | 0.004 | 0.092 | 0.008 |
| 2WikiMultihopQA (dev) | R (One-shot retrieval) | 0.000 | 0.000 | 0.000 |
| 2WikiMultihopQA (dev) | R+ES (Sentence selection) | 0.004 | 0.101 | 0.000 |
| 2WikiMultihopQA (dev) | R+ES+Path (2-hop iterative) | 0.004 | 0.101 | -0.001 |

### *I. Ablation analysis*

Table 6 reports the incremental effect of adding modules relative to one-shot retrieval (R), and Figure 4 summarizes the Answer F1 gains. On HotpotQA, adding evidence selection increases Answer F1 by 0.003, and adding the path module increases Answer F1 by 0.004 relative to R. The larger effect is on Supporting Fact F1: the full system increases it by 0.092 compared with R. On 2WikiMultihopQA, the dominant improvement comes from evidence selection, which increases Supporting Fact F1 by 0.101 compared with R; the path module does not improve Answer F1 on this split. These ablations confirm that the Evidence Selector is the primary driver of explanation quality. At the same time, the Path module improves retrieval structure and paragraph coverage in datasets with a stronger two-hop bias.

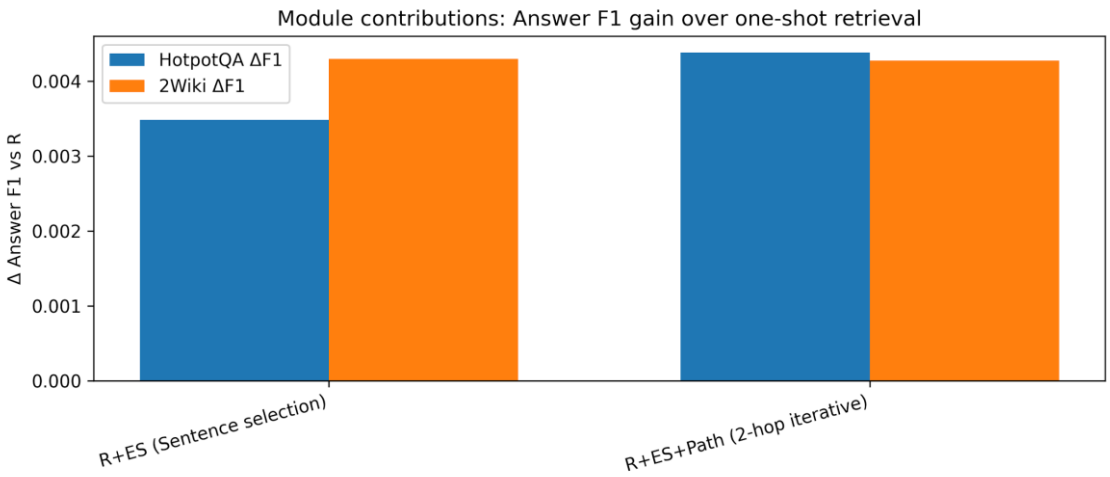



**Figure 4. Module Contributions: Answer F1 Gains Over One-Shot Retrieval**

### *J. Hop-bucket performance*

Table 7 and Figure 5 analyze performance as a function of the hop proxy |S*| (the number of gold supporting-fact sentences). For both datasets, Answer F1 decreases as |S*| increases, which is consistent with the intuition that questions requiring more reasoning steps or more evidence sentences are harder for a lightweight rule-based reasoner. In contrast, Supporting Fact F1

