SMART: Structural Entropy-Based Multi-perspective Abstraction for Retrieval in RAG Systems

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) systems address the factual inaccuracies limitation of Large Language Models (LLMs). However, the absence of principled methodologies for extracting facts has led current RAG systems to uninformative fact-abstractions. To address these challenges, we propose two principles for fact abstraction in RAG systems: the *Information-Maximization Principle* and the *Multi-Perspective Principle*. These principles reformulate the task of extracting facts into an optimization problem based on information-theoretic quantities, providing a reliable framework for fact abstraction. Building on these principles, we introduce the *Structural Entropy-Based Multi-Perspective Abstraction for Retrieval Technique* (SMART). Extensive experiments on three real-world datasets demonstrate that SMART significantly improves RAG systems, achieving notable gains in multi-hop/perspective question-answering tasks [1].

1 Introduction

Although effective in various tasks, Large Language Models (LLMs) often exhibit factual inaccuracies due to their lack of task-specific real-world knowledge (Hu et al., 2023; Chen et al., 2024). Information Retrieval (IR) techniques have emerged as essential tools for accessing relevant information in LLM-based systems, forming the foundation of Retrieval-Augmented Generation (RAG) systems (Lewis et al., 2020; Gao et al., 2023; Salemi et al., 2024). Research on RAG systems reveals that the requirements for IR models in such systems differ from those of traditional IR, which primarily focuses on retrieving precise most relevant information for a given query. In the context of RAG systems, while LLMs can process complex knowledge, the key challenge lies in retrieving sufficient relevant information to provide adequate context for generation. LLMs require IR methods to supply not only question-relevant information but also latent and comprehensive knowledge beyond what is explicitly stated in documents.

Studies such as (Hu et al., 2024; Mavromatis and Karypis, 2024; Wang et al., 2024a) suggest that ideal generation often requires multi-hop relationships between entities, which are implied but not explicitly stated in the documents. Additionally, (Guo et al., 2024; Edge et al., 2024) demonstrate that incorporating latent hierarchical abstractions can enhance the generation process. Furthermore, (Salemi et al., 2024; Feng et al., 2024) emphasize that considering diverse perspectives on the same fact enriches the analysis. Based on these studies, we classify applicable facts in a database into two types: 1) *Explicit facts*, which are directly extracted from document chunks (e.g., "Cows produce milk", "Cats produce milk", "Dogs produce milk", "Bats produce milk", "Chicken do not produce milk" and "Ducks cannot produce milk";and 2) *Implicit facts*, which cannot be directly extracted but are inferred by combining explicit facts. For example, from the above explicit facts, one can infer: "A group of animals, including cows, cats, dogs, and bats, share biological similarities as milk producers." and "There are other animals that do not produce milk". Implicit facts can come from multiple perspectives. For instance, animals could also be categorized based on their habitat traits, providing alternative viewpoints.

Humans excel at extracting multi-perspective implicit facts by combining or comparing known facts. However, existing IR methods in RAG systems struggle to reliably handle implicit facts. Although (Guo et al., 2024; Edge et al., 2024) highlight the importance of implicit facts, most approaches capture them using heuristic designs, making it unclear whether the retrieved facts align with the task's demands. Unlike traditional IR tasks, IR for RAG systems lacks guiding principles, leading to over-

[1] *https://anonymous.4open.science/r/SMART_CODES/.*

looked critical information and incomplete retrieval of implicit facts. This raises a pivotal question: *How can we design an IR system that provides LLMs with implicit facts in addition to explicit facts to effectively address information gaps in RAG systems?*

In this paper, we address this question in two steps. First, by analyzing the intrinsic knowledge structure of implicit facts, we propose two principles for abstracting such facts from documents. Second, based on these principles, we introduce the Structural entropy-based Multi-perspective Analysis for Retrieval Technique (SMART), a plug-in framework that enhances RAG systems by providing comprehensive context, including both explicit and implicit facts from multiple perspectives.

Specifically, we analyze the knowledge structure of implicit facts using Structural Information Theory. Inspired by prior research on graph analysis, we observe that the usefulness of an extracted fact can be evaluated by analyzing the information content of its corresponding encoding tree. To this end, we define two principles for fact abstraction:

P1. **Information-maximization principle** defines optimiation problem of extracting informative implicit facts.

P2. **Multi-perspective principle** transform the goal of exploring implicit facts diversity into mathematical constraints

By adhering to these principles, the fact abstraction task becomes an optimization problem, simplifying its application in RAG systems.

In the SMART framework, a GNN-based approach extracts informative multi-perspective implicit facts under the guidance of a loss function derived from the fact abstraction principles. This GNN-based abstractor captures the most relevant implicit facts and augments the database with these facts, bridging information gaps left by explicit facts. Additionally, SMART replaces the original retriever in RAG systems with a comprehensive retrieval model that retrieves both explicit and implicit facts. This design enables SMART to provide RAG systems with sufficient context for generation.

Through extensive experiments on three real-world datasets, we demonstrate that SMART significantly improves RAG system performance, achieving up to 11%, 2%, and 11% increases in answer accuracy on the MultiHop, TriviaQA, and HotpotQA datasets, respectively. We also generate multi-hop and 2–4 perspective QA sets based on these datasets. SMART achieves 11%, 3%, and 11% improvements in multi-perspective scenarios, establishing itself as a powerful IR framework that equips LLMs with both explicit and implicit knowledge across diverse perspectives. The main contributions of this paper are summarized as follows:

- Introduction of fact abstraction principles, based on Structural Information Theory, to guide implicit fact extraction in RAG systems.
- Design of the Structural entropy-based Multi-perspective Analysis for Retrieval Technique to enhance RAG systems' information retrieval capabilities.
- Experimental validation demonstrating that SMART improves RAG system performance.

2 Related Works

Recent studies have rapidly expanded the IR toolbox to develop methods capable of addressing the comprehensive knowledge demands of RAG systems. A significant portion of research focuses on improving the quality of relevance-based retrieval. These studies highlight the contributions of effective IR models and propose various approaches to constructing robust IR tools for RAG systems, such as fine-tuning LLMs to cooperate with IR models (Zhang et al., 2024) and modifying IR models based on LLM knowledge distillation (Salemi et al., 2024). While these methods demonstrate impressive progress, they overlook a critical limitation of relevance-only retrieval workflows: information with latent relevance to the query is often ignored by smaller IR models. Research has shown that such omissions can substantially hinder the performance of LLMs, which excel at handling complex, real-world knowledge. For example, (Mavromatis and Karypis, 2024) demonstrates that providing all documents of the entities in the shortest paths connecting queried entities and answer candidates is more effective than retrieving only documents directly relevant to the answer candidates. Similarly, (Hu et al., 2024) underscores the importance of latent relational graph structures in documents for RAG systems. These findings suggest that extracting relations between entities in documents is essential for effective IR in RAG systems. However, many existing methods fail to consider hierar-

chical abstractions that cannot be directly extracted from the provided documents. This oversight leads to an overemphasis on local information while neglecting the retrieval of global information across all provided documents. To address this, (Edge et al., 2024) introduces an IR framework that generates hierarchical abstractions of documents using pre-defined hierarchical clustering methods. Additionally, (Guo et al., 2024) proposes a mechanism to aggregate information across multiple related entities and relationships, enabling the retrieval of high-level document abstractions. These approaches successfully improve generation results by leveraging hierarchical abstractions. However, their predefined mechanisms are not optimized for effective information retrieval and may fail to close the information gaps in the original documents. Furthermore, studies on LLMs reveal that incorporating multiple perspectives on the same fact can enhance analytical comprehensiveness and improve generation results (Salemi et al., 2024; Feng et al., 2024). Despite this, existing RAG research lacks sufficient exploration into retrieving diverse perspective-based information for LLMs.

Our work is also inspired by research on structural entropy. Measuring information in structural data has been regarded as a significant challenge in 21st-century computer science. Structural entropy has been extensively studied as a tool to analyze the information embodied in structural representations (Brooks Jr, 2003). Numerous entropy measures have been developed to study the topology of networks (Dehmer, 2008; Anand and Bianconi, 2009). For instance, Infomap (Rosvall et al., 2009) analyzes graphs using random walks, while (Li and Pan, 2016; Li et al., 2016) employ structural entropy to evaluate network properties. Furthermore, (Liu et al., 2019) introduces residual entropy derived from noise generation for community detection. Despite these advancements, structural entropy techniques have yet to be applied in studies of LLM-based systems.

3 Problem Formulation

Basic Components of RAG Systems. Let $\mathcal{W}$ represent the set of tokens. A query can be defined as a positive closure of the token set: $Q \in \mathcal{W}^+$. The collected facts are stored in a database. Denoting the power set of $\mathcal{W}^+$ as $\mathcal{P}(\mathcal{W}^+)$, the database is represented as $\mathcal{D} \in \mathcal{P}(\mathcal{W}^+)$, where each $D \in \mathcal{D}$ corresponds to a document chunk containing collected facts. Upon receiving a query, a retriever $f_{IR} : \mathcal{W}^+ \times \mathcal{P}(\mathcal{W}^+) \to \mathcal{W}^+$ identifies relevant facts to support the LLM[2]. Let $f_{\text{Emb}} : \mathcal{W}^+ \to \mathbb{R}^\ell$ denote an embedding function that maps a token sequence to an ℓ-dimensional representation vector. Denote the inner product of two vectors by $\langle \cdot, \cdot \rangle$. A vector-based retriever is defined as[3]:

$$f_{IR}(Q, \mathcal{D}) := \arg\max_{D \in \mathcal{D}} \langle f_{\text{Emb}}(Q), f_{\text{Emb}}(D) \rangle.$$

Let $\mathcal{M} : \mathcal{W}^+ \to \mathcal{W}^+$ denote a pre-trained LLM, and let $\oplus$ denote the concatenation of token sequences. The answer $\mathcal{A} \in \mathcal{W}^+$ generated by a RAG system is expressed as:

$$\mathcal{A} = \mathcal{M}(Q \oplus f_{IR}(Q, \mathcal{D})).$$

Representing Facts in the Database with Graphs. In a RAG system, certain facts are directly conveyed through documents stored in the database $\mathcal{D}$. These are referred to as *explicit facts*. As the volume of collected explicit facts grows, their knowledge structure becomes increasingly complex. Many RAG systems leverage graphs to model these explicit facts, as graphs effectively capture complex relationships between entities and their interactions. A graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ is constructed where nodes $\mathcal{V}$ represent entities and edges $\mathcal{E}$ represent relationships extracted from $\mathcal{D}$. For example, let $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$ represent the entities "cows" and "milk", respectively, and let e denote the relation "produce". The fact "Cows produce milk" is represented as the edge $(v_i, e, v_j) \in \mathcal{E}$. Conversely, let $v_{i'} \in \mathcal{V}$ represent the entity "ducks". The absence of an edge between $v_{i'}$ and v_j implies the fact "Ducks do not produce milk". For simplicity, we represent edges as (v_i, v_j) instead of (v_i, e, v_j) in subsequent discussions, as this does not affect the overall analysis. In practice, the graph $\mathcal{G}$ is constructed using an entity-relation extraction algorithm powered by an LLM. The detailed process is described in Algorithm 2 in Appendix A.1,

Additionally, we introduce two functions that model the links between documents and nodes in the graph $\mathcal{G}$ during the graph construction process outlined in Algorithm 2. Define $f_{\text{sour}} : \mathcal{V} \to \mathcal{W}^+$

[2]Various types of retrievers exist; in this paper, we use the widely adopted vector-based retriever as an example. Note that our proposed method is not restricted to this type of retriever; it is also applicable to other retrieval approaches.

[3]In real-world scenarios, the retriever typically retrieves the top-k results rather than a single maximum. However, for simplicity, we use the maximum retrieval here, which does not affect the overall design of the RAG system.

as a function that maps nodes in the graph $\mathcal{G}$ to token sequences composed of the document chunks in $\mathcal{D}$ that reference them. Similarly, define $f_{\text{extr}} : \mathcal{D} \to \mathcal{P}(\mathcal{V})$ as a function that maps document chunks in the database to a set of nodes representing the entities they reference. These two functions are essential for maintaining relationship between nodes and documents within RAG system.

Recent studies have revealed that RAG systems often fail to emulate the human ability to comprehend implied facts from the provided documents (Edge et al., 2024; Guo et al., 2024). Most previous RAG systems, which primarily focus on explicit facts, fail to capture latent information. This limitation raises a critical research question: *How can abstracted latent information be effectively provided to LLMs within a RAG system?*

4 Methodology

4.1 Principle for Fact Abstraction

Humans have the ability to extract facts from high-level combinations or comparisons of superficial explicit facts. We refer to these facts as *implicit facts*. These implicit facts help human construct high-level understanding of the world. Implicit facts cannot be directly extracted from one specific document chunk in the data base. They are inferred through combinations of multiple explicit facts.

Information-max principle. For a RAG system, the captured implicit facts are expected to help LLM under the world better. Form the view of Information, the information quantity of captured implicit facts is expected to be as much as possible. Thus, the first principle is drawn from quantitative analysis of information inside captured implicit facts. This principle translate the task of finding helpful implicit facts into an optimization problem.

As discussed above, the facts make up a hierarchical structure with implicit facts on high-level part while explicit facts in the lowest-level and explicit facts can be modeled by graphs. To model such hierarchical structure, Encoding Tree of graphs introduced by (Li and Pan, 2016) is proved effective and widely applied (Cao et al., 2024).

In this RAG scenario, we build Encoding Tree of graph $\mathcal{G}$. Construct a tree denoted by T, with V denoting its node set. Define $f_{\text{child}} : V \to \mathcal{P}(V)$ as a function returning the child nodes of input node, where $\mathcal{P}(V)$ denotes the power set of V. Then, leaf node set of T is defined as:

$$V_{\text{leaf}} := \{v \in V \mid f_{\text{child}}(v) = \emptyset\}.$$

T is an Encoding Tree of $\mathcal{G}$ if all leaf nodes in T correspond to nodes in $\mathcal{G}$, i.e.:

$$V_{\text{leaf}} = \mathcal{V}.$$

Define $f_{\text{parent}} : V \to V$ as a function returning the parent node of input node. The internal node (non-leaf, non-root) set of T is defined as:

$$V_{\text{inte}} := \{v \in V \mid f_{\text{child}}(v) \neq \emptyset \wedge f_{\text{parent}}(v) \in V\}.$$

Each inter node $v \in V_{\text{inte}}$ represents combination of its children nodes, and thus indicates abstract implicit fact derived from its child nodes. For example, let $v_\alpha \in V_{\text{inte}}$ be a node follows $f_{\text{child}}(v_\alpha) = \{v \in \mathcal{V} \mid (v, v_j) \in \mathcal{E}\}$, where $v_j \in \mathcal{V}$ be the node indicate entity "milk". Then, from the sub-tree with v_α as root, denoted by T_{v_α}, a implicit fact can be implied that "A group of animals, including cows, cats, dogs, and bats, share biological similarities as milk producers.".

Since the sub-tree structure of each $v \in V_{\text{inte}}$ reveals the implicit fact, from information view, capturing sufficient implicit fact is equivalent to making the sub-tree structure of all internal nodes informative. In other words, a RAG system demand information content captured by the sub-tree structure of all internal nodes being maximized. Let $\theta_v = \log \frac{|f_{\text{child}}(v)|}{|f_{\text{child}}(f_{\text{parent}}(v))|}$. Structural information theory (Li and Pan, 2016) introduced a tool to evaluate the information content of the sub-tree of an internal node v. Let $\mathcal{N}_v := f_{\text{child}}(v)$ denote the set of v's children:

$$\mathcal{H}^T(\mathcal{G}; v) := \theta_v \times \left\{ \sum_{\mathcal{E}} p(v_i \in \mathcal{N}_v \wedge v_j \notin \mathcal{N}_v) \right\}. \quad (1)$$

Based on (1), they provide the evaluation of the information content of the sub-tree of all internal nodes in T, called the structural entropy of T, denoted as $\mathcal{H}^T(\mathcal{G})$:

$$\mathcal{H}^T(\mathcal{G}) = \sum_{v \in V_{\text{inte}}} \mathcal{H}^T(\mathcal{G}; v). \quad (2)$$

Structural entropy $\mathcal{H}^T(\mathcal{G})$ increases with the information content of implicit facts captured by T. Thus, we have the first principle for abstraction of implicit facts, the *Information-max Principle*:

P1. **During fact abstraction, extract implicit facts whose corresponding encoding tree has the maximized structural information.**

$$\arg\max_{T} \mathcal{H}^{T}(\mathcal{G}). \tag{3}$$

Specially, we notice that for a balanced T, where the number of children at each level is uniform, θ becomes constant, and $\mathcal{H}^{T}(\mathcal{G})$ approximates the graph modularity. This special case aligns with the design in GraphRAG(Edge et al., 2024). In GraphRAG, the implicit fact abstraction is done by applying the Leiden algorithm (Traag et al., 2019) which maximize the graph modularity of $\mathcal{G}$. Aligning with Information-max Principle provide GraphRAG helpful implicit facts and Information-max Principle print out the direction for more common cases.

Multi-perspective principle. In real-world scenarios, explicit facts can be analyzed from various perspectives, leading to different implicit facts. Analyzing explicit facts from a single perspective risks overlooking some implicit relationships. For example, given the explicit fact "Bats produce milk and possess wings", one can infer the implicit fact "A group of animals (cows, cats, dogs, bats) that produce milk are biologically similar" from the perspective of milk production. Conversely, from the perspective of possessing wings, the implicit fact "A group of animals (chickens, ducks, bats) with wings are biologically similar" can be derived. These two perspectives cannot be effectively represented using a single encoding tree.

To address this limitation, we employ multiple encoding trees to capture implicit facts from different perspectives:

$$\mathcal{T} = \{T \mid V_{\text{leaf}} = \mathcal{V}\}. \tag{4}$$

Two distinct trees within the ideal encoding tree set $\mathcal{T}$ should reflect implicit facts abstracted from different perspectives. In other words, the internal nodes of two different trees in $\mathcal{T}$ should contain no overlapping information. To model this scenario, we use mutual information (MI) (Cover, 1999), a widely applied tool. Here, we compute the MI between the internal nodes of two distinct encoding trees.

Let $T^k \in \mathcal{T}$ and $T^{k'} \in \mathcal{T}$ denote two encoding trees, and let V^k_{inte} and $V^{k'}_{\text{inte}}$ represent their internal nodes, respectively. Let $v \in \mathcal{V}$ denote a node randomly selected from the graph $\mathcal{G}$. The MI between the internal nodes is defined as:

$$\begin{aligned}
\mathcal{I}(V^k_{\text{inte}}; V^{k'}_{\text{inte}}) &= \sum_{v_\alpha \in V^k_{\text{inte}}} \sum_{v_\beta \in V^{k'}_{\text{inte}}} P_{\alpha\beta} \times \mathcal{H}_{\alpha\beta}, \\
\text{where } P_{\alpha\beta} &= p(v \in f_{\text{child}}(v_\alpha) \cap f_{\text{child}}(v_\beta)), \\
\mathcal{H}_{\alpha\beta} &= \log \frac{p(v \in f_{\text{child}}(v_\alpha) \cap f_{\text{child}}(v_\beta))}{p(v \in f_{\text{child}}(v_\alpha))p(v \in f_{\text{child}}(v_\beta))}.
\end{aligned}$$

For T^k and $T^{k'}$ to capture different perspectives, the condition $\mathcal{I}(V^k_{\text{inte}}; V^{k'}_{\text{inte}}) = 0$ holds. The ideal set $\mathcal{T}$ is expected to include all encoding trees reflecting different perspectives. However, due to the input length limitations of LLMs in RAG systems and the computational cost of constructing encoding trees, it is often impractical to include all such trees. As a practical solution, we select K encoding trees that maximize the information content of implicit facts while ensuring no overlap among them. Thus, let $\kappa \in \mathbb{N}^+$, the demanded set of κ encoding trees can be formulated as:

$$\{T^k \mid \forall k \neq k', \mathcal{I}(V^k_{\text{inte}}; V^{k'}_{\text{inte}}) = 0.\} \tag{5}$$

However, the constraint in (5) is too restricted that makes it hard to search for the tree set satisfies it. Thus, we transform it into a principle to make multiple encoding trees have the least overlaps, i.e., the *Multi-perspective Principle*:

P2. **During fact abstraction, extract implicit facts from views whose corresponding encoding tree has the minimized mutual information.**

$$\arg\min_{\mathcal{T}} \sum_{T^k, T^{k'} \in \mathcal{T}} \mathcal{I}(V^k_{\text{inte}}; V^{k'}_{\text{inte}}). \tag{6}$$

Combining the Information-max principle and the Multi-perspective principle, an appropriate goal for implicit fact abstraction of a RAG system can be formulated by optimizing:

$$\begin{aligned}
\hat{\mathcal{T}} = \arg\min_{\mathcal{T}} \Big\{ &- \sum_{T^k \in \mathcal{T}} \mathcal{H}^{\mathcal{T}}(\mathcal{G}) \\
&+ \sum_{T^k, T^{k'} \in \mathcal{T}}^{k \neq k'} \mathcal{I}(V^k_{\text{inte}}; V^{k'}_{\text{inte}}) \Big\}.
\end{aligned} \tag{7}$$

4.2 SMART framework

Following the fact abstraction principles, we propose Structural entropy-based Multi-perspective Abstraction framework for Retrieval Technique (SMART), a plug-in framework to improve the information retrieval method of RAG systems. SMART include two crucial components, the GNN-based Fact-Abstraction model and the Comprehensive Retrieval model.

GNN-based Fact-Abstraction. We apply a Graph Neural Network (GNN) based model to learn the $\hat{\mathcal{T}}$. Specifically, let $d \in \mathbb{N}^+$ denote the dimension of GNN learned embeddings. Let $f_{\text{GNN}} : \mathcal{V} \to \mathbb{R}^d$ denote a GNN model. Then we learns embedding $H \in \mathbb{R}^{|\mathcal{V}| \times d}$ that represents each node of $\mathcal{G}$:

$$H = f_{\text{GNN}}(\mathcal{G}).$$

Then, let $d_m \in \mathbb{N}$ be hyper parameters denote the node number of learned encoding trees at level m. Note that for any T^k, d_0 is the dimension of GNN output,i.e., $d_0 = d$. We apply learnable matrix set $\mathcal{C} = \{C_{km} \mid C_{km} \in \mathbb{R}^{d_{(m-1)} \times d_m}\}$, and function $\sigma : \mathbb{R} \to [0, 1]$ that maps real value into probabilities. Thus we obtain indicator matrix Y_{km} for each encoding tree T^k at level m:

$$Y_{km} = \sigma(H \cdot \prod_{k=1}^{\kappa} C_{km}). \tag{8}$$

Let $Y_{km}[i, \alpha]$ be the i-th row α-th column value of Y_{km}, we use it to indicate whether the node v_i is in the subtree with v_α as root:

$$\begin{aligned} v_i \in \Big\{ \underbrace{f_{\text{child}}(\cdots f_{\text{child}}(}_{\text{repeat } m \text{ times}} v_\alpha) \cdots) \Big\} \quad \text{if} \quad Y_{km}[i, \alpha] = 1, \\ v_i \notin \Big\{ \underbrace{f_{\text{child}}(\cdots f_{\text{child}}(}_{\text{repeat } m \text{ times}} v_\alpha) \cdots) \Big\} \quad \text{if} \quad Y_{km}[i, \alpha] = 0. \end{aligned} \tag{9}$$

Then, with (9), we are able to construct every $T^k \in \mathcal{T}$ Let $\mu \in \mathbb{R}$ and $\lambda \in \mathbb{R}$ denote a hyper-parameters, $\vec{1}^m = 1^{d_m \times 1}$ denote a vector with all elements be 1, and $A \in [0, 1]^{|\mathcal{V}| \times |\mathcal{V}|}$ denote the normalized adjacency matrix of graph $\mathcal{G}$. Applying the indicator matrix in (8), the search for $\hat{\mathcal{T}}$ following (7) can be reformulated as an optimization problem to minimize $\mathcal{L}$:

$$\begin{aligned} \mathcal{L} = &- \sum_{km} Y_{km}[i, j](A[i, j] - \mu \sum_{i'j'} A[i, j']A[i', j]) \\ &+ \lambda \sum_{k \neq k', m} (Y_{km} \otimes Y_{k'm} \cdot \vec{1}^m) \log \frac{(Y_{km} \otimes Y_{k'm} \cdot \vec{1}^m)}{(Y_{km}\vec{1}^m) \times (Y_{k'm} \cdot \vec{1}^m)} \end{aligned} \tag{10}$$

Thus, we can train GNN model $\{_{\text{GNN}}$ and update learnable matrix set $\mathcal{C}$ with back propagation minimizing loss $\mathcal{L}$. After training process, the learned encoding tree set $\hat{\mathcal{T}}$ can be constructed according to indicator matrix set $\{Y_{km}\}$.

Comprehensive Retrieval. With Encoding trees constructed by (10), the implicit facts are indicated by internal nodes of these encoding trees. Then we transform the abstract tree structure into documents chunks that LLM can understand. We apply the function f_{sour} in Section 3 to generate description, i.e., token sequence, for node v_i in graph $\mathcal{G}$, by mapping it to documents it was mentioned in $\mathcal{D}$. Define $\hat{D}_i^k \in \mathcal{W}^+$ denote the description of $v_i \in V^k$ of encoding tree T^k. $\hat{D}_i^k$ is constructed by :

$$\hat{D}_i^k = \begin{cases} f_{\text{sour}}(v_i) & \text{if } v_i \in V_{\text{leaf}}^k \\ \mathcal{M}(T_{v_i}^k) & \text{if } v_i \in V_{\text{inte}}^k. \end{cases} \tag{11}$$

Then we form a SMART adaption of RAG system with retriever f_{IR} and database $\mathcal{D}$. Let $\sum$ for documents denote concatenate all included document chunks, $\mathcal{V}' = f_{\text{extr}}(f_{\text{IR}}(Q, \mathcal{D})) \subset \mathcal{V}$ denote the entities mentioned in the returned documents in $f_{\text{IR}}(Q, \mathcal{D})$. This SMART adaption of RAG system retrieves information with $\mathcal{F}(Q, \mathcal{D}, f_{\text{IR}})$:

$$\mathcal{F}(Q, \mathcal{D}, f_{\text{IR}}) = f_{\text{IR}}(Q, \mathcal{D}) \oplus \sum_{v_i \in \mathcal{V}'} \sum_{m} \sum_{Y_{km}[i\alpha]=1} \hat{D}_\alpha^k. \tag{12}$$

The over all process of SMART is described in Algorithm 1:

Algorithm 1 SMART framework

Require: Documents $\mathcal{D}$, LLM $\mathcal{M}$, Basic retriever f_{IR}, Query Q.
Ensure: answer $\mathcal{A}$

```
Construct graph 𝒢 with Algorithm 2
Train the GNN-based fact abstracter with (10)
Construct encoding trees 𝒯̂ according to (9)
for each encoding tree T^k ∈ 𝒯̂ do
    for each internal node v_α ∈ V^k_inte do
        Generate description D̂^k_i according to (11)
    end for
end for
Retrieve context with ℱ(Q, 𝒟, f_IR) by (12)
Generate answer 𝒜 = ℳ(Q ⊕ ℱ(Q, 𝒟, f_IR))
return 𝒜
```

Table 1: Dataset Statistics

Dataset	Basic Statistics			Multi-perspective	
	Entity Count	Relation Count	Document Count	Perspective Count	QA Count
MultiHop RAG	39 359	29 523	600	1	500
				2	677
				3	6 779
				4	2 544
HotpotQA	73 643	64 659	1 000	1	428
				2	1 572
				3	2 668
				4	2 262
TriviaQA	169 374	150 583	1 700	1	300
				2	419
				3	2 581
				4	2 415

Table 2: Comparison on Original QA

	Word Coverage			Acc@0.25			Acc@0.5		
Model	Multihop RAG	HotpotQA	TriviaQA	Multihop RAG	HotpotQA	TriviaQA	Multihop RAG	HotpotQA	TriviaQA
RAG	0.75	0.18	0.73	0.75	0.49	0.73	0.74	0.01	0.73
GRAG	0.70	0.36	0.71	0.73	0.84	0.74	0.71	0.08	0.72
LightRAG	0.70	0.42	0.68	0.70	0.90	0.70	0.69	0.10	0.67
SMART	**0.80**	**0.42**	**0.81**	**0.83**	**0.92**	**0.82**	**0.81**	**0.14**	**0.79**
Impro.	↑ 0.07	0.00	↑ 0.11	↑ 0.11	↑ 0.02	↑ 0.11	↑ 0.09	↑ 0.4	↑ 0.08

Table 3: Comparison on Multi-Perspective QA

	Word Coverage			Acc@0.25			Acc@0.5			Perspective Coverage		
Model	Multihop RAG	HotpotQA	TriviaQA	Multihop RAG	HotpotQA	TriviaQA	Multihop RAG	HotpotQA	TriviaQA	Multihop RAG	HotpotQA	TriviaQA
RAG	0.72	0.25	0.71	0.72	0.52	0.69	0.71	0.03	0.66	0.18	0.15	0.20
GRAG	0.75	0.38	0.66	0.74	0.83	0.75	0.74	0.10	0.73	0.52	0.25	0.47
LightRAG	0.76	0.45	0.69	0.76	0.89	0.74	0.70	0.15	0.74	0.58	0.38	0.51
SMART	**0.81**	**0.47**	**0.82**	**0.84**	**0.92**	**0.83**	**0.82**	**0.18**	**0.79**	**0.78**	**0.65**	**0.73**
Impro.	↑ 0.07	↑ 0.04	↑ 0.15	↑ 0.11	↑ 0.03	↑ 0.11	↑ 0.11	↑ 0.20	↑ 0.07	↑ 0.34	↑ 0.71	↑ 0.43

Table 4: Ablation Study on Head Count

Head Count	Ave. Modularity	Ave. NMI	Word Coverage	Acc@0.25	Acc@0.5
2-head	0.680	0.410	0.70	0.73	0.71
3-Head	0.625	0.480	0.75	0.77	0.74
4-Head	0.762	**0.510**	**0.78**	0.80	0.76
5-Head	**0.789**	0.395	0.77	**0.81**	**0.78**
6-Head	0.785	0.360	**0.78**	0.79	0.77

5 Experiments

Datasets. We evaluate our SMART framework using three widely utilized question answering (QA) datasets: **MultiHop RAG**, **HotpotQA**, and **TriviaQA**. These datasets were chosen due to their inherent complexity in multi-hop reasoning and their natural incorporation of multi-perspective answers. Each dataset contains questions that require synthesizing knowledge from multiple documents, emphasizing the importance of implicit multi-perspective reasoning in QA tasks. For generating multi-perspective questions, we use the **ChatGPT-4o mini** (Achiam et al., 2023) to generate questions from the datasets by providing multiple documents and ask it to generate questions that demand knowledge from all provided documents, including different perspectives. For detailed statistics of the datasets, please refer to Table 1

Baselines We compare our SMART framework against several popular RAG approach es: **Naive RAG** (Lewis et al., 2020), **GRAG** (Edge et al., 2024) and **LightRAG**(Guo et al., 2024). While Naive RAG focuses on explicit facts, both GRAG and LightRAG attempt to enhance performance by incorporating implicit facts, but with varying degrees of complexity and efficiency. By comparing these models, we can better highlight the advantages of the SMART framework in handling multi-hop and multi-perspective question-answering tasks. To evaluate the performance of the models, we use the Word Coverage, Accuracy, Perspective Coverage and Average Modularity. For more about baseline, implementation and evaluation, please refer to Appendix A.3, A.4 and A.5

5.1 Comparison on Original QA

Firstly, we evaluate the performance of the baseline models and our SMART model on the original QA datasets, as shown in Table 2. SMART consistently outperforms other models across all evaluation metrics when applied to original QA tasks. It achieves the highest scores in Word Coverage, indicating its superior ability to generate more relevant and comprehensive answers. Similarly, SMART excels in Acc@0.25 and Acc@0.5, further confirming its effectiveness in producing accurate and precise answers across different datasets. From Table 2, it is clear that SMART surpasses RAG, GRAG, and LightRAG in handling original QA tasks, validating its superior performance in this domain.

5.2 Comparison on Multi-Perspective QA

We also compare the performance of the models on the multi-perspective QA datasets(Table 3). In addition to **Word Coverage**, **Acc@0.25**, and **Acc@0.5**, we evaluate **Perspective Coverage** to assess how well the models capture multiple perspectives. From Table 3, it is evident that SMART demonstrates a clear advantage in Perspective Coverage, which evaluates the model's ability to capture multiple viewpoints—an essential aspect for multi-perspective QA tasks. It excels in generating answers that incorporate diverse perspectives across various datasets, showcasing its strength in

Table 5: Case Study.

Question:What are the key factors contributing to OpenAI's rapid growth and adoption in various sectors?			
No-RAG	Naive-RAG	Graph-RAG	SMART
OpenAI's rapid growth is driven by its early development of powerful language models like GPT-3 and GPT-4, strategic partnerships (e.g., with Microsoft), strong developer-friendly APIs, commercialization through ChatGPT, consistent breakthroughs in AI, successful monetization, and attracting top talent and investment.	OpenAI's growth is fueled by the potential of generative AI, with estimates suggesting it could add up to 4.4 trillion in value. Business leaders widely recognize AI's importance, and global AI spending is projected to reach 154 billion, driving further investment and interest in AI technologies.	Key factors contributing to OpenAI's rapid growth and adoption in various sectors include the overarching AI Revolution that emphasizes the integration of artificial intelligence across different industries, the leadership of influential figures like Mark Surman $\cdots$ Furthermore, the emergence of generative AI technologies, represented by concepts like GPT-X,$\cdots$	OpenAI's rapid growth and adoption in various sectors can be attributed to several key factors, including its development of advanced AI models like ChatGPT and GPT-4 Turbo, which have gained widespread recognition and usage across industries. $\cdots$ Additionally, strategic partnerships, such as that with Microsoft, $\cdots$

addressing the complexities of multi-perspective question answering.

5.3 Ablation Study

We conduct an ablation study on **MultiHop RAG**, where we progressively reduce the number of heads in the SMART model to observe the impact on performance (Table 4). The results show that increasing the number of heads improves model performance across all metrics. From Table 4, it can be seen that the 5-Head configuration achieves the best scores in Word Coverage and Acc@0.25, while the 4-Head model performs well in Acc@0.5, balancing complexity and performance. In terms of Average Modularity and Average NMI, the 5-Head configuration also leads, capturing diverse perspectives without significant overhead. Reducing the head count to 3-Head or 2-Head causes performance to drop, especially in Acc@0.5, with 2-Head showing the weakest performance. Overall, the results confirm that more heads improve the model's ability to handle complex QA tasks, with 5-Head providing the best trade-off between performance and computational cost.

5.4 Case Study

Based on the evaluation results, the SMART model provides the most accurate response, demonstrating strong relevance and a structured analysis that effectively aligns with the knowledge base and directly addresses the core aspects of the question. In contrast, the Naive RAG model, while presenting macro-level data on the factors contributing to OpenAI's growth, lacks specific details about OpenAI itself, failing to delve into its technological innovations and strategic partnerships. The Graph RAG model places excessive emphasis on community aspects, which detracts from focusing on OpenAI's key factors, resulting in a more general response with insufficient analysis of its technological advancements and partnerships. As for the No RAG model, while concise, it lacks depth and data support due to the absence of an external knowledge base, making it difficult to capture the complexities behind OpenAI's growth, rendering it relatively basic and unable to compete with the depth of analysis seen in the other models.

6 Conclusion and Future Work

In this paper, we propose two principles for fact abstraction in RAG systems: the *Information-Maximization Principle* and the *Multi-Perspective Principle*. Building on these principles, we introduce the *Structural Entropy-Based Multi-Perspective Abstraction for Retrieval Technique* (SMART). SMART integrates explicit and implicit facts with multi-perspective analysis, offering comprehensive knowledge to enhance the generation capabilities of LLMs while mitigating issues such as factual inaccuracies. Experiments on three real-world datasets demonstrate that SMART significantly outperforms existing RAG frameworks, validating its effectiveness in handling complex information in question-answering tasks.

Limitations

Despite these advancements, