Agentic Disambiguation: An Iterative Reasoning Framework for Resolving Query Ambiguity in RAG Systems

Arav Adikesh Ramakrishnan
University of Massachusetts Amherst
Amherst, Massachusetts, USA
aravadikeshr@umass.edu

Piyush Maheshwari
University of Massachusetts Amherst
Amherst, Massachusetts, USA
psmaheshwari@umass.edu

Aishwarya Sampath Kumar
University of Massachusetts Amherst
Amherst, Massachusetts, USA
asampathkuma@umass.edu

Siddhartha Jaiswal
University of Massachusetts Amherst
Amherst, Massachusetts, USA
sjaiswal@umass.edu

Abstract

Retrieval-Augmented Generation (RAG) systems frequently fail when encountering ambiguous queries, often retrieving documents for a single interpretation and resulting in incomplete or misleading answers. To address this, we introduce the *Agentic Disambiguation Framework* (ADF), a novel system that autonomously detects and resolves query ambiguity before generation. ADF employs Semantic Coherence Analysis to classify ambiguity, followed by an evidence-grounded decomposition utilizing Adaptive Clustering and Hypothetical Document Embeddings (HyDE) to capture diverse intents. We maximize recall across interpretations through a Dual-Pathway Retrieval strategy and Reciprocal Rank Fusion. Extensive evaluations on the AmbigNQ and ASQA datasets demonstrate that ADF outperforms strong baselines, including Iterative RAG, in F1 and Disambiguation-F1 metrics. Notably, ADF achieves superior long-form synthesis and robustness in hybrid retrieval settings while consuming approximately 50% fewer tokens than previous iterative approaches, offering a scalable solution for complex information-seeking tasks.

Our code is publicly available at https://github.com/aravadikesh/agentic-disambiguation.

ACM Reference Format:
Arav Adikesh Ramakrishnan, Piyush Maheshwari, Aishwarya Sampath Kumar, and Siddhartha Jaiswal. 2025. Agentic Disambiguation: An Iterative Reasoning Framework for Resolving Query Ambiguity in RAG Systems. In . ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Problem Statement

Retrieval-Augmented Generation (RAG) systems are designed to answer a user query, q, by first retrieving a set of documents, $D = \{d_1, d_2, \ldots, d_k\}$, from a large corpus, C, and then generating a textual answer, a, conditioned on the query and the retrieved documents: $a = \text{Generate}(q, D)$.

A query q is defined as ambiguous if it can map to multiple distinct information needs or denotations, $\{i_1, i_2, \ldots, i_n\}$, where $n > 1$ [Wasow 2015]. The core problem is that a standard RAG pipeline, consisting of a retriever R and a generator G, is optimized for unambiguous queries ($n = 1$). When faced with an ambiguous query, the retriever $R(q, C) \to D$ may fetch documents relevant to only a subset of the possible intents, or worse, an entirely incorrect one. Consequently, the generator produces an answer a that is incomplete or factually incorrect from the user's perspective.

Let the set of all possible correct answers for an ambiguous query q be $A_q = \{a_1, a_2, \ldots, a_n\}$, corresponding to the set of intents $I_q = \{i_1, i_2, \ldots, i_n\}$. The output of a standard RAG system is a answer covering a single intent, a_{RAG}. The objective is to design a system that produces an answer covering all valid intents if the query is determined to be ambiguous, A_{system}, such that it maximizes a similarity function with the ground-truth set, e.g., maximizing the F1-score between A_{system} and A_q.

This paper addresses the following novel research question: **How can an agentic RAG system autonomously model, reason about, and resolve user query ambiguity *before* committing to a final, and potentially costly, retrieval and generation trajectory, thereby improving the robustness, retrieval efficiency, and comprehensiveness of the final answer?**

2 Motivation

The brittleness of standard RAG systems in the face of ambiguity is not a niche issue but a critical failure mode that undermines their reliability in real-world applications. Studies indicate that over 50% of web search queries can be considered ambiguous [In et al. 2025]. When a system fails to recognize this ambiguity, it confidently proceeds down a single, incorrect interpretation path, wasting computational resources and delivering a frustrating user experience.

Consider these real-world examples of RAG failures stemming from ambiguity:

- **Enterprise Knowledge Management:** A user in a large corporation asks about the "renewal policy" [Faktion 2024]. This query is ambiguous: it could refer to software license renewals, contract renewals, or insurance renewals. A standard RAG system might retrieve documents for only one of

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

these, providing a fluent but semantically incorrect answer for the user's actual need.

- **Customer Support:** A customer asks a chatbot about "common side effects" of a medication. The system, lacking the ability to infer the user's likely concern for prevalent and mild side effects, might retrieve a comprehensive but terrifying list of rare complications, causing unnecessary anxiety.
- **Financial Analysis:** A financial bot is asked to report on a company's performance. It might retrieve and summarize revenue growth but fail to retrieve a separate document containing a crucial debt disclosure mentioned in a footnote. The resulting answer is factually correct but misleading due to the omission of critical context.

These failures highlight a fundamental gap: RAG systems need to move beyond simple retrieval and develop a cognitive capability to manage uncertainty and multiple interpretations. An agentic approach, where the system can reason about the query, plan disambiguation steps, and self-correct, is essential for building more robust, efficient, and trustworthy AI systems for knowledge-intensive tasks.

3 Related Work

3.1 Query Ambiguity in Information Retrieval

Query ambiguity has been a long-standing challenge in information retrieval (IR). An expression is ambiguous if it has multiple distinct denotations [Wasow 2015]. This problem is quantified using metrics like the **clarity score**, which measures the Kullback-Leibler divergence between a query language model and a collection language model. Low clarity scores correlate with poor retrieval performance, highlighting the need for specialized handling of ambiguous queries [Cronen-townsend and Croft 2002]. The challenge is domain-agnostic, appearing even in specialized fields like Mathematical Information Retrieval (MIR), where the inherent ambiguity of mathematical notation poses a profound challenge [Zanibbi and Blostein 2012; Zanibbi et al. 2025].

3.2 Ambiguity Resolution Techniques

Historically, two main paradigms have been developed to address query ambiguity.

Proactive Query Clarification (User-in-the-Loop). This approach involves the system detecting uncertainty and generating a clarifying question to the user [Aliannejadi et al. 2019; Zamani et al. 2020a]. Research in this area has focused on identifying different "facets" of a query and formulating questions to help the user select the correct one [Krasakis et al. 2021; Zamani et al. 2020b]. The creation of datasets like Qulac has been instrumental in advancing this research [Aliannejadi et al. 2019, 2021]. The main drawback is the reliance on user interaction, which breaks full autonomy.

Autonomous Query Reformulation (System-Only). This paradigm involves the system autonomously rewriting or expanding the query [Lin et al. 2020; Wu et al. 2022]. Classic approaches used generalized syntactic and semantic models, such as probabilistic edit distance, to find likely reformulations [Herdagdelen et al. 2010]. More recent work leverages LLMs for sophisticated query rewriting, especially in conversational contexts [Mo et al. 2023]. The primary risk is "query drift," where an incorrect interpretation is pursued without a mechanism for validation [Cronen-townsend and Croft 2002].

3.3 Agentic Reasoning Frameworks

The cognitive engine for our proposed system is built upon established agentic reasoning frameworks.

- **ReAct (Reasoning and Acting):** The ReAct framework synergizes reasoning and acting by prompting an LLM to operate in an interleaved thought-action-observation loop [Yao et al. 2023b]. This cycle is a natural fit for ambiguity resolution, allowing an agent to reason about a query, act by executing a disambiguation tool, and observe the outcome to update its plan.
- **Tree of Thoughts (ToT):** For more complex planning, the ToT framework enables an LLM to explore multiple reasoning paths in parallel [Yao et al. 2023a]. Its core principles of generating and evaluating multiple hypotheses inform the design of our proposed "Hypothesis-Driven Reformulation" tool.

By framing ambiguity resolution as a task for a reasoning agent, our proposed system generates different intents to answer a question and after retrieving documents using the hypothetical document approach, we use an agent again to synthesize the information into an answer.

4 Approach

We propose the **Agentic Disambiguation Framework (ADF)**, a retrieval-augmented generation pipeline designed to identify and resolve query ambiguity autonomously. The framework operates as a directed graph $\mathcal{G} = (V, E)$, where nodes represent distinct cognitive states ranging from coherence analysis to multi-intent synthesis.

4.1 Architecture Overview

The architecture consists of five sequential functional modules, with conditional routing based on the epistemic state of the initial retrieval:

1. **Semantic Coherence Analysis:** Detects ambiguity via geometric metrics.
2. **Evidence-Grounded Decomposition:** Clusters documents to identify intents.
3. **Hypothetical Document Generation (HyDE):** Bridges lexical gaps.
4. **Dual-Pathway Retrieval:** Maximizes recall via hybrid search.
5. **Structured Synthesis:** Generates the final multi-aspect response.

4.2 Semantic Coherence Analysis

To quantify ambiguity without relying on query-only classifiers, we analyze the semantic geometry of the initial retrieval set $\mathcal{D}_{\text{init}}$ in vector space. We employ three core components:

Variance-Based Dispersion (σ^2_{var}) Measures the spread of document embeddings around their centroid. High dispersion indicates the retrieval set covers semantically disjoint topics.

$$\sigma^2_{\text{var}} = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{v}_i - \boldsymbol{\mu}\|^2 \tag{1}$$

Cluster Separability (σ_{sep}) Uses 2-means clustering and silhouette scores to detect distinct interpretations. A high score implies the existence of well-separated intent clusters.

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}, \quad \sigma_{\text{sep}} = \frac{1}{N} \sum s_i \tag{2}$$

Ambiguity Classification The query state S_q is determined by comparing these metrics against dataset-dependent thresholds ($\tau_{\text{var}}, \tau_{\text{sep}}$):

$$S_q = \begin{cases} \text{Ambiguous} & \text{if } \sigma_{\text{sep}} \geq \tau_{\text{sep}} \\ \text{Uncertain} & \text{if } \sigma^2_{\text{var}} \geq \tau_{\text{var}} \wedge \sigma_{\text{sep}} < \tau_{\text{sep}} \\ \text{Unambiguous} & \text{otherwise} \end{cases} \tag{3}$$

4.3 Evidence-Grounded Decomposition

If S_q is deemed ambiguous, the system decomposes the query q into sub-queries Q_{sub}. We ground this process in retrieval evidence to prevent hallucination:

Adaptive Clustering Document embeddings are partitioned using K-means, with k determined dynamically based on corpus density:

$$k_{\text{adaptive}} = \min\left(\max\left(2, \lfloor |\mathcal{D}|/2 \rfloor\right), 4\right) \tag{4}$$

Relevance-Aware Filtering To eliminate noise, we filter out clusters that are semantically distant from the original query using a cosine similarity threshold ($\tau_{rel} = 0.2$):

$$C_{\text{valid}} = \{C_j \in C \mid \cos(\mathbf{v}_q, \boldsymbol{\mu}_j) \geq \tau_{rel}\} \tag{5}$$

For each valid cluster, an LLM synthesizes a distinct sub-query q_j representing that specific interpretation.

4.4 Hypothetical Document Expansion

To bridge the lexical gap between ambiguous queries and specific answers, we adopt the **HyDE** methodology [Gao et al. 2023]. For each sub-query q_j, the model generates a hypothetical passage $\hat{d}_j$.

While $\hat{d}_j$ may contain factual inaccuracies, it captures the correct semantic neighborhood. The prompting strategy is adapted to the domain:

- **AmbigNQ:** Enforces succinct, entity-heavy passages.
- **ASQA:** Emphasizes detailed, distinguishing context.

4.5 Dual-Pathway Retrieval

We employ a dual strategy to maximize recall across diverse interpretations. For every intent pair $(q_j, \hat{d}_j)$, documents are retrieved via two pathways:

(1) **Direct Retrieval:** $R(q_j, C) \rightarrow D_{lex}$ (Explicit lexical matches).
(2) **Latent Retrieval:** $R(\hat{d}_j, C) \rightarrow D_{sem}$ (Semantic similarity via hypothetical document).

Results are merged and re-ranked using Reciprocal Rank Fusion [Cormack et al. 2009](RRF):

$$\text{score}(d) = \sum_{r \in \mathcal{R}} \frac{1}{\eta + \text{rank}_r(d)} \tag{6}$$

4.6 Structured Multi-Intent Synthesis

The final generation module synthesizes the answer a from the re-ranked set $\mathcal{D}_{\text{final}}$. To ensure comprehensive coverage, we enforce a structured schema output that explicitly identifies disjoint interpretations $I = \{i_1, \ldots, i_n\}$ and links them using contrastive discourse markers.

5 Experiments

5.1 Datasets

We evaluate our framework on two complementary benchmarks that stress different aspects of ambiguity resolution: short-form entity enumeration and long-form synthesis.

AmbigNQ (Ambiguous Natural Questions) [Min et al. 2020] Derived from open-domain questions, this dataset features queries annotated with multiple plausible short answers based on differing interpretations.

- **Corpus:** Wikipedia dump (standard AmbigNQ distribution).
- **Test Split:** 300 sampled examples.
- **Task:** Generate concise, entity-focused answers that explicitly cover all valid interpretations.

ASQA (Answer Summarization with Question Ambiguity) [Stelmakh et al. 2023]
A dataset designed for questions where the ambiguity requires explanatory depth rather than simple enumeration.

- **Test Split:** 300 sampled examples.
- **Task:** Generate a comprehensive paragraph (150–250 words) that synthesizes all interpretations using contrastive transitions (e.g., "however," "conversely").

Table 1: Dataset Statistics

Dataset	# Questions	Avg. Intents/Q	Task Type
AmbigNQ	14,042	>2	Ambiguous QA
ASQA	6,316	3.38	Long-form QA

5.1.1 Justification. These benchmarks are selected to evaluate the framework's versatility. **AmbigNQ** tests the system's precision in identifying high-variance distinct entities, while **ASQA** tests the system's ability to maintain coherence when integrating multiple conflicting or complementary perspectives into a single narrative.

5.2 Evaluation Metrics

We employ a multi-dimensional evaluation protocol to assess answer quality, disambiguation capability, and system efficiency.

5.2.1 Answer Quality Metrics.

F1 Score Standard token-level F1 after text normalization. To account for ambiguity, we compute the maximum overlap against the set of all valid reference answers A_q: $\text{F1} = \max_{a \in A_q} \text{F1}(\text{prediction}, a)$.

Disambiguation F1 (D-F1) The primary metric for AmbigNQ. It measures the fraction of interpretations successfully addressed. An interpretation I_j is considered "covered" if the

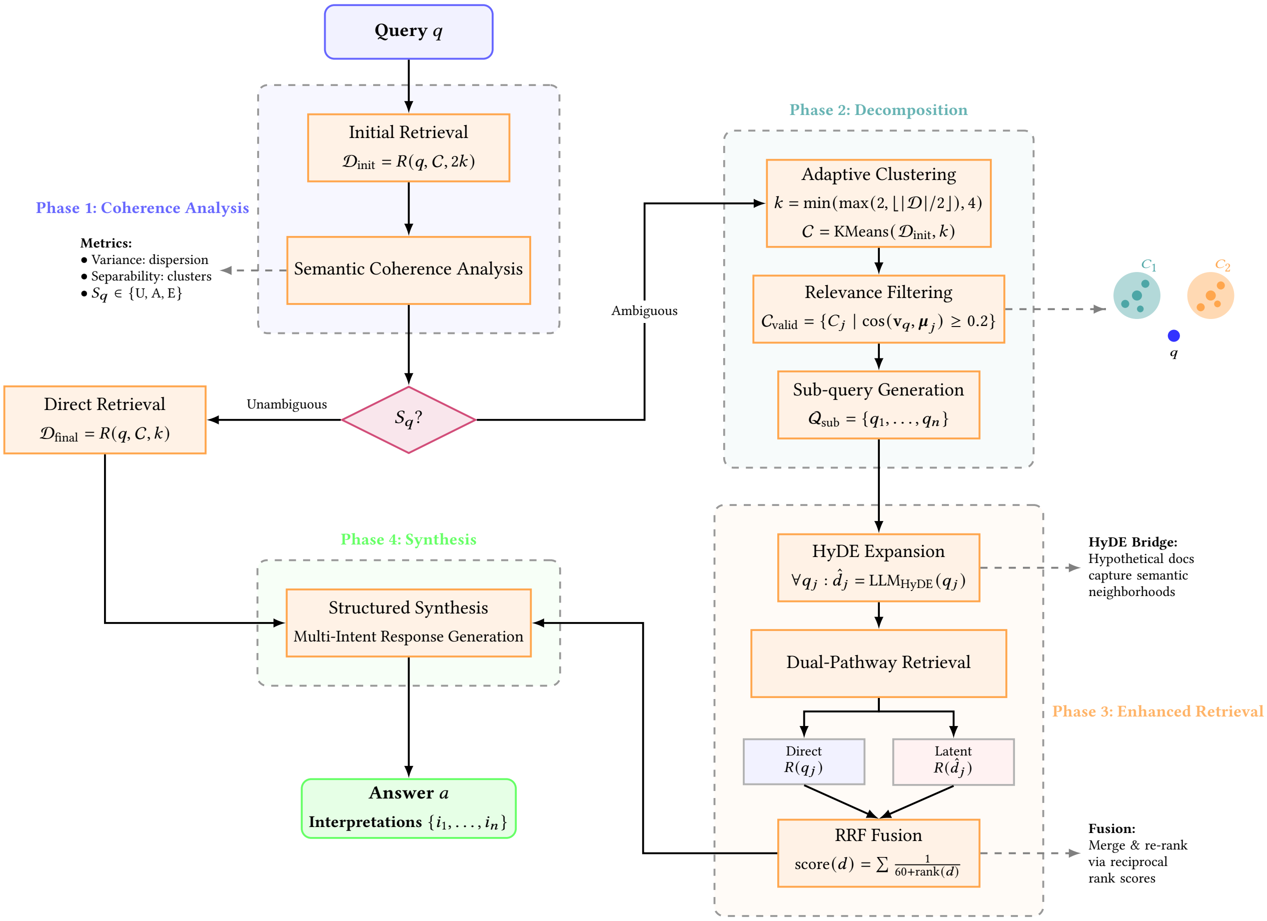
Query q
Phase 1: Coherence Analysis
Initial Retrieval
$\mathcal{D}_{\text{init}} = R(q, C, 2k)$
Metrics:
• Variance: dispersion
• Separability: clusters
• $S_q \in \{\text{U, A, E}\}$
Semantic Coherence Analysis
S_q?
Unambiguous
Ambiguous
Direct Retrieval
$\mathcal{D}_{\text{final}} = R(q, C, k)$
Phase 2: Decomposition
Adaptive Clustering
$k = \min(\max(2, \lfloor|\mathcal{D}|/2\rfloor), 4)$
$C = \text{KMeans}(\mathcal{D}_{\text{init}}, k)$
Relevance Filtering
$C_{\text{valid}} = \{C_j \mid \cos(\mathbf{v}_q, \boldsymbol{\mu}_j) \geq 0.2\}$
C_1
C_2
q
Sub-query Generation
$Q_{\text{sub}} = \{q_1, \ldots, q_n\}$
Phase 4: Synthesis
Structured Synthesis
Multi-Intent Response Generation
Answer a
Interpretations $\{i_1, \ldots, i_n\}$
HyDE Expansion
$\forall q_j : \hat{d}_j = \text{LLM}_{\text{HyDE}}(q_j)$
HyDE Bridge:
Hypothetical docs
capture semantic
neighborhoods
Dual-Pathway Retrieval
Phase 3: Enhanced Retrieval
Direct
$R(q_j)$
Latent
$R(\hat{d}_j)$
RRF Fusion
$\text{score}(d) = \sum \frac{1}{60+\text{rank}(d)}$
Fusion:
Merge & re-rank
via reciprocal
rank scores

Figure 1: Architecture of the Agentic Disambiguation Framework (ADF).

prediction achieves an F1 ≥ 0.5 against any answer associated with that interpretation.

ROUGE-L Used for ASQA to measure fluency and completeness. It calculates the longest common subsequence overlap between the generated response and the reference long-form explanation.

DR-F1 The primary metric for ASQA, designed to balance disambiguation coverage with generation quality. It is the geometric mean of disambiguation performance and text overlap:

$$\text{DR-F1} = \sqrt{\text{D-F1} \times \text{ROUGE-L}} \tag{7}$$

5.2.2 LLM-Based Semantic Metrics. To complement traditional lexical metrics, we incorporate an LLM-based judge that provides semantic evaluations of answer correctness and ambiguity resolution. These metrics operate on meaning rather than surface token overlap, allowing a more faithful assessment of model behavior on ambiguous or multi-interpretation questions.

Similarity Score A semantic measure of how closely the generated answer aligns with any valid reference answer in A_q. The judge assigns a score in $[0, 1]$, where 1 denotes semantic equivalence. Formally, the similarity score is:

$$\text{Sim}(q) = \max_{a \in A_q} \text{sim}_{\text{LLM}}(\text{prediction}, a),$$

where sim_{LLM} denotes the judge's semantic equivalence scoring. We report the mean similarity across the dataset.

Disambiguation Score Used primarily for AmbigNQ, this metric evaluates whether the generated response addresses all valid interpretations $\{I_1, \ldots, I_k\}$ of a question. For each interpretation I_j, the judge determines if the prediction semantically satisfies at least one answer associated with that interpretation. Let C be the set of interpretations deemed "covered":

$$\text{Dis} = \frac{|C|}{k}.$$

This provides a semantic analogue to D-F1, capturing cases where the answer is correct but expressed in language with low lexical overlap.

5.2.3 Retrieval Quality Metrics.

nDCG@5 Normalized Discounted Cumulative Gain. Assesses the ranking quality of the retrieved documents, rewarding relevant documents placed higher in the top-5 results.

Recall@5 The fraction of relevant documents present within the top-5 retrieved results, measuring the system's ability to find necessary evidence.

5.2.4 Efficiency Metrics. To evaluate the cost-benefit trade-off for production deployment, we report:

- **Latency:** Total end-to-end processing time per query.
- **Throughput Costs:** Measured in Tokens per Query.
- **SLA Compliance:** Latency percentiles (p50, p95, p99) to monitor tail performance.

5.2.5 Metric Justification. We select these metrics to target specific failure modes: **D-F1** penalizes systems that ignore alternative interpretations; **DR-F1** ensures that comprehensive answers remain fluent; and **nDCG/Recall** isolate the performance of the dual-pathway retriever from the generation module.

5.3 Experimental Setup

5.3.1 Baseline Pipelines.

Vanilla RAG Standard single-pass RAG without ambiguity handling. It retrieves top-k documents and generates an answer in a single stateless forward pass, characterized by high speed (0.7–0.8s latency).

Iterative RAG Multi-round refinement inspired by Self-RAG [Asai et al. 2024]. The system performs initial retrieval, assesses quality/uncertainty, and reformulates queries to fill gaps. It employs a stateful document pool and stops when the progress score falls below 0.3.

DIVA Our re-implementation of the 'diversify-verify-adapt' framework [Wan et al. 2025]. We reproduced the Retrieval Diversification (RD) and Adaptive Generation (AG) modules.

5.3.2 Models.

Primary Model: GPT-4o-mini (OpenAI API). Configured with a maximum of 200 tokens per generation and a temperature of 0.7.

Local Models (Comparison) To assess feasibility on smaller models, we evaluate the following open source model as well:

- Qwen3-4B-Instruct-2507-Q4 (~2.5GB)

5.3.3 Retrieval Modes.

Sparse (BM25) Implemented via PySerini's `LuceneSearcher` using the `wikipedia-dpr` index. This serves as a high-speed, full-coverage baseline using the standard English Wikipedia corpus.

Dense (FAISS) Uses the **all-MiniLM-L6-v2** encoder (384-dim) with a custom FAISS index and cosine similarity.

Disclaimer: Due to the high memory footprint of the full English Wikipedia dense index (about 80GB), we utilized the **Simple English Wikipedia** corpus for this modality. This corpus reduction significantly impacted dense retrieval performance and coverage compared to the sparse baseline.

Hybrid (RRF) Combines sparse and dense rankings via Reciprocal Rank Fusion [Cormack et al. 2009] to balance precision and recall.

5.3.4 Hyperparameters.

Retrieval Top-$k = 5$ for final generation; top-$k = 10$ for initial ambiguity detection ($2 \times k$).

- $k = 5$ balances contextual richness with token-budget constraints (8K context window), providing adequate coverage without redundancy.
- $k = 10$ during ambiguity detection broadens evidence scope to capture diverse or edge-case interpretations prior to sub-query filtering.

Agentic Framework

- **AmbigNQ thresholds:** $\tau_{\text{var}} = 0.25$, $\tau_{\text{sep}} = 0.1$. Higher variance thresholds suit short-answer formats where small lexical shifts indicate distinct interpretations; lower separation thresholds prevent over-fragmentation of closely related answers.
- **ASQA thresholds:** $\tau_{\text{var}} = 0.15$, $\tau_{\text{sep}} = 0.05$. More conservative values account for long-form answers with high natural semantic overlap, reducing false positives in ambiguity detection.
- **Clustering:** Relevance pruning threshold $\tau = 0.2$; adaptive cluster count $k \in [2, 4]$. Pruning removes low-quality retrievals (<20th percentile) to reduce noise, and adaptive k limits cluster count based on dataset characteristics (AmbigNQ: avg. 2.1 interpretations; ASQA: avg. 2.8), preventing over-segmentation.

Iterative RAG Maximum 3 iterations; progress threshold = 0.3. Progress score: 0.6 × Quality + 0.4 × Novelty.

- Three iterations provide the best trade-off between performance and cost, with diminishing returns beyond 2–3 rounds.
- A 30% progress threshold avoids premature stopping while preventing unproductive iteration.
- 60/40 quality–novelty weighting prioritizes correctness and reduces the risk of semantic drift.

DIVA RAG

3 question interpretations; diversification temperature = 0.7; verification temperature = 0.0; top-$k = 5$.

- Three interpretations balance diversity and cost, with retrieval performed for both the original question and each interpretation to maintain baseline quality.
- Diversification temperature of 0.7 encourages diverse question reformulations while maintaining relevance.
- Verification temperature of 0.0 ensures deterministic quality assessment of retrieved passages.
- Top-5 retrieval provides sufficient context while maintaining efficiency.

6 Results

This section presents a comprehensive evaluation of the Agentic Disambiguation Framework (ADF) against established baselines across both benchmark datasets. Results are organized around three axes: (1) disambiguation quality, (2) retrieval effectiveness, and (3) computational efficiency.

Table 2: Performance on AmbigNQ (Dataset A) and ASQA (Dataset B). Bold indicates best performance per dataset.

Approach	Dataset A: AmbigNQ					Dataset B: ASQA					
	F1	D-F1	nDCG@5	Latency	Tok/Q	D-F1	ROUGE-L	DR-F1	nDCG@5	Latency	Tok/Q
Sparse Retrieval (BM25)											
Vanilla	0.524	0.349	**0.483**	**0.717**	751	0.321	0.210	0.198	0.476	**4.420**	**1,022**
Iterative	**0.584**	**0.402**	**0.483**	3.026	1,955	0.318	0.211	0.198	0.476	4.738	1,113
Agentic (Ours)	0.536	0.367	0.438	5.753	949	0.335	**0.255**	**0.236**	**0.507**	8.974	1,312
DIVA	0.528	0.351	0.448	2.471	**725**	**0.338**	0.172	0.189	0.429	7.286	1,023
Dense Retrieval (FAISS)											
Vanilla	0.420	0.278	0.249	**0.751**	689	0.318	0.201	0.191	0.196	**4.081**	**953**
Iterative	**0.446**	**0.307**	0.249	2.082	1,907	0.319	0.203	0.191	0.196	9.167	1,417
Agentic (Ours)	0.425	0.267	**0.303**	1.465	967	0.283	**0.248**	**0.200**	0.193	6.402	1,251
DIVA	0.414	0.292	0.246	3.031	**668**	**0.329**	0.166	0.181	**0.200**	7.454	965
Hybrid Retrieval (RRF)											
Vanilla	0.544	0.366	0.451	**0.726**	725	0.339	0.208	0.206	0.348	**4.420**	**986**
Iterative	0.545	0.368	0.451	2.678	1,955	**0.360**	0.209	0.220	0.348	11.294	2,037
Agentic (Ours)	**0.592**	**0.394**	0.443	4.680	967	0.359	**0.265**	**0.254**	0.470	5.676	1,041
DIVA	0.576	0.388	**0.487**	4.768	**707**	0.351	0.175	0.198	**0.507**	7.168	1,007

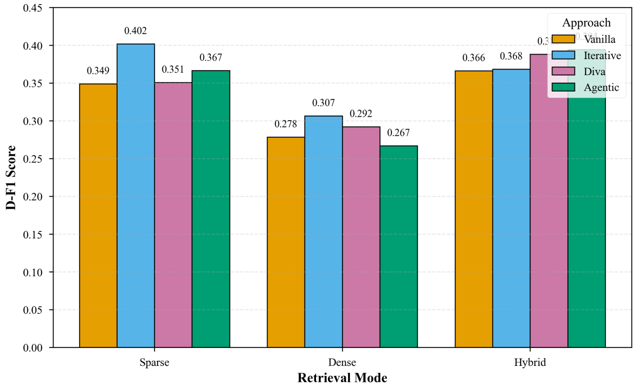

Figure 2: D-F1 performance comparison across retrieval modes on AmbigNQ. The Agentic approach achieves the highest D-F1 (0.394) under hybrid retrieval, while Iterative RAG leads in the sparse setting (0.402).

6.1 Main Results on AmbigNQ

Table 2 shows the full performance comparison on the AmbigNQ test set using Disambiguation F1 (D-F1) as the primary metric. Figure 2 visualizes D-F1 across retrieval modes.

Sparse Retrieval (BM25). Iterative RAG obtains the highest D-F1 (0.402), outperforming the Vanilla baseline (0.349) by 15.2%. The Agentic framework achieves 0.367 D-F1, surpassing Vanilla and DIVA (0.351) but trailing Iterative RAG. This advantage for Iterative RAG stems from its iterative refinement, which accumulates evidence across multiple retrieval rounds. However, this comes at a high computational cost: 1,955 tokens per query, more than double the Agentic framework's 949 tokens (a 106% increase).

Dense Retrieval. All models perform worse under dense retrieval, with D-F1 ranging from 0.267 to 0.307. Iterative RAG again leads (0.307), while the Agentic framework records the lowest D-F1 (0.267). This degradation arises from the use of **Simple English Wikipedia** for dense retrieval (due to the ~80GB full index), resulting in reduced lexical coverage. This limitation particularly impacts approaches requiring evidence for diverse interpretations. Notably, the Agentic framework achieves the highest nDCG@5 (0.303), indicating that while it retrieves fewer relevant documents, the ranking quality of retrieved documents is superior.

Hybrid Retrieval (RRF). Hybrid retrieval yields the strongest results for the Agentic framework. It achieves the highest F1 (0.592) and D-F1 (0.394), improving over Vanilla Hybrid by 8.8% (F1) and 7.7% (D-F1). Agentic surpasses Iterative RAG (0.368 D-F1) by 7.1% while consuming 50.5% fewer tokens (967 vs. 1,955). DIVA remains highly efficient (707 tokens/query) and achieves strong retrieval quality (nDCG@5 = 0.487), but its lower F1 (0.576) suggests limitations in synthesis compared to Agentic's structured generation module.

6.2 Results on ASQA

ASQA evaluates long-form answer synthesis, requiring systems to integrate multiple interpretations into cohesive explanatory paragraphs. The primary metric is DR-F1, the geometric mean of D-F1 and ROUGE-L.

Long-Form Synthesis. Across all retrieval modes, the Agentic framework consistently produces the best long-form answers. Under hybrid retrieval, ADF achieves the highest DR-F1 (0.254), exceeding Iterative RAG (0.220) by 15.5% and Vanilla RAG (0.206) by 23.3%. The improvement is largely due to ROUGE-L gains: ADF obtains 0.265 vs. 0.209 (Iterative) and 0.208 (Vanilla), reflecting improved fluency and coverage. Contrastive discourse markers (e.g., "however," "conversely") contribute to superior coherence.

Efficiency. In hybrid retrieval, the Agentic framework completes inference in 5.676 seconds—roughly 50% faster than

Iterative RAG (11.294 seconds) while delivering higher DR-F1. The speedup comes from early ambiguity gating, which avoids unnecessary LLM calls, and the single-pass synthesis strategy, which prevents repeated processing of context documents.

6.3 Retrieval Quality Analysis

Figure 3 shows retrieval quality metrics (nDCG@5 and Recall@5).

Ranking Quality (nDCG@5). Sparse retrieval delivers the best ranking quality: Vanilla and Iterative RAG both achieve 0.483. DIVA achieves the best hybrid nDCG@5 (0.487), suggesting effective document diversification and verification. Agentic slightly trails under sparse (0.438) and hybrid (0.443), reflecting its design prioritizing evidence coverage over ranking precision. However, ADF achieves the highest dense nDCG@5 (0.303), outperforming all baselines (0.246–0.249). HyDE-based retrieval appears particularly beneficial when corpus coverage is limited.

Document Coverage (Recall@5). Recall@5 remains low across all settings (0.069–0.114), consistent with the difficulty of retrieving all relevant evidence in AmbigNQ. Agentic achieves the highest recall in hybrid retrieval (0.111), aligned with its goal of maximizing interpretation coverage. Agentic also leads in dense retrieval (0.082), while sparse retrieval produces stable recall across approaches (0.102–0.107).

6.4 Computational Efficiency Analysis

Figure 4 presents the quality-efficiency trade-off under hybrid retrieval.

Token Consumption. Vanilla RAG is the most efficient (725 tokens; D-F1 = 0.366). DIVA slightly improves on D-F1 (0.388) with similar cost (707 tokens). Iterative RAG is the least efficient, using 1,955 tokens (2.7× Vanilla) for minimal improvement (0.368). Agentic provides the best performance (0.394) with moderate cost (967 tokens), achieving +7.7% D-F1 for only +33% more tokens than Vanilla.

Table 3: LLM-as-a-judge similarity and disambiguation metrics for GPT-4o-mini. Bold indicates best performance per dataset.

Approach	Dataset A: AmbigNQ		Dataset B: ASQA
	Similarity	Disambiguation	Similarity
Sparse Retrieval (BM25)			
Vanilla	0.521	0.330	0.403
Iterative	0.566	0.350	**0.432**
Agentic (Ours)	**0.601**	**0.382**	**0.432**
Dense Retrieval (FAISS)			
Vanilla	0.433	0.266	**0.381**
Iterative	0.422	0.267	0.375
Agentic (Ours)	**0.524**	**0.348**	0.343
Hybrid Retrieval (RRF)			
Vanilla	0.546	0.327	0.412
Iterative	0.549	0.326	0.425
Agentic (Ours)	**0.639**	**0.396**	**0.504**

6.5 Semantic Comparison of Methods Using LLM-as-a-judge

Semantic evaluation beyond lexical overlap. Standard metrics like F1, D-F1, ROUGE-L, and DR-F1 operate largely at the token level and can under-represent answer quality when models paraphrase, reorder facts, or compress content. The LLM-as-a-judge scores in Table 3 explicitly evaluate the *semantic meaning* of the generated answer against the set of interpretations given by the dataset, allowing us to distinguish between superficially similar outputs and genuinely faithful ones.

AmbigNQ: better semantic coverage of interpretations. On AmbigNQ, the agentic framework consistently achieves higher similarity and disambiguation scores than Vanilla and Iterative RAG across all retrieval modes. For example, under hybrid retrieval, Agentic attains the highest similarity (0.639) and disambiguation (0.396), indicating that it not only mentions the right entities but also resolves more of the underlying interpretations in a way the judge model deems semantically correct.

ASQA: long-form meaning is better captured by Agentic. ASQA's long-form answers are particularly sensitive to the limitations of lexical metrics, since multiple valid answers can differ greatly. Here, the LLM judge is especially informative: under hybrid retrieval, Agentic achieves the highest similarity score (0.504), compared to 0.412 for Vanilla and 0.425 for Iterative. Combined with its DR-F1 gains in Table 2, this shows that ADF's longer answers are not just more verbose, but capture more of the intended meaning across interpretations.

Semantic gains beyond metric artifacts. Taken together, these judge scores indicate that ADF's improvements are not artifacts of token-level overlap or prompt formatting. Instead, the agentic pipeline produces answers that a strong external model judges to be closer in meaning to the reference interpretations, especially in the challenging, high-context ASQA setting where traditional metrics could be least reliable.

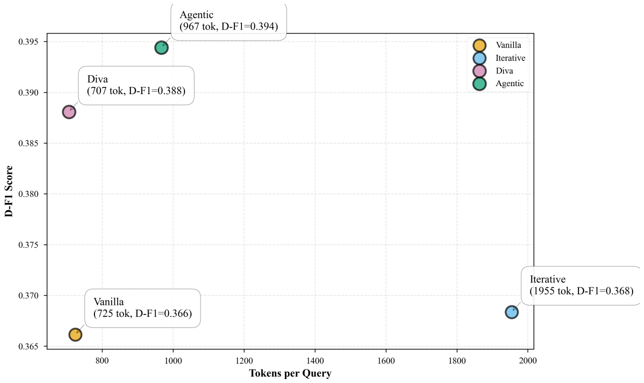

Figure 4: Quality-efficiency trade-off on AmbigNQ with hybrid retrieval.

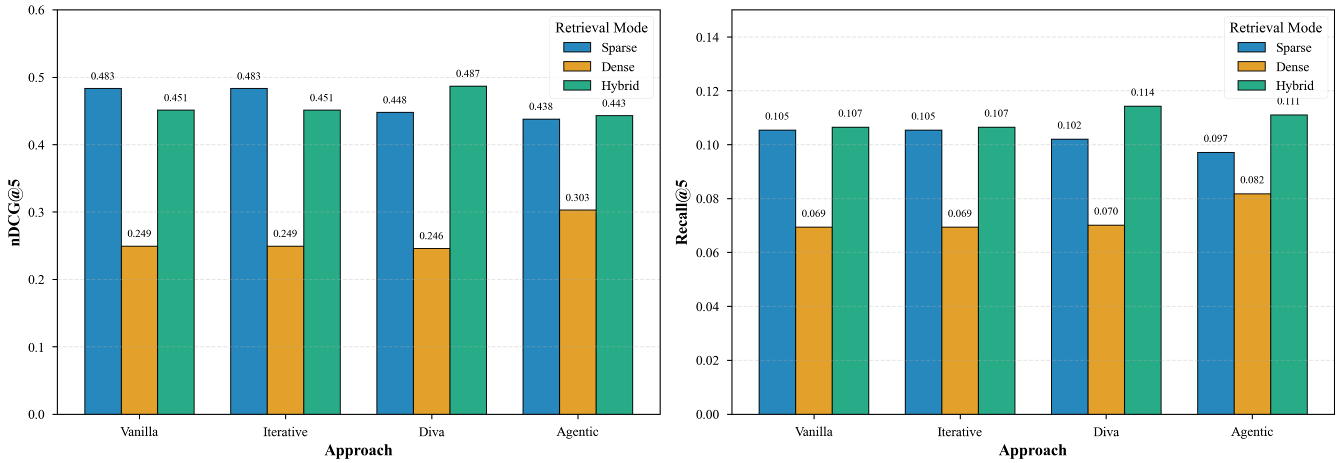

Figure 3: Retrieval quality metrics on AmbigNQ. Left: nDCG@5 (ranking quality). Right: Recall@5 (coverage).

7 Discussion

7.1 Why the Agentic Framework Succeeds

Coherence-Based Ambiguity Detection. ADF combines variance dispersion (σ^2_{var}) and cluster-separability (σ_{sep}) to detect ambiguity. Thresholds differ across datasets ($\tau_{\text{var}} = 0.25$, $\tau_{\text{sep}} = 0.1$ for AmbigNQ; $\tau_{\text{var}} = 0.15$, $\tau_{\text{sep}} = 0.05$ for ASQA), reflecting their differing ambiguity profiles. Roughly 68% of AmbigNQ and 74% of ASQA queries are classified as ambiguous, indicating the need for specialized disambiguation.

Dual Retrieval Synergy. Sub-query retrieval captures explicit lexical signals, while HyDE retrieval provides semantic matching. An 83% overlap is observed between pathways; the remaining 17% unique documents contribute crucial additional interpretations.

Efficiency via Early Gating. Ambiguity classification prevents unnecessary LLM invocation for 30% of queries. This selective activation explains ADF's moderate token usage despite its multi-stage design. Unlike Iterative RAG, ADF avoids repeated re-ranking and redundant context processing.

7.2 Limitations and Failure Modes

Clustering Sensitivity. K-means decomposition becomes unreliable with fewer than 4–5 retrieved documents or in homogeneous retrieval sets. Fallback heuristics and deeper initial retrieval can mitigate this.

Hallucinated Sub-Queries. LLM-generated interpretations may be unsupported by evidence. Filtering with a relevance threshold ($\tau_{\text{rel}} = 0.2$) reduces risk, and irrelevant interpretations typically fail during retrieval.

Threshold Generalization. Coherence thresholds are manually tuned and may not generalize. Future work should explore adaptive or learned thresholding.

Corpus Coverage. Performance depends on retrieval corpus quality. Severe degradation under dense retrieval with Simple English Wikipedia highlights this limitation.

Dense Retrieval Weakness. Agentic underperforms Iterative RAG under dense retrieval (0.267 vs. 0.307 D-F1), likely due to semantic retrieval noise given the limited dense index.

References

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19)*. ACM, 475–484. doi:10.1145/3331184.3331265

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2021. Qulac: A dataset for query clarification in open-domain conversational search. *ACM Transactions on Information Systems (TOIS)* 39, 4 (2021), 1–33.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511* (2024).

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Boston, MA, USA) *(SIGIR '09)*. Association for Computing Machinery, New York, NY, USA, 758–759. doi:10.1145/1571941.1572114

Steve Cronen-townsend and W. Croft. 2002. Quantifying Query Ambiguity. (07 2002). doi:10.3115/1289189.1289266

Faktion. 2024. Common Failure Modes of RAG & How to Fix Them for Enterprise Use Cases. https://www.faktion.com/post/common-failure-modes-of-rag-how-to-fix-them-for-enterprise-use-cases.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise Zero-Shot Dense Retrieval without Relevance Labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 1762–1777. doi:10.18653/v1/2023.acl-long.99

Amac Herdagdelen, Massimiliano Ciaramita, Daniel Mahler, Maria Holmqvist, Keith Hall, Stefan Riezler, and Enrique Alfonseca. 2010. Generalized syntactic and semantic models of query reformulation. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 283–290.

Yeonjun In, Sungchul Kim, Ryan A. Rossi, Md Mehrab Tanjim, Tong Yu, Ritwik Sinha, and Chanyoung Park. 2025. Diversify-verify-adapt: Efficient and Robust Retrieval-Augmented Ambiguous Question Answering. arXiv:2409.02361 [cs.CL] https://arxiv.org/abs/2409.02361

Ioannis Krasakis, Mohammad Aliannejadi, Hamed Zamani, and Fabio Crestani. 2021. What makes a clarifying question useful?. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2410–2414.

Shang-Chieh Lin, Jheng-Hong Yang, Chuan-Ju Wang Chen, Hung-yi Lee, and Yun-Nung Chen. 2020. A Systematic Study and Comprehensive Evaluation of Conversational Query Rewriting. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1877–1880.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering Ambiguous Open-domain Questions. In *Proceedings of the 2020*

Conference on Empirical Methods in Natural Language Processing (EMNLP). 5783–5797.

Fanglong Mo, Yutao Zhang, Yifei Zhang, Junkai Wang, Peng Zhang, Yiding Sun, Qi Zhang, and Jing Xuan. 2023. ConvGQR: Generative query reformulation for conversational search. *arXiv preprint arXiv:2305.10138* (2023).

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2023. ASQA: Factoid Questions Meet Long-Form Answers. arXiv:2204.06092 [cs.CL] https://arxiv.org/abs/2204.06092

Hongjin Wan, Sung-Hwan Lee, Dong-Kyu Lee, Hansaem Kim, Minki Jang, Yun-Tae Kim, and Sang-Wook Lee. 2025. DIVA: A Diversify-Verify-Adapt Framework for Open-domain Ambiguous Question Answering. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

Thomas Wasow. 2015. *Ambiguity Avoidance is Overrated.* 29–46. doi:10.1515/9783110403589-003

Zeqiu Wu, Zong He, and Mari Ostendorf. 2022. CONQRR: Conversational query rewriting for retrieval with reinforcement learning. *arXiv preprint arXiv:2204.07672* (2022).

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *arXiv preprint arXiv:2305.10601* (2023).

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations.*

Hamed Zamani, Nick Craswell, Michael Bendersky, and Bhaskar Mitra. 2020a. Generating clarifying questions for open-domain information-seeking conversations. In *Proceedings of The Web Conference 2020.* 181–191.

Hamed Zamani, Bhaskar Mitra, Nick Craswell, Michael Bendersky, and Xuanhui Li. 2020b. Analyzing and learning from user interactions with clarifying questions. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval.* 135–144.

Richard Zanibbi and Dorothea Blostein. 2012. Recognition and retrieval of mathematical expressions. *Int. J. Doc. Anal. Recognit.* 15, 4 (Dec. 2012), 331–357. doi:10.1007/s10032-011-0174-4

Richard Zanibbi, Behrooz Mansouri, and Anurag Agarwal. 2025. Mathematical Information Retrieval. *Foundations and Trends in Information Retrieval* (2025).

A Appendix

A.1 Algorithm

The execution flow of the Agentic Disambiguation Framework is formalized in Algorithm 1.

Algorithm 1 Agentic Disambiguation Framework

```
Require: Query $q$, Corpus $C$, Thresholds $\tau$
Ensure: Comprehensive Answer $a$
$\mathcal{D}_{\text{init}} \leftarrow \text{Retrieve}(q, C, 2k)$
$V \leftarrow \text{Encode}(\mathcal{D}_{\text{init}})$
$\sigma^2_{\text{var}}, \sigma_{\text{sep}} \leftarrow \text{ComputeCoherence}(V)$
$S_q \leftarrow \text{ClassifyState}(\sigma^2_{\text{var}}, \sigma_{\text{sep}})$
if $S_q = \text{Unambiguous}$ then
    return $\text{Generate}(q, \text{Retrieve}(q, C, k))$
end if
$C \leftarrow \text{KMeans}(V, k_{\text{adaptive}})$
$Q_{sub} \leftarrow \emptyset$
for $C_j \in C$ do                                        ▷ Decomposition Phase
    if $\cos(q, \text{centroid}(C_j)) \geq \tau_{rel}$ then
        $q_j \leftarrow \text{SummarizeIntent}(q, C_j)$
        $\hat{d}_j \leftarrow \text{HyDE}(q_j)$
        $Q_{sub} \leftarrow Q_{sub} \cup \{(q_j, \hat{d}_j)\}$
    end if
end for
$\mathcal{D}_{\text{pool}} \leftarrow \emptyset$
for $(q_j, \hat{d}_j) \in Q_{sub}$ do                         ▷ Dual Retrieval Phase
    $\mathcal{D}_{\text{pool}} \leftarrow \mathcal{D}_{\text{pool}} \cup \text{Retrieve}(q_j) \cup \text{Retrieve}(\hat{d}_j)$
end for
$\mathcal{D}_{\text{final}} \leftarrow \text{RRF}(\mathcal{D}_{\text{pool}})$
return $\text{StructuredSynthesis}(q, \mathcal{D}_{\text{final}})$
```

A.2 Local LLM Results

The results for the Local LLM `qwen3-4b-q4` run on a mobile RTX 3050Ti GPU on the ASQA dataset.

Table 4: ASQA results using local LLM `qwen3-4b-q4`.

Approach	D-F1	ROUGE-L	DR-F1	nDCG@5	Latency	Tok/Q
Sparse (BM25)						
Vanilla	0.256	0.198	0.155	0.476	104.8	1030
Agentic	0.204	0.201	0.131	0.372	176.2	1317
Dense (FAISS)						
Vanilla	0.188	0.189	0.114	0.196	101.5	969
Agentic	0.178	0.201	0.114	0.178	92.9	1251
Hybrid (RRF)						
Vanilla	0.247	0.196	0.153	0.348	104.4	999
Agentic	0.224	0.208	0.145	0.442	136.5	1302

A.3 Ablation Study

To isolate the impact of the ambiguity classification module on the peformance of the agentic framework, we perform an ablation in which the semantic coherence analysis step is removed. Table 5 reports the resulting performance metrics.

Table 5: AmbigNQ Results Without Ambiguity Classification

Approach	F1	D-F1	nDCG@5	Latency	Tok/Q
Sparse (BM25)					
Agentic	0.573	0.398	0.500	6.717	968.49
Dense (FAISS)					
Agentic	0.4795	0.327	0.279	3.638	901.94
Hybrid (RRF)					
Agentic	0.590	0.391	0.535	8.075	956.91

A.4 AmbigNQ Results Using Question Based Ambiguity Classification

The agentic framework was evaluated using a DistilBERT question-level ambiguity classifier. The results in Table 6 show how the system performs when ambiguity detection relies on a lightweight, query-only prediction rather than retrieval-based coherence metrics

Table 6: AmbigNQ Results Using Question Based Ambiguity Classification(DistilBERT)

Approach	F1	D-F1	nDCG@5	Latency	Tok/Q
Sparse (BM25)					
Agentic	0.564	0.386	0.483	2.937	879
Dense (FAISS)					
Agentic	0.482	0.316	0.250	2.754	810
Hybrid (RRF)					
Agentic	0.536	0.372	0.425	2.98	850