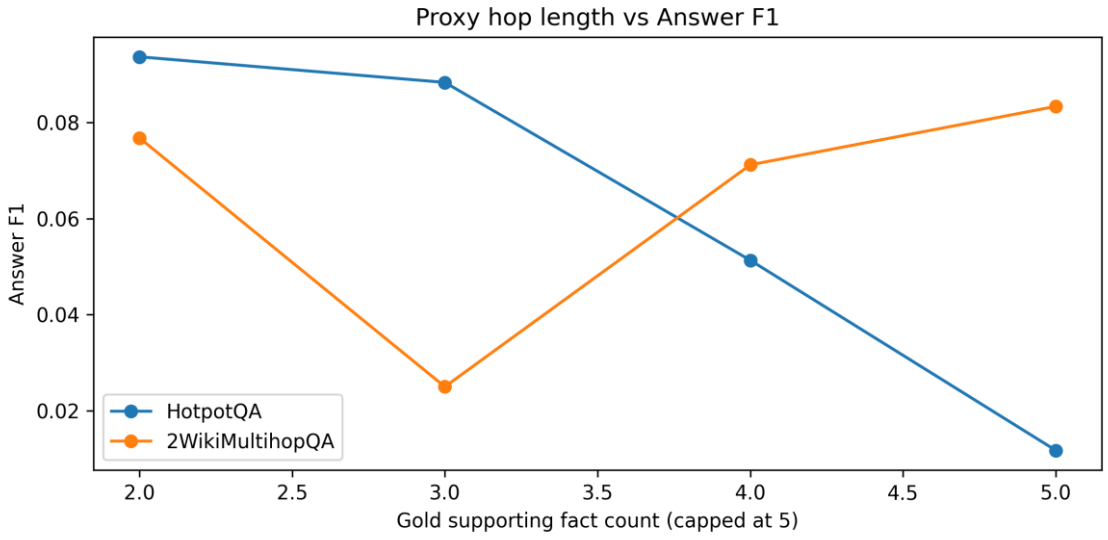
increases with |S*| in HotpotQA because retrieving and selecting more evidence sentences partially offsets the increased difficulty: when more sentences are labeled as supporting, the selector can recover a higher fraction of them even with a fixed top-4 budget. The hop-bucket trend in 2Wiki differs: Supporting Fact F1 remains relatively stable across the buckets, indicating that the selector maintains evidence quality even as questions require longer evidence chains.

**Table 7. Hop-bucket Analysis Using |S*| (Gold Supporting-Fact Sentence Count) as a Hop Proxy**

| Dataset | System | Hop bucket | n | Answer EM | Answer F1 | SF F1 |
|---|---|---|---|---|---|---|
| HotpotQA (dev) | R+ES+Path (2-hop iterative) | 2-hop (\|SF\|<=2) | 4990 | 0.037 | 0.094 | 0.400 |
| HotpotQA (dev) | R+ES+Path (2-hop iterative) | 3-hop (\|SF\|>=3) | 2415 | 0.027 | 0.077 | 0.479 |
| 2WikiMultihopQA (dev) | R+ES (Sentence selection) | 2-hop (\|SF\|<=2) | 9805 | 0.038 | 0.077 | 0.422 |
| 2WikiMultihopQA (dev) | R+ES (Sentence selection) | 3-hop (\|SF\|>=3) | 2771 | 0.038 | 0.071 | 0.451 |

### *K. Numerical hop-bucket trends*

The hop proxy highlights different stress cases for the pipeline. In HotpotQA, the 2-hop bucket (|S*|≤2, n=4990) yields Answer F1=0.094 and Supporting Fact F1=0.400, whereas the 3-hop bucket (|S*|≥3, n=2415) yields a lower Answer F1=0.077 but a higher Supporting Fact F1=0.479. This combination indicates that longer evidence chains challenge the deterministic span extraction logic more than they challenge evidence selection: the selector still identifies many of the labeled sentences, but the reasoner fails to reliably map those sentences to the exact answer string. In 2WikiMultihopQA, the degradation in Answer F1 from 0.077 (2-hop) to 0.071 (3-hop proxy) is more pronounced, reflecting the larger subset of examples whose evidence spans four gold titles or more than four labeled sentences.



*Journal of Technology Informatics and Engineering (JTIE)*
**Vol. 5 No. 1 April 2026**
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 219-240
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

**Figure 5. Proxy Hop Length (Gold Supporting-Fact Count) vs Answer F1**

### *L. One-shot versus iterative retrieval*

The path module changes how the second paragraph is selected: rather than taking the global top-2 by a single scoring function, it forces an ordered hop structure by conditioning hop 2 on hop 1's title. In HotpotQA, this constraint increases paragraph recall@2 from 0.595 (R) to 0.603 (R+ES+Path), which is consistent with the dataset's two-page construction. In contrast, on 2WikiMultihopQA, the same constraint slightly reduces paragraph recall@2 compared to one-shot retrieval. Unlike learned reasoning-path or beam-based retrievers that can maintain multiple path hypotheses and adapt to more variable hop structures (Asai et al., 2020; Zhang et al., 2024), our module keeps a single ordered chain for transparency. This result suggests that explicit single-path constraints are best used when the domain exhibits consistent two-step information needs (e.g., "entity → attribute" chains) and that more flexible path mechanisms are preferable in domains where questions may require multiple intermediate pages.

**Table 8. Error Analysis by Paragraph recall@2 Coverage (para_hit)**

| Dataset | para_hit | n | EM | F1 | SF_F1 | ParaRecall |
|---|---|---|---|---|---|---|
| HotpotQA (dev) | 0 (missing ≥1 gold title in top-2) | 5126 | 0.023 | 0.069 | 0.297 | 0.427 |
| HotpotQA (dev) | 1 (all gold titles in top-2) | 2279 | 0.060 | 0.132 | 0.716 | 1.000 |
| 2WikiMultihopQA (dev) | 0 (missing ≥1 gold title in top-2) | 8666 | 0.026 | 0.060 | 0.343 | 0.482 |
| 2WikiMultihopQA (dev) | 1 (all gold titles in top-2) | 3910 | 0.065 | 0.111 | 0.617 | 1.000 |

### *M. Paragraph recall as a bottleneck*

Table 8 groups results by whether all gold evidence titles are present in the top-2 retrieved paragraphs (para_hit=1) or whether at least one gold title is missing (para_hit=0). Across datasets, the gap is large: when the retriever covers all gold titles, both Answer F1 and Supporting Fact F1 are substantially higher. This observation aligns with the retriever–reader literature: multi-hop QA pipelines often fail primarily due to retrieval errors rather than reader errors (Chen et al., 2017; Karpukhin et al., 2020). In customer service, this implies that improving evidence recall (e.g., via better indexing or hybrid sparse+dense retrieval) can have a direct impact on both correctness and explainability.

### *N. Retrieval coverage and downstream quality*

Table 8 quantifies how strongly paragraph recall@2 governs downstream performance. On HotpotQA, when the retriever covers all gold titles (para_hit=1), Answer F1 rises to 0.132 and Supporting Fact F1 to 0.716; when at least one gold title is missing (para_hit=0), Answer F1 drops to 0.069 and Supporting Fact F1 to 0.297. On 2WikiMultihopQA, the gap is even larger:

para_hit=1 yields Answer F1=0.111 and Supporting Fact F1=0.617, whereas para_hit=0 yields Answer F1=0.060 and Supporting Fact F1=0.343. These differences show that evidence recall is the primary bottleneck for both answer correctness and explanation faithfulness in multi-hop settings.

