BMGQ: A Bottom-up Method for Generating Complex Multi-hop Reasoning Questions from Semi-structured Data

Bingsen Qiu, Zijian Liu, Xiao Liu, Haoshen Yang, Zeren Gao, Bingjie Wang, Feier Zhang, Yixuan Qin, Chunyan Li

ByteDance DMC*

{qiubingsen.123, liuzijian.109, liuxiao.0423, yanghaoshen, wangbingjie, lichunyan.ivy}@jiyunhudong.com

{zhangfeier, qinyixuan.1}@bytedance.com

October 29, 2025

Abstract

Building training-ready multi-hop question answering (QA) datasets that truly stress a model's retrieval and reasoning abilities remains highly challenging recently. While there have been a few recent evaluation datasets that capture the characteristics of hard-to-search but easy-to-verify problems—requiring the integration of ambiguous, indirect, and cross-domain cues—these data resources remain scarce and are mostly designed for evaluation, making them unsuitable for supervised fine-tuning (SFT) or reinforcement learning (RL). Meanwhile, manually curating non-trivially retrievable questions—where answers cannot be found through a single direct query but instead require multi-hop reasoning over oblique and loosely connected evidence—incurs prohibitive human costs and fails to scale, creating a critical data bottleneck for training high-capability retrieval-and-reasoning agents.

To address this, we present an automated framework for generating high-difficulty, training-ready multi-hop questions from semi-structured knowledge sources. The system (i) grows diverse, logically labeled evidence clusters through Natural Language Inference (NLI)-based relation typing and diversity-aware expansion; (ii) applies reverse question construction to compose oblique cues so that isolated signals are underinformative but their combination uniquely identifies the target entity; and (iii) enforces quality with a two-step evaluation pipeline that combines multi-model consensus filtering with structured constraint decomposition and evidence-based matching. The result is a scalable process that yields complex, retrieval-resistant yet verifiable questions suitable for SFT/RL training as well as challenging evaluation—substantially reducing human curation effort while preserving the difficulty profile of strong evaluation benchmarks.

1 Introduction

Large language models (LLMs) have achieved remarkable progress across many natural language processing tasks, yet their ability to conduct deep research involving multi-hop retrieval and reasoning remains limited. This capability is crucial for solving problems where answers cannot be found through a single lookup but require integrating multiple pieces of indirect, cross-domain evidence. While existing multi-hop QA training datasets provide useful testbeds, most of them rely on relatively shallow reasoning chains, offering limited value for training models that can handle genuinely complex retrieval and reasoning tasks.

In response to this gap, a new line of evaluation benchmarks has emerged, specifically targeting challenging deep reasoning scenarios. These datasets are intentionally designed to be hard to search and easy to verify: individual clues are vague and insufficient to reach the answer, but when combined, they collapse the search space to a single, verifiable entity. While such benchmarks effectively capture the difficulty profile required to improve LLM reasoning capabilities, they are expensive to construct and cannot be used for large-scale SFT or RL.

To overcome these scalability limitations, recent work has explored automatic question generation pipelines. These approaches demonstrate the feasibility of synthesizing multi-hop questions at scale, but most remain limited in depth: they often involve only a few reasoning steps, lack complex cross-domain structures, and do not enforce strict verification of answer uniqueness. As a result, the generated questions are still far from the difficulty and reliability required for training powerful reasoning agents.

Motivated by these gaps, we introduce an automated framework for constructing high-difficulty multi-hop datasets suitable for both training and evaluation. Our approach transforms semi-structured knowledge sources into structured evidence clusters, builds diverse and logically coherent evidence clusters, and employs a reverse

*ByteDance Data and Model Service Center, Data Exploration Team

construction strategy that composes indirect clues to ensure that the answer is uniquely identifiable only when all clues are considered together. Finally, a dedicated quality evaluation system combines multi-model consensus filtering with structured constraint decomposition and evidence-based matching, ensuring that each retained question is both challenging and verifiably correct.

By combining the difficulty profile of high-end evaluation benchmarks with the scalability of automated synthesis, our framework enables the creation of large, training-ready multi-hop datasets that are capable of truly stressing and improving the deep retrieval and reasoning abilities of large language models.

2 Related Works

2.1 Early Benchmarks: Progress and Limitations

A series of early multi-hop QA benchmarks laid the foundation for research on multi-hop reasoning. Among them, several datasets progressively expanded the scope of reasoning—from basic two-hop evidence combination to more compositional chains and structured knowledge integration—marking an important trajectory in the evolution of the field. However, despite these advances, they remain fundamentally limited in reasoning depth, retrieval difficulty, and answer uniqueness, making them insufficient for training or evaluating models designed for complex, open-domain reasoning.

HotpotQA [1] represents a seminal milestone, introducing large-scale multi-document QA with annotated supporting sentences to encourage explicit reasoning. While it successfully established the importance of evidence combination, most of its questions involve only two reasoning hops and can often be solved through shallow retrieval heuristics or surface-level lexical matching, rather than genuine multi-step inference.

Building on this foundation, ComplexWebQuestions [2] sought to increase reasoning complexity by automatically generating compositional questions from knowledge base paths. This approach broadened the scope beyond simple paragraph linking and introduced more structured multi-hop reasoning. However, its template-based construction limited linguistic variability, and the early versions suffered from train-test leakage, ultimately constraining its diagnostic power.

MuSiQue [3] further advanced this line by deliberately composing single-hop facts into multi-hop chains and introducing mechanisms to reduce spurious shortcuts. Compared to its predecessors, it provides richer compositional structures and greater control over reasoning paths. Yet, most questions still require relatively short reasoning chains, and the dataset does not explicitly enforce retrieval difficulty or the "hard-to-search but easy-to-verify" property crucial for deep research scenarios.

Other datasets, such as QASC [4] and 2WikiMultiHopQA [5], extend this trend into specific domains or structured-unstructured hybrids. QASC focuses on elementary science reasoning through sentence combination, while 2Wiki combines Wikidata triples with Wikipedia passages to provide explicit reasoning path annotations. Though each brings incremental improvements, their domain restrictions, template-driven construction, and reliance on structured or encyclopedic sources limit their ability to model realistic open-domain retrieval settings.

Overall, while these benchmarks were instrumental in shaping the multi-hop QA landscape, they largely emphasize shallow reasoning, short-hop paths, and constrained domains. As a result, they fall short of capturing the full complexity, ambiguity, and uniqueness constraints that characterize real deep-research multi-hop retrieval and reasoning tasks.

2.2 BrowseComp: A Shift Toward "Hard-to-Search, Easy-to-Verify"

The BrowseComp dataset [6] represents a significant step forward toward realistic multi-hop evaluation. Unlike earlier benchmarks, BrowseComp is designed around the principle of "hard to search, easy to verify." Individually, these clues are too vague to retrieve the answer directly, but only by jointly reasoning over multiple pieces of information can the search space be narrowed to a single correct answer.

BrowseComp questions exhibit several distinctive characteristics:

- Deep and wide reasoning chains: Solving them often requires traversing multiple knowledge domains and integrating scattered evidence across several web pages.
- Fuzzy and indirect clues: Signals are deliberately vague or partial, demanding reasoning over context rather than direct lookup.
- Hard-to-search but easy-to-verify: While the answer is difficult to retrieve directly, once found, it is short, concrete, and objectively verifiable.
- Bias toward niche domains: Many answers are rare entities or domain-specific facts, discouraging memorization and forcing deeper retrieval strategies.

Building on this idea, BrowseComp-ZH [7] extends the framework to the Chinese web domain, aiming to benchmark multilingual web-browsing capabilities of large language models. However, our internal manual evaluation of both BrowseComp and BrowseComp-ZH reveals that a substantial portion of their questions lack strict answer uniqueness or contain ambiguities in clue interpretation. In many cases, multiple plausible answers can satisfy the same set of fuzzy constraints, highlighting the difficulty of consistently encoding high reasoning complexity while ensuring unambiguous ground truths.

This observation underscores a crucial gap in the current landscape of multi-hop datasets. Both BrowseComp and BrowseComp-ZH were intentionally designed as evaluation benchmarks, not training corpora, aiming to push the limits of retrieval and reasoning capabilities of large language models. While they effectively capture realistic reasoning difficulty, our manual inspection further reveals that many questions exhibit imperfect answer uniqueness, making them less suitable for direct use in training scenarios where strict supervision is required. This highlights the need for automated construction pipelines with explicit quality control that can generate large-scale datasets matching the difficulty of BrowseComp while ensuring unambiguous and verifiable ground truths for model training.

2.3 Automatic Construction of Multi-Hop Datasets

As the demand for large-scale, high-quality multi-hop datasets grows, researchers have explored various automatic construction frameworks to alleviate the prohibitive costs of manual curation. These approaches aim to generate questions and reasoning chains directly from structured knowledge bases, unstructured text, or a combination of both. However, existing methods remain limited in scope and fall short of replicating the depth, ambiguity, and reasoning complexity seen in challenging datasets such as BrowseComp[6].

Template- and Knowledge-Graph-Based Methods. Early efforts in automatic dataset construction relied heavily on rule-based templates and knowledge graph traversal. For instance, ComplexWebQuestions [2] generated compositional questions by expanding seed questions in WebQuestionsSP with additional relations from Freebase. This approach enabled scalable generation but produced synthetic and predictable question structures, with reasoning typically confined to simple relation chaining. Similarly, 2WikiMultiHopQA [5] constructed multi-hop questions by linking Wikidata triples and retrieving supporting passages from Wikipedia, but its reliance on predefined templates and explicit relation paths limited linguistic diversity and often failed to capture the subtle reasoning required in real-world retrieval. While these methods demonstrated the feasibility of automatic construction, they struggled with several key challenges:

- Limited reasoning diversity: Most generated questions followed fixed reasoning patterns (e.g., bridge or comparison), leading to repetitive question structures.
- Lack of ambiguity and indirectness: Template generation typically results in explicit, easily searchable questions, lacking the fuzzy clues and oblique phrasing that characterize harder multi-hop tasks.
- Weak retrieval difficulty: Because many questions can be answered by direct lookup of a single knowledge base triple, they fail to replicate the retrieval complexity of open-domain scenarios.

HopWeaver: Toward Automated Multi-Hop Question Generation. A more recent and influential attempt at automatic dataset construction is HopWeaver [8], which introduces a retrieval-augmented generation framework to synthesize multi-hop questions. HopWeaver systematically constructs bridge-type and comparison-type questions by identifying semantically related entity pairs and retrieving intermediate evidence passages from Wikipedia. It then uses large language models to compose natural-language questions that integrate information across multiple hops. This work represents a substantial advance beyond purely template-based approaches, as it leverages both retrieval signals and generative models to produce more natural and varied questions. However, despite these improvements, HopWeaver still exhibits several critical limitations:

- Restricted reasoning types: The framework focuses primarily on bridge and comparison questions, leaving out many reasoning forms involving casual, compositional, or conditional relations.
- Shallow reasoning depth: Most generated questions remain within two to three hops, lacking the depth and cross-domain traversal seen in BrowseComp-style problems.
- Insufficient ambiguity: HopWeaver does not explicitly enforce fuzzy clue construction or the "hard-to-search but easy-to-verify" principle, resulting in questions that are still vulnerable to direct retrieval.
- Limited control over answer uniqueness: The framework does not guarantee that a question points uniquely to a single answer without external disambiguation, which is critical for training and evaluating reasoning models.

In summary, existing automatic construction frameworks — from template-based systems to retrieval-augmented generation like HopWeaver — represent important milestones in scaling multi-hop dataset creation. Yet they remain insufficient for producing BrowseComp-level questions that combine deep reasoning chains, ambiguous and indirect clues, domain-crossing knowledge transitions, and strict answer uniqueness. Bridging this gap requires a fundamentally new approach that integrates structured evidence graph construction, bottom-up question synthesis, and rigorous quality evaluation — capabilities that motivate the framework proposed in this work.

2.4 Dataset Quality Control in Automatic Generation

While automatic generation greatly reduces the cost of dataset construction, it also introduces significant challenges in ensuring the correctness, uniqueness, and reasoning validity of the resulting question–answer pairs. Poorly controlled generation often leads to issues such as ambiguous questions, incorrect answers, shallow reasoning chains, or spurious shortcuts that undermine the value of the dataset. As a result, effective post-generation quality control is essential for building reliable multi-hop datasets.

Rule-Based and Discriminative Filtering Approaches. Early approaches to quality assurance primarily relied on rule-based heuristics or discriminative models applied after question generation. For instance, ComplexWebQuestions employed rule-based checks to discard syntactically malformed questions or those with mismatched entities. Other systems used lightweight classifiers to verify whether retrieved documents supported the predicted answers.

LLM-Based Evaluation and Self-Verification. Recent work has explored leveraging large language models themselves as evaluators, introducing more semantic and reasoning-aware quality control mechanisms. SelfCheckGPT [9] uses a model to assess the factual consistency of its own generations by generating multiple paraphrases and measuring agreement. Similarly, LLM-as-a-Judge approaches [10] use language models as evaluators to assess factuality, reasoning soundness, or answer correctness without explicit labels. These methods represent a step forward in capturing semantic errors and deeper reasoning flaws beyond surface matching.

However, most existing LLM-based evaluation frameworks are model-centric and rely on a single model's judgment, which introduces bias and lacks robustness. They often do not incorporate structured representations of the question's internal logic, nor do they enforce explicit constraints on time, location, or entity properties. As a result, they cannot fully guarantee that a question is logically coherent, uniquely solvable, and well-aligned with available evidence.

To address these limitations, we propose a structured Data Quality Evaluation System that integrates multi-model consensus with constraint-based verification. Rather than treating quality control as a single-step classification, our approach first filters ambiguous questions through model agreement and then verifies candidate answers via decomposed, evidence-grounded predicates, ensuring logical soundness and answer uniqueness.

Therefore, against this backdrop, our work contributes a novel pipeline for automatic multi-hop dataset construction that explicitly addresses these gaps. We design a methodology capable of producing BrowseComp-level questions at scale for training purposes, and we augment it with a Data Quality Evaluation System that combines multi-model consensus, structured clue decomposition, and evidence-based verification. This integration ensures that the resulting dataset is not only large and diverse, but also auditable, logically coherent, and uniquely solvable, laying the groundwork for advancing both training and evaluation of reasoning-capable large language models.

3 Methodology

Our methodology for constructing a multi-hop reasoning dataset follows a structured four-stage pipeline, as illustrated in Figure 1. In **Part 1 (Data Sources & Adaptation)**, raw Wikipedia and Wikidata data are adapted into a lightweight relational database and pre-constructed evidence network. In **Part 2 (Node Information Construction)**, we retrieve raw page content, perform text preprocessing, remove non-entity evidence, and link candidate entities with their supporting paragraphs. In **Part 3 (Evidence Chain Construction)**, relation classification guarantees the validity of edges, diversity-aware evaluation mitigates semantic redundancy, and breadth-first expansion yields a balanced and interpretable evidence cluster. Finally, in **Part 4 (Question Construction & Optimization)**, a bottom-up reverse generation strategy constructs the initial question, followed by multi-round optimization to enhance abstraction, difficulty, and uniqueness. Together, these stages enable the automatic generation of high-quality, logically coherent, and challenging multi-hop questions grounded in real-world knowledge.

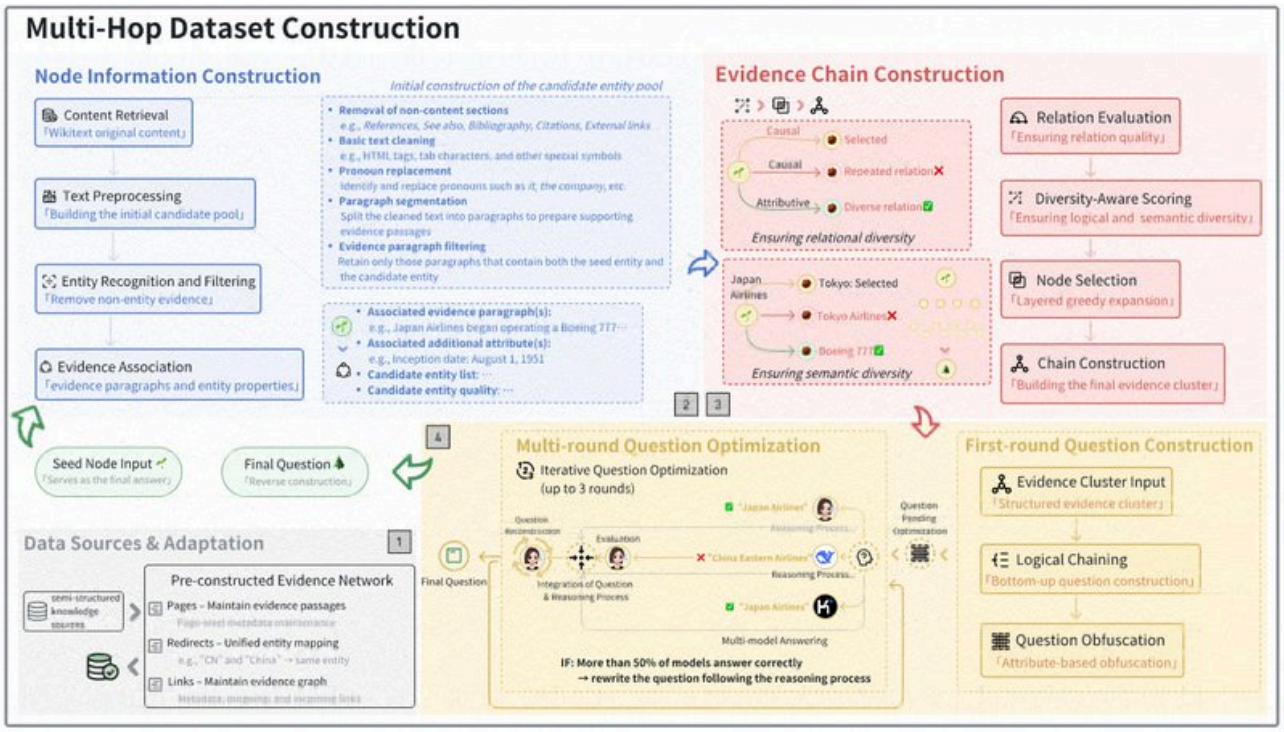


Figure 1: Multi-hop dataset construction framework of our system. The construction process includes: (1) **Data Sources & Adaptation**, (2) **Node Information Construction**, (3) **Evidence Chain Construction**, and (4) **Question Construction & Optimization**. Colored blocks correspond to each stage, with step indices annotated for cross-reference in the main text.

3.1 Data Source & Adaptation

The construction of our dataset is distributed as large-scale SQL files. While comprehensive, this semi-structured format is not directly suitable for efficient graph queries or multi-hop reasoning. To address this issue, we preprocess the raw data and transform it into a lightweight, high-performance relational database. This design ensures structured organization, millisecond-level query efficiency, and a robust foundation for subsequent reasoning tasks.

3.2 Node Information Construction

Once the structured database has been established, the next step is to construct an informative contextual environment for a given seed entity. This process aims to identify high-quality candidate entities and their corresponding evidence passages, which serve as the foundation for multi-hop evidence chain construction.

The key challenge arises from the fact that a single Wikipedia page may contain hundreds or even thousands of outbound links, many of which correspond to generic terms, abstract concepts, or weakly related references. Blindly expanding all such links leads to **semantic drift** [11][12] — a phenomenon where the reasoning path gradually loses topical or contextual relevance, resulting in misleading or meaningless connections. To mitigate this issue, we introduce a multi-stage pipeline that integrates preprocessing, entity recognition, and evidence alignment. An annotated example illustrating the prevalence of irrelevant or weakly connected links in raw Wikipedia pages is provided in Appendix A.

3.2.1 Goals and Challenges

- **Objective**: Extract a set of candidate entities (outlinked Wikipedia pages) and their supporting evidence passages from the seed entity's page.
- **Challenge**: Avoid semantic drift by filtering out irrelevant, overly abstract, or weakly connected terms (e.g., privatised, capital, tons), which otherwise disrupt logical consistency across reasoning chains.

For instance, in the entry *Japan Airlines*, some outbound links correspond to irrelevant common nouns (e.g., *mail*, *capital*), whereas others correspond to meaningful entities (e.g., *All Nippon Airways*, *Tokyo*). A full annotated example is provided in Appendix A.

3.2.2 Processing Pipeline

The Node Information Construction stage (Figure 1 Part 2) builds a structured, semantically coherent neighborhood around each seed entity to support subsequent evidence chain expansion. This process involves lightweight text preprocessing followed by a core filtering step based on named entity recognition (NER).

1. **Text Preprocessing and Candidate Extraction**

   Given a seed entity, we retrieve its webpage text content and remove non-essential sections (e.g., "See also", "References") and markup artifacts. The cleaned text is segmented into paragraphs, and all internal hyperlinks are extracted as initial candidate entities. These links preserve webpage's rich contextual connectivity while keeping the preprocessing lightweight.

2. **NER-Based Candidate Filtering**

   The cornerstone of this stage is robust entity filtering. After evaluating multiple approaches—including rule-based heuristics and statistical sequence models, which are still commonly used as baselines in recent NER research [13][14][15]—we adopt a transformer-based BERT NER model [16](dslim/bert-large-NER [17]) due to its superior contextual discrimination.

   The model reliably separates valid entities from noisy or abstract terms, significantly reducing semantic drift.

   Example 1

   - **Accepted**: *Albert Einstein* (Person), *Pacific Ocean* (Location), *Bytedance Inc* (Organization), *World War II* (Event).
   - **Rejected**: *Philosophy* (abstract concept), *Meditation* (general term), *List of countries* (meta-page), *Ocean* (overly broad category).

   Through a combination of preprocessing, NER-based filtering, and evidence alignment, we construct a high-quality contextual environment around each seed entity. This design minimizes semantic drift and ensures that only semantically grounded, evidence-supported entities progress to the next stage of evidence chain construction.

3. **Evidence Association and Context Refinement**

   For each retained entity, we retrieve its originating paragraph to guarantee textual co-occurrence with the seed entity. We then apply coreference resolution to replace ambiguous mentions with explicit names and optionally enrich entities with structured attributes from Wikidata (e.g., inception date, headquarters, affiliations).

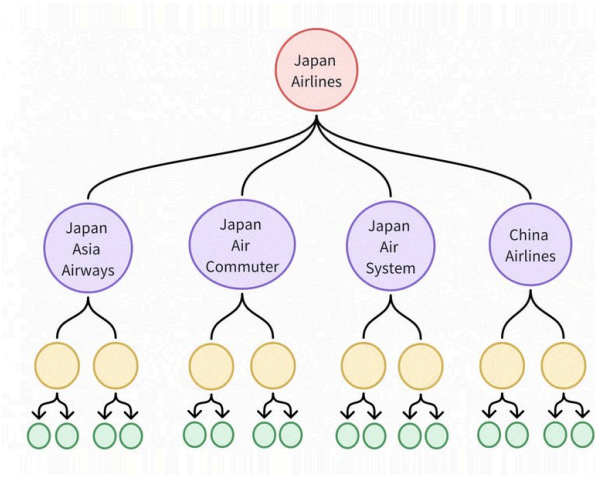
   This selective filtering pipeline yields a clean, semantically grounded candidate set and associated evidence passages, minimizing semantic drift and ensuring logical coherence for downstream evidence chain construction.

3.3 Evidence Chain Construction

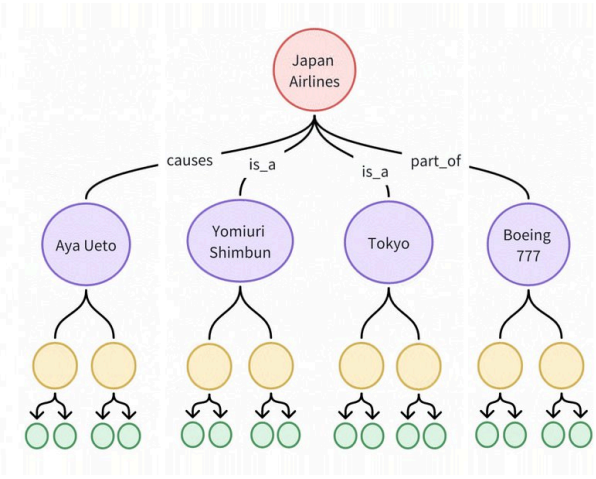
After generating high-quality candidate entities and evidence passages in Section 3.2, the next stage is to construct evidence chains—multi-hop paths that connect entities through explicit, logically interpretable relationships. The goal is to move beyond surface-level similarity and build reasoning paths that mimic human-style exploration, ensuring both semantic diversity and logical rigor.

The central challenge in constructing evidence chains lies in ensuring **logical connectivity** rather than mere topical similarity. Pure similarity-based methods risk producing **semantic homogeneity**, where expansions remain trapped in a single domain. For example, starting from Japan Airlines, semantic similarity expansion tends to generate only other airlines, resulting in horizontally enumerated entities without explanatory depth. Such graphs fail to uncover meaningful relations (e.g., usage of aircraft models, regulatory reforms, or cultural connections) and thus cannot support multi-hop reasoning.

To overcome this limitation, our approach incorporates **relation classification** and **diversity constraints**, ensuring that expansions move beyond surface similarity. With these mechanisms, the evidence cluster evolves into a semantically diverse and logically coherent graph, where entities span multiple categories such as people, locations, organizations, and cultural events. A comparative visualization of the semantic-only graph and the relation-augmented graph is provided in Figure 2.



(a) Evidence graph generated by semantic similarity expansion from "Japan Airlines." The graph shows severe semantic homogeneity, with all expansions remaining within the airline domain and lacking logical diversity.



(b) Evidence graph generated from the same seed ("Japan Airlines") after incorporating NLI-based relation classification, logical diversity, and semantic diversity constraints.

Figure 2: A comparative visualization of the semantic-only graph and the relation-augmented graph.

3.3.1 From Similarity to NLI-Based Relations

Early experiments with vector embeddings (e.g., BAAI BGE-M3 [18]) encoded entity descriptions and selected expansions by cosine similarity. While effective in grouping thematically related entities, this approach lacked explanatory depth and failed to produce multi-hop reasoning paths.

To overcome this, we adopt a **Natural Language Inference (NLI)** [19] framework. Instead of asking whether two entities are similar, we ask whether an evidence passage entails a hypothesized relation between them. Using the *facebook/bart-large-mnli* [20] model in a zero-shot setting, relation classification is reframed as an entailment task:

- **Premise (P)**: A sentence from webpage containing both entities.
- **Hypothesis (H)**: A templated statement describing a candidate relation.

If the model predicts that P entails H, then the relation is accepted, and the entailment probability serves as its confidence score. Relations are categorized into six logical types, each with predefined hypothesis templates, as shown in Table 1 and an example described in Example 2.

Example 2: Japan Airline - Shingo Katori

Consider the relation between *Japan Airlines* and *Shingo Katori* (a Japanese actor and member of SMAP).

- **Premise**: "At the end of 2005, Japan Airlines began using a Boeing 777 (JA8941) featuring Japanese actor Shingo Katori on one side, and the television series Saiyūki on the other."
- **Hypotheses**: Generated across all relation templates, in both directions (U→V and V→U).
- **Result**: The NLI model predicted highest confidence (0.92) for "Shingo Katori is an attribute of Japan Airlines."
- **Decision**: The relation is classified as has_attribute, direction = backward, confidence = 0.92.

This edge no longer represents vague similarity, but a **precise semantic link**: Japan Airlines (via aircraft livery) was associated with Shingo Katori as a cultural symbol.

3.3.2 Graph Expansion: Controlled Breadth-First Growth

The construction of evidence chains proceeds through a **controlled breadth-first expansion** [21] strategy. Unlike depth-only approaches that risk producing monotonous, semantically narrow chains, our method ensures that expansions are both logically valid and semantically diverse. The process integrates relation judgment, diversity-aware scoring, and greedy node selection into a unified workflow.

Table 1: Predefined Relation Types Used for NLI-Based Classification

Relation Type	Description	Example Template
`causes`	Causal relation	"U causes V" "U leads to V" "U induces V"
`part_of`	Compositional	"U is part of V" "U belongs to V" "U is a component of V"
`is_a`	Taxonomic	"U is a kind of V" "U is a type of V" "U is an instance of V"
`has_attribute`	Attributive	"U has attribute V" "U has property V" "U is characterized by V"
`requires`	Conditional	"U requires V" "U needs V" "U depends on V"
`used_for`	Functional	"U is used for V" "U is used to access V" "U serves the purpose of V"

Figure 1 Part 3 provides a schematic overview of this pipeline, showing how relation classification guarantees the validity of edges, diversity evaluation prevents redundancy, and breadth-first expansion yields a balanced and interpretable evidence cluster.

The algorithm operates layer by layer, starting from a seed node and expanding outward according to a user-specified strategy (e.g., $[4, 2, 2]$, meaning four nodes at depth 1, two per node at depth 2, and so on). At each step, three main components govern the expansion:

1. **Candidate Sampling**
   - Remove nodes already present in the graph to avoid cycles.
   - Perform frequency-based sampling: 70% high-frequency entities from the source text, combined with 30% randomly chosen candidates to preserve exploration.
2. **Relation Evaluation**
   - Retrieve evidence sentences containing both parent and candidate entities.
   - Apply NLI-based relation classification (see Section 3.2) to assign a relation type, direction, and confidence score.
   - Discard edges below the confidence threshold (e.g., 0.45).
3. **Diversity-Aware Scoring**
   Each candidate is assigned a composite score that integrates logical and semantic considerations:

$$\begin{aligned} \text{Score} = \alpha \cdot \text{NLI Confidence} \\ + \beta \cdot \text{Relation Diversity} \\ + \gamma \cdot \text{Semantic Diversity} \end{aligned} \tag{1}$$

   - *Relation diversity:* Penalize repeated relation types within the same layer.
   - *Semantic diversity:* Penalize nodes whose titles are too similar to already selected nodes.
   - *Paragraph diversity:* Favor nodes extracted from different sections of the page, ensuring broader contextual coverage.
4. **Greedy Selection and Expansion**
   - Rank all candidates by their composite scores.

- Select the top-K candidates as the new child nodes.
- Add these nodes and their corresponding edges (including evidence passage, relation type, direction, and confidence) into the graph.

The final result of this expansion process is a **graph object** that records the layered evidence cluster in a structured form. Each node stores standardized entity information (`ID`, `title`, `type`), while each edge stores its associated `evidence passage`, `relation type`, `direction`, and `confidence score`. The entire graph is serialized into a JSON format (`GraphData`), making it both human-interpretable and machine-readable.

For example, starting from the seed entity *Japan Airlines*, the expansion yields a multi-layered graph that links not only to other airlines, but also to related people (e.g., Shingo Katori), organizations (e.g., SMAP), locations (e.g., Tokyo, Sapporo), and cultural events (e.g., Kōhaku Uta Gassen). In addition, the system enriches the seed with discriminative attributes—such as founding year (1951), hub (Osaka–Kansai), and alliance membership (Oneworld)—so that these attribute cues alone can uniquely identify the correct seed entity (Japan Airlines). This demonstrates that the system avoids semantic homogeneity, instead producing logically coherent and semantically diverse chains that serve as high-quality raw material for downstream multi-hop question generation.

3.4 Final Question Construction

The final stage of our pipeline transforms the evidence clusters generated in Step 3 into **complex multi-hop questions**. The input consists of the **evidence cluster** (nodes, edges, and supporting passages) and the *seed* entity (the answer). Unlike naive question generation, which often yields factoid-style queries easily solved by direct retrieval, our method employs a **bottom-up, reverse reasoning strategy**. This ensures that the constructed question faithfully reproduces the reasoning path encoded in the evidence cluster.

3.4.1 Reverse Question Generation

We adopt a bottom-up approach that begins from the **leaf nodes** of the evidence cluster and progressively backtracks to the seed node. At each step, the model is prompted to convert local evidence into a **descriptive but oblique clue**, which avoids explicitly mentioning the seed or its immediate neighbors. These clues are then recursively nested to form a **composite reasoning chain**, culminating in a single question that uniquely points to the *seed*.

To guarantee difficulty and uniqueness, the generation process enforces several constraints:

- Each question must integrate at least n deep cues, where n is a configurable parameter depending on the project setting. Here, *depth* is defined as the number of edges between the seed entity (the answer node) and the evidence node in the underlying evidence graph. This ensures that questions require reasoning beyond shallow retrieval.
- Preference is given to cues drawn from deeper nodes rather than immediate neighbors ($depth = 1$), reducing the likelihood of trivial single-hop retrieval.
- Descriptions favor **category–relation–event** expressions over surface names (e.g., "a major East Asian carrier" instead of "Japan Airlines").

3.4.2 Question Obfuscation

Once an initial draft is generated, the question undergoes a controlled obfuscation process. Explicit anchors such as exact years, city names, or institutional titles are generalized into higher-level descriptors (e.g., "in the early 21st century" instead of "2003"; "a North American diplomatic authority" instead of "the U.S. Secretary of State"). Similarly, personal names are replaced by roles or contextualized identifiers (e.g., "an artist who specialized in landscape paintings featuring misty and rainy scenes" instead of the exact name). This abstraction raises the difficulty of direct retrieval while preserving solvability.

3.4.3 Iterative Refinement

To further ensure robustness, the system performs automatic question optimization. After optimization, multiple large language models attempt to answer the constructed question. If a majority of models successfully identify the seed entity, the system triggers a **rewriting procedure** that systematically increases the reasoning difficulty while preserving answer uniqueness.

This rewriting procedure is grounded in the underlying evidence graph. First, the system selects a minimal supporting subgraph containing multiple deep cues (i.e., nodes located several hops away from the seed, beyond the most immediate neighbors). To prevent shallow retrieval, anchor terms and "killer pairs" (e.g., award +

year, model + maker) are softened or replaced with implicit formulations. Rather than directly naming entities, the system rephrases them using higher-level categories, relations, and event descriptors, ensuring that no direct level-1 node or alias appears in the surface text.

The rewriting process follows a structured pipeline:

1. Graph parsing and subgraph selection (core axis + deep cues);
2. Text generation using implicit and oblique formulations;
3. Self-verification, including checks on the number and depth of cues, logical coherence, strict uniqueness, alias avoidance, and length constraints;
4. If any constraint fails, the system rolls back and regenerates the question.

Through this loop, the discriminative properties required to uniquely identify the seed are preserved, but the search complexity is progressively increased. This ensures that even as the question becomes harder, the seed remains the only correct answer. The loop terminates once the question reaches the target difficulty level or the maximum number of refinement rounds.

4 Data Quality Evaluation System

To ensure that automatically constructed questions are not only of intended difficulty, but also solvable and unique, we design a Data Quality Evaluation System that serves as a filtering layer between question generation and final dataset inclusion. It consists of two complementary components: a **graph-based textual structure** (Section 4.1) that provides early structural screening based on linguistic and reasoning features, and a **data quality evaluation workflow** (Section 4.2) that conducts constraint decomposition and evidence-based validation. Together, these components ensure that only structurally coherent, sufficiently complex, and verifiably unique questions enter the final dataset.

4.1 Graph-Based Textual Structure

To systematically assess the quality of constructed questions before formal evaluation, we introduce a graph-based textual structure representation. This representation extracts the **subject**, **object**, and **attribute** elements embedded in the constructed question and links them using a predefined set of six linguistic relations (section 3.3.1): *is_a*, *part_of*, *has_attribute*, *causes*, *requires*, and *used_for*. By converting the textual prompt into a structured graph, we make the underlying reasoning chain explicit, auditable, and easily visualizable.

We place this graph-structuring step immediately after question construction but before the formal Data Quality Evaluation (Section 4.2). This allows us to conduct early-stage structural screening: questions that fail to form coherent and solvable graphs can be automatically discarded before entering the more computationally expensive verification stage. Only structurally valid and potentially high-quality questions flow into the evaluation pipeline.

To assess graph quality, we compute four key structural indicators:

1. **Orphan nodes** — Whether the graph contains nodes that are not connected to the main reasoning chain.
   - Criterion: OrphanNodes = 0 guarantees basic solvability.
2. **Attribute count (T_a)** — The number of attribute nodes attached to the core graph.
   - Criterion: $T_a \geq \alpha$, where α is a project-specific threshold. A larger T_a indicates richer semantic specificity, which strengthens answer uniqueness.
3. **Edge count (T_e)** — The total number of edges in the graph.
   - Criterion: $T_e \geq \beta$, where β is project-dependent. A higher T_e reflects more complex reasoning paths and multi-constraint composition.
4. **Graph diameter (T_d)** — The longest shortest path between any two nodes in the graph.
   - Criterion: $T_d \geq \gamma$, where γ is a tunable threshold. A larger diameter indicates deeper multi-hop reasoning chains.

The thresholds α, β, and γ can be flexibly set according to the difficulty level required by specific projects. Absence of orphan nodes ensures basic solvability, while sufficiently large T_e and T_d indicate reasoning complexity, and a higher T_a enforces uniqueness.

An example of a well-structured graph extracted from a constructed question is shown in Figure3. This example illustrates how a linguistically complex prompt can be decomposed into a structured graph of entities, attributes, and relations, forming the foundation for subsequent quality verification.

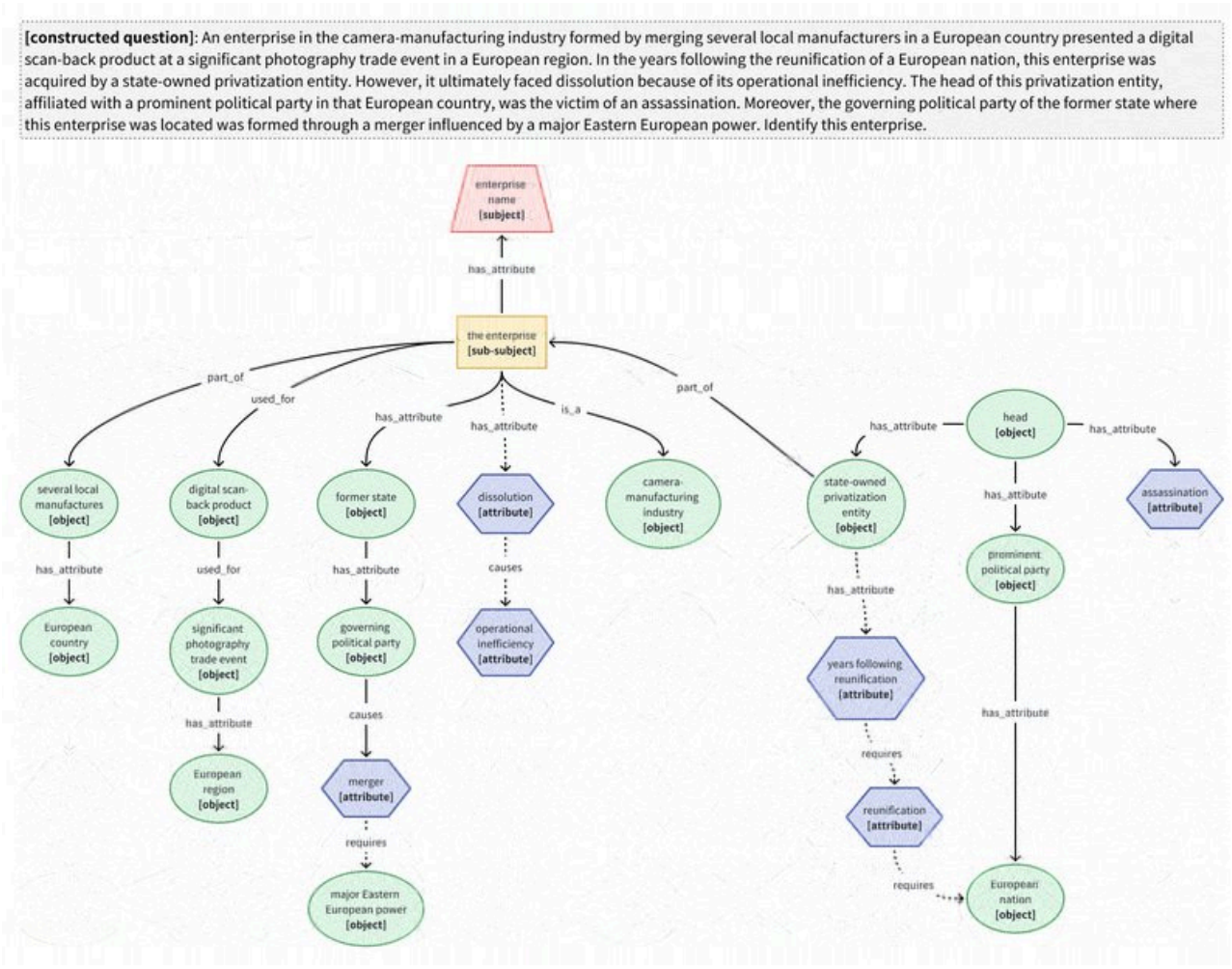


Figure 3: Example of graph-based textual structure extracted from a constructed multi-hop question. Nodes represent subjects, objects, and attributes, while edges encode linguistic relations (is_a, part_of, has_attribute, causes, requires, used_for).

4.2 Data Quality Evaluation Workflow

To ensure the reliability and discriminative power of the constructed dataset, we design a two-step Data Quality Evaluation Workflow, as shown in Figure 4. This system integrates multi-model answering, structured clue extraction, and evidence-based validation. The first step quickly eliminates invalid questions through majority voting, while the second step provides a rigorous, auditable evaluation based on explicit clue decomposition and structured evidence matching. As a result, only those questions that are both uniquely solvable and well-supported by evidence are preserved in the final dataset.

4.2.1 Prediction-Based Validation

In the first stage, the constructed question is answered by multiple inference runs, yielding a set of candidate predictions $\{\text{seed}, \text{prediction1}, \text{prediction2}, \dots\}$. If the majority of predictions match the seed answer, the question is considered stable and accepted directly. Otherwise, it is flagged for further analysis, together with the seed and all alternative predictions. This serves as a coarse filtering mechanism, effectively removing questions with unclear or inconsistent reasoning.

4.2.2 Constraint Decomposition and Evidence-Based Verification

For questions that fail the initial validation, we apply a structured verification procedure composed of three tightly coupled steps:

1. **Clue Decomposition and Normalization**

   The question is decomposed into a set of atomic, verifiable constraints expressed as structured predicates.

   - Vague time expressions are normalized into explicit intervals (e.g., "early 2020s" $\rightarrow [2020, 2023]$).

- Fuzzy geographic or categorical terms are mapped into structured hints (e.g., "southern Indian state" → {region_hint: South}).
- Each predicate is annotated with its source span and a confidence score, while uncertain values are represented as null with an accompanying explanation.

This yields a structured representation that ensures full coverage of the question text.

2. **Explicit Condition Screening**

   The structured predicates are used to evaluate each candidate answer against explicit conditions such as time, location, entity type, and key attributes.

   - Candidates that fail to satisfy any explicit constraints are discarded immediately.
   - If only the seed remains, the question is accepted at this stage.
   - If multiple candidates remain, they are ranked based on their explicit-match score S_{exp}. If the seed achieves the uniquely highest score, the question is accepted; otherwise, it moves to fine-grained verification.

3. **Fine-Grained Evidence Matching**

   For remaining candidates, we perform a detailed comparison against an evidence pack built from local knowledge bases and, if necessary, external retrieval.

   - Each predicate is evaluated under a four-value scheme: Y (Match), P (Partial), U (Unknown), N (Contradiction).
   - High-priority contradictions (N) result in immediate elimination.
   - A normalized score S_{norm} is computed by aggregating weighted evaluations across all constraints.
   - Explicit evidence references and concise justifications are required for each decision.
   - If the seed achieves the uniquely highest S_{norm}, the question is retained; otherwise, it is discarded.

5 Conclusion

In this work, we present a comprehensive framework for the automatic construction of multi-hop question answering datasets that bridge the gap between existing benchmarks and real-world reasoning demands. Motivated by the limitations of current datasets—such as the shallow reasoning depth of early benchmarks and the evaluation-only nature of BrowseComp—we aim to generate large-scale, training-ready data that retain the hard-to-search yet easy-to-verify characteristics crucial for deep reasoning evaluation.

Our approach introduces a four-stage pipeline encompassing data adaptation, node construction, evidence chain generation, and question synthesis, transforming raw Wikipedia dumps into logically rich, semantically diverse, and structurally coherent reasoning graphs. Furthermore, we propose a Data Quality Evaluation System that integrates multi-model consensus, structured clue decomposition, and evidence-based verification. This system ensures that only questions with unique answers, consistent logic, and verifiable evidence are preserved, significantly improving the reliability and discriminative power of the generated dataset.

By automating the construction of BrowseComp-level multi-hop datasets, our framework reduces the prohibitive cost of manual curation while enabling scalable production of challenging, high-quality training data. This advancement has the potential to accelerate research on reasoning-centric large language models and facilitate the development of models capable of complex, multi-step inference.

In future work, we plan to extend this pipeline beyond text-based sources to incorporate multimodal evidence, explore cross-lingual dataset construction, and integrate our framework with reinforcement learning workflows to further enhance model reasoning capabilities. We believe this direction will help bridge the gap between benchmark-oriented evaluation and real-world reasoning applications.

References

[1] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

[2] Alon Talmor and Jonathan Berant. Repartitioning of the complexwebquestions dataset. *arXiv preprint arXiv:1807.09623*, 2018.

[3] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

[4] Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090, 2020.

[5] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*, 2020.

[6] Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

[7] Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, et al. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese. *arXiv preprint arXiv:2504.19314*, 2025.

[8] Zhiyu Shen, Jiyuan Liu, Yunhe Pang, and Yanghui Rao. Hopweaver: Synthesizing authentic multi-hop questions across text corpora. *arXiv preprint arXiv:2505.15087*, 2025.

[9] Potsawee Manakul, Adian Liusie, and Mark JF Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*, 2023.

[10] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

[11] Mingcai Yuan, Qiang Lu, Xianhao Zeng, Jake Luo, and Dawei Li. Alleviating semantic drift in multi-hop question answering on knowledge graphs with bidirectional semantics. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2024.

[12] Shiyue Zhang and Mohit Bansal. Addressing semantic drift in question generation for semi-supervised question answering. *arXiv preprint arXiv:1909.06356*, 2019.

[13] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70, 2020.

[14] Vikas Yadav and Steven Bethard. A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*, 2019.

[15] Jue Wang and Wei Lu. Two are better than one: Joint entity and relation extraction with table-sequence encoders. *arXiv preprint arXiv:2010.03851*, 2020.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Bidirectional encoder representations from transformers. *arXiv preprint arXiv:1810.04805*, 15, 2018.

[18] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multilingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024.

[19] Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. Label verbalization and entailment for effective zero-and few-shot relation extraction. *arXiv preprint arXiv:2109.03659*, 2021.

[20] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[21] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. Introduction to algorithms (3-rd edition). *MIT Press and McGraw-Hill*, 2009.

Appendix

A. Example of Raw Wikipedia Page Noise

To illustrate the challenges inherent in processing raw Wikipedia data, we provide an annotated excerpt from the Japan Airlines page (Figure A1). The figure highlights typical sources of semantic noise that must be addressed during preprocessing and candidate filtering:

- **Excessive citations and references**: Sections such as *See also*, *References*, and *External links* introduce numerous links that do not directly contribute to entity relationships.
- **Irrelevant common nouns**: Terms like *tons* or *mail* are common words that, if expanded as graph nodes, would produce meaningless connections.
- **Abstract concepts and verbs**: Words such as *privatised* or *capital* are overly generalized or contextually ambiguous.
- **Weakly related references**: Links embedded in bibliographic or supplementary sections are often tangential and unlikely to contribute to coherent reasoning chains.

By systematically removing such elements through preprocessing and Named Entity Recognition (NER)-based filtering, we ensure that only concrete, contextually grounded entities (e.g., All Nippon Airways, Tokyo) remain as candidate nodes for multi-hop reasoning.

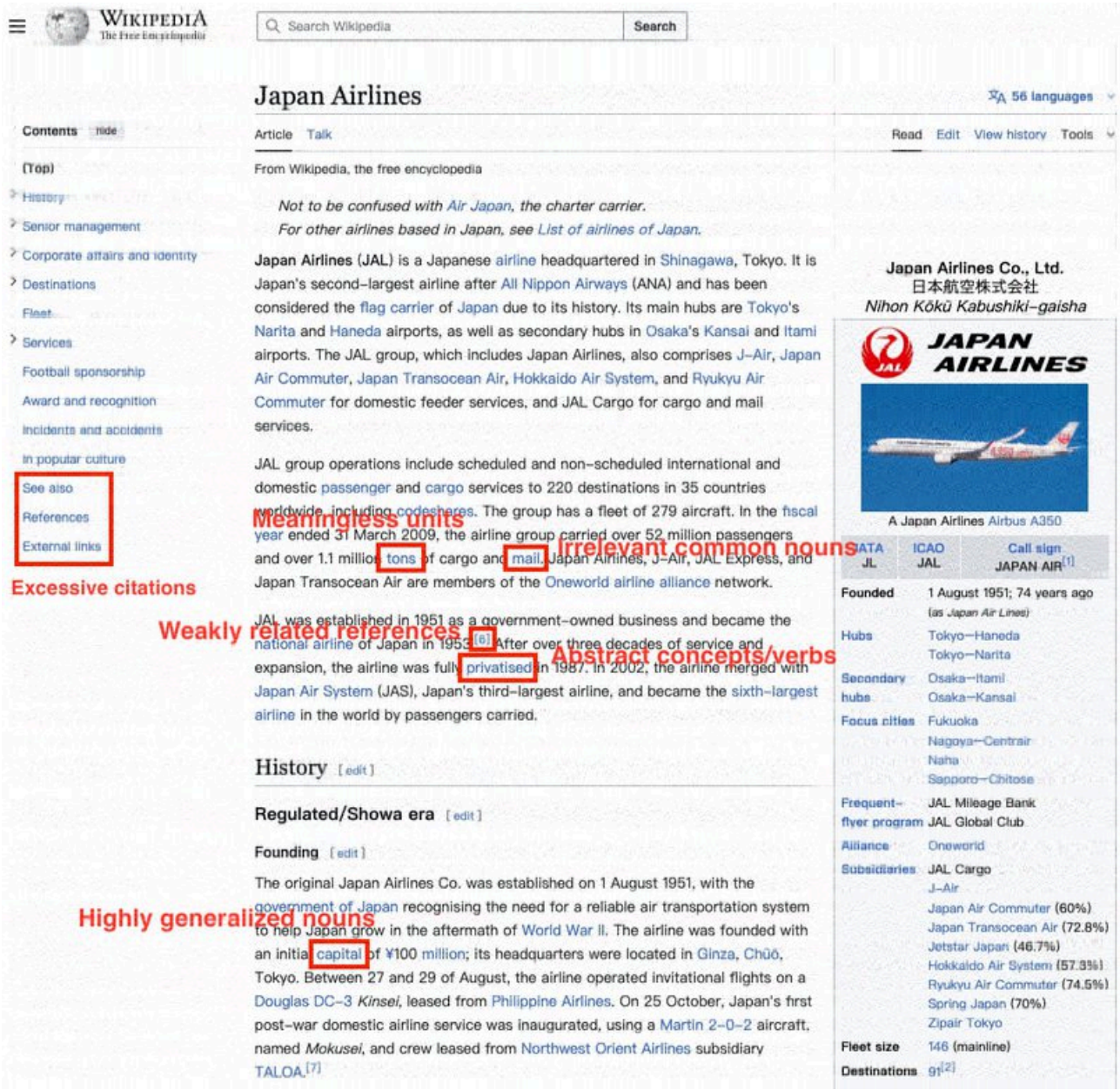


Figure A1: Annotated excerpt from the Wikipedia page "Japan Airlines" highlighting excessive citations, irrelevant common nouns, abstract concepts, and weak references. These noisy elements motivate the preprocessing and filtering steps described in Section 3.2.