**Table 9. Error Analysis by Answer Type (Yes/No vs Span/Other)**

| Dataset | answer_is_yesno | n | EM | F1 | SF_F1 | ParaRecall |
|---|---|---|---|---|---|---|
| HotpotQA (dev) | Span/other | 6947 | 0.032 | 0.090 | 0.419 | 0.594 |
| HotpotQA (dev) | Yes/No | 458 | 0.068 | 0.068 | 0.531 | 0.747 |
| 2WikiMultihopQA (dev) | Span/other | 11281 | 0.001 | 0.043 | 0.416 | 0.634 |
| 2WikiMultihopQA (dev) | Yes/No | 1295 | 0.362 | 0.362 | 0.536 | 0.725 |

*O. Answer-type analysis*

Table 9 separates performance for yes/no questions versus span/other answers. On 2WikiMultihopQA, yes/no questions constitute approximately 10.3% of the development set (Table 1), and the rule-based reasoner performs much better on this subset because its constrained yes/no heuristic directly targets a common comparison pattern (same-country or same-nationality questions). For span answers, Answer EM is near zero in this deterministic baseline, which highlights a key limitation: high-quality span extraction typically requires a learned reader or a more sophisticated parsing model. Nevertheless, Supporting Fact metrics remain meaningful across answer types because evidence selection is independent of the final span choice.

*P. Yes/no versus span answers*

Table 9 shows that answer-type sensitivity is substantial. On 2WikiMultihopQA, yes/no questions (n=1295) achieve Answer F1=0.362 because the reasoner includes an explicit comparison heuristic for same-country and same-nationality patterns. Span/other answers (n=11281) achieve Answer F1=0.043, which reflects the difficulty of extracting precise entities or phrases with rules alone. On HotpotQA, the yes/no subset is much smaller, and the gap is correspondingly narrower. These results indicate that in customer service settings, deterministic rules can be highly effective for a narrow class of structured comparison questions, while open-ended span extraction requires either a learned reader or a carefully engineered domain schema.

*Journal of Technology Informatics and Engineering (JTIE)*
**Vol. 5 No. 1 April 2026**
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 219-240
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

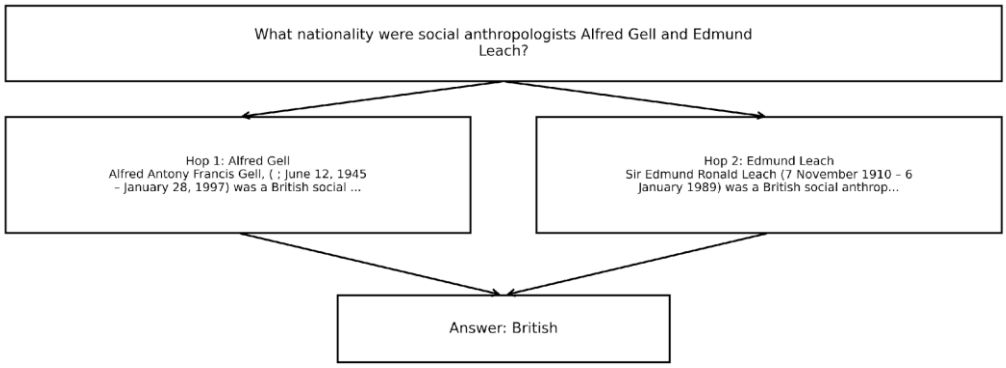



**Figure 6. Qualitative Example: Explicit Two-Hop Path and Selected Supporting Sentences (HotpotQA)**

Figure 6 illustrates a representative successful example from HotpotQA. The system retrieves two paragraphs, selects a small set of supporting sentences, and produces the answer "British," while exposing the ordered hop1 → hop2 path. This style of explanation maps well to QA-assistant user interfaces: the assistant can show the selected sentences as citations, and the explicit path can be surfaced as a "Reasoning steps" panel that lists the intermediate entities or pages used to connect the evidence.

Implications for QA assistants and knowledge customer service. The experimental results show that sentence-level evidence selection provides a substantial improvement in supporting-fact F1 on both datasets. In deployed assistants, this translates into shorter, higher-precision citations that users can scan quickly. Explicit two-hop paths provide additional value by making the retrieval process auditable: a reviewer can see whether hop 2 was retrieved because of the question itself or because of an intermediate title discovered at hop 1. This property can support compliance workflows where the system must justify why a particular internal policy page was consulted.

*Q. Faithfulness and user experience*

Evidence-based explanations are most valuable when they are faithful (every cited sentence is relevant) and sufficient (the citations collectively justify the answer). The Evidence Selector directly optimizes faithfulness by increasing precision, while the Path module improves sufficiency by increasing the probability that the retrieved paragraphs contain the missing intermediate link. In user interfaces, this suggests a practical design: show a small citation set by default, but allow the user to expand the explanation to view the full hop-level path and additional sentences when needed. This design is compatible with model-agnostic explanation frameworks (Ribeiro et al., 2016; Lundberg & Lee, 2017) but focuses on a more actionable artifact for QA: natural-language sentences that can be read directly.

Deployment guidance for knowledge customer service. The modular outputs of the proposed system align with practical UI and policy requirements. First, the assistant should always present the selected supporting sentences as citations, because they provide direct justifications that a user can read. Second, the assistant can present the explicit hop1 → hop2 path as a transparent "why this source?" trace, improving trust and enabling support agents to debug retrieval errors. Third, the assistant can implement conservative abstention: if paragraph recall@2 is low (e.g., no high-confidence paragraph match), the system should explicitly indicate insufficient evidence rather than generate an unsupported answer. Finally, because the pipeline is deterministic, it can be used as a policy-compliant evidence layer that constrains more powerful generators to remain grounded in retrieved sentences.

**Discussion**

The proposed pipeline is intentionally simple and does not compete with stronger learned multi-hop QA systems. Recent baselines such as HGN jointly model paragraph selection, supporting-fact extraction, and answer prediction, while reasoning-path and beam-based retrievers optimize multi-step retrieval more directly (Asai et al., 2020; Fang et al., 2020; Zhang et al., 2024). These methods help explain the pattern observed here: learned joint modeling is particularly important for answer extraction, and flexible multi-hop retrieval is especially beneficial when the evidence chain is not well approximated by a fixed two-hop path. The purpose of our study is therefore narrower and more controlled—to quantify how much explainability can be obtained from explicit retrieval structure and sentence selection alone. Because the modules are separable, they can also be used as building blocks inside larger systems: for example, a stronger neural reader can be constrained to generate answers only from the selected supporting sentences, improving faithfulness while retaining auditable evidence outputs.

**Limitations and scope**

The main limitation is answer extraction. The weighted phrase-extraction reasoner is designed to be transparent, but it lacks the linguistic and contextual capacity of trained extractive readers; as shown in Table 9, open-ended span answers remain difficult for this baseline. A second limitation is retrieval flexibility: the explicit path module enforces a strict ordered two-hop structure, which matches HotpotQA better than 2WikiMultihopQA and may fail when more than two titles, alternative hop orders, or latent intermediate entities are needed. A third limitation is lexical dependence. Because the retriever uses surface-form overlap, it can miss semantically relevant paragraphs when the question and evidence are connected mainly by paraphrase or latent relations. Accordingly, the approach is most applicable as a transparent baseline or evidence layer in settings with bounded candidate contexts, relatively explicit entity mentions, and a need for

auditable supporting sentences; it should not be generalized to open-domain, variable-hop, or highly paraphrastic settings without stronger retrieval and reading components.

**Threats to validity**

The experiments use the provided candidate contexts in each dataset rather than an open-domain retrieval index. As a result, absolute answer accuracy should be interpreted only within this controlled setting and not as an estimate of full web-scale QA performance. We partially address robustness by evaluating two datasets and additional scenario slices by hop proxy, paragraph coverage, and answer type, but these checks do not replace external-domain validation. A second consideration is the hop proxy based on the number of labeled supporting sentences: it correlates with reasoning depth but is not identical to the number of logical inference steps. Finally, the deterministic reasoner intentionally sacrifices answer coverage for transparency; a more powerful reader would likely improve Answer EM/F1 while leaving the core evidence-selection interface unchanged. These limitations narrow the scope of the conclusions but do not change the central finding that explicit sentence selection consistently improves supporting-fact quality.

Across both datasets, the evidence selector consistently increases Supporting Fact F1 and improves the compactness of explanations. The path module increases paragraph recall and Supporting Fact F1 on HotpotQA, where a strict two-hop design matches the dataset's construction, but it does not improve results on 2WikiMultihopQA, which contains more variable evidence structures. These findings support a practical recommendation for QA assistants: always include sentence-level evidence selection, and enable explicit multi-hop paths when the domain and query distribution are dominated by two-step evidence chains.

## V. CONCLUSION AND RECOMMENDATION

We presented a fully interpretable pipeline for explainable multi-hop question answering that decomposes answering into Retriever, Evidence Selector, and Reasoner modules and exposes both sentence-level supporting facts and an ordered two-hop path. The aim of the study is not to introduce a new state-of-the-art reader, but to clarify the contribution of explicit evidence selection and path-structured retrieval in a controlled, reproducible setting.

The results answer the three objectives posed in this paper. First, sentence-level evidence selection consistently improves supporting-fact quality on both datasets and is the main driver of explanation compactness and precision. Second, an explicit ordered two-hop path modestly improves paragraph recall and supporting-fact quality on HotpotQA, where the dataset structure aligns with a two-page reasoning chain, but does not improve 2WikiMultihopQA, where evidence

structures are more variable. Third, the scenario analyses show that retrieval coverage is the main bottleneck, while deterministic answer extraction remains the main limitation for span answers.

These findings support a bounded conclusion: the proposed pipeline is best viewed as a transparent baseline and evidence layer for explanation-critical QA assistants operating on bounded candidate contexts, not as a replacement for stronger learned multi-hop QA systems. Future work should therefore integrate stronger readers or constrained generators, extend the path mechanism to variable-length reasoning chains, and combine lexical with dense retrieval to improve coverage without losing inspectability.

## REFERENCES

Asai, A., Hashimoto, K., Hajishirzi, H., Socher, R., & Xiong, C. (2020). Learning to Retrieve Reasoning Paths over Wikipedia Graph for Question Answering. *International Conference on Learning Representations (ICLR)*. https://doi.org/10.48550/arXiv.1911.10455

Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1870–1879. https://doi.org/10.18653/v1/p17-1171

De Cao, N., Aziz, W., & Titov, I. (2019). Question Answering by Reasoning across Documents with Graph Convolutional Networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2306–2317. https://doi.org/10.18653/v1/n19-1240

DeYoung, J., Jain, S., Rajani, N. F., Lehman, E., Xiong, C., Socher, R., & Wallace, B. C. (2020). ERASER: A Benchmark to Evaluate Rationalized NLP Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 4443–4458. https://doi.org/10.18653/v1/2020.acl-main.408

Fang, Y., Sun, S., Gan, Z., Pillai, R., Wang, S., & Liu, J. (2020). Hierarchical Graph Network for Multi-Hop Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8823–8838. https://doi.org/10.18653/v1/2020.emnlp-main.710

Gario, H., Fathurrohman, H., & Putra, H. R. (2026). Literature Review: Chatbots in Popular Culture: Roles, Trends, and Their Social Impact. *Jurnal Ilmiah Sistem Informasi*, *5*(1), 460–475. https://doi.org/10.51903/q8svda58

Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M.-W. (2020). Retrieval Augmented Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning (ICML)* (Vol. 119), 3929–3938. https://proceedings.mlr.press/v119/guu20a.html

Hao, L. W., & Liu, R. K. (2025). Transfer Learning Approach for Sentiment Analysis in Low-Resource Austronesian Languages Using Multilingual BERT. *Journal of Technology Informatics and Engineering*, *4*(1), 75–94. https://doi.org/10.51903/jtie.v4i1.276

Ho, X., Sugawara, S., & Aizawa, A. (2020). Constructing a Multi-Hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, 6609–6625. https://doi.org/10.18653/v1/2020.coling-main.580

Izacard, G., & Grave, E. (2021). Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 874–880. https://doi.org/10.18653/v1/2021.eacl-main.74

Jain, S., & Wallace, B. C. (2019). Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 3543–3556. https://doi.org/10.18653/v1/n19-1357

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-t. (2020). Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

Khattab, O., & Zaharia, M. (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 (SIGIR)*, 39–48. https://doi.org/10.1145/3397271.3401075

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Kulkarni, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474. https://proceedings.neurips.cc/paper/2020/hash/6b4922153558e45403063857e4e9766d-abstract.html

Li, R., & Du, X. (2023). Leveraging Structured Information for Explainable Multi-Hop Question Answering and Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 6779–6789. https://doi.org/10.18653/v1/2023.findings-emnlp.452

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 30, 4765–4774. https://proceedings.neurips.cc/paper/2017/hash/8a20a862115ef7d44bc9a5036582c36f-abstract.html

Miller, T. (2019). Explanation in Artificial Intelligence: Insights from the Social Sciences. *Artificial Intelligence*, 267, 1–38. https://doi.org/10.1016/j.artint.2018.07.007

Min, S., Chen, D., Hajishirzi, H., & Zettlemoyer, L. (2019). Multi-Hop Reading Comprehension through Question Decomposition and Rescoring. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 6097–6109. https://doi.org/10.18653/v1/d19-1632

Nogueira, R., & Cho, K. (2019). Passage Re-Ranking with BERT. *arXiv*, arXiv:1901.04085. https://arxiv.org/abs/1901.04085

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2383–2392. https://doi.org/10.18653/v1/d16-1264

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?" Explaining the Predictions of any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 1135–1144. https://doi.org/10.1145/2939672.2939778

Robertson, S., & Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4), 333–389. https://doi.org/10.1561/1500000019

Sriasih, S. D. S., Razak, F. A., & Ikhsan, H. A. I. (2025). AI-Driven Sentiment Analysis of Retail Investor Behavior during Market Volatility: A Study of Twitter Data in Southeast Asia. *Journal of Management and Informatics*, *4*(1), 741–756. https://doi.org/10.51903/jmi.v4i1.179

Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). FEVER: A Large-Scale Dataset for Fact Extraction and Verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 809–819. https://doi.org/10.18653/v1/n18-1074

Trivedi, H., Balasubramanian, N., Khot, T., & Sabharwal, A. (2022). MuSiQue: Multihop Questions via Single-Hop Question Composition. *arXiv*, arXiv:2205.09682. https://arxiv.org/abs/2205.09682

Welbl, J., Stenetorp, P., & Riedel, S. (2018). Constructing Datasets for Multi-Hop Reading Comprehension across Documents. *Transactions of the Association for Computational Linguistics*, 6, 287–302. https://doi.org/10.1162/tacl_a_00021

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E. H., Le, Q. V., & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35, 24824–24837. https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-abstract.html

Wiegreffe, S., & Pinter, Y. (2019). Attention Is Not Not Explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 11–20. https://doi.org/10.18653/v1/d19-1002

Xiong, W., Li, X. L., Iyer, S., Du, J., Lewis, P., Wang, W. Y., Mehdad, Y., Yih, W.-t., Riedel, S., Kiela, D., & Oguz, B. (2021). Answering Complex Open-Domain Questions with Multi-

Hop Dense Retrieval. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=asf_at7mjj

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., & Manning, C. D. (2018). HotpotQA: A Dataset for Diverse, Explainable Multi-Hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2369–2380. https://doi.org/10.18653/v1/d18-1259

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv*, arXiv:2210.03629. https://arxiv.org/abs/2210.03629

Zhang, J., Zhang, H., Zhang, D., Yong, L., & Huang, S. (2024). 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)*, 1718–1731. https://doi.org/10.18653/v1/2024.naacl-long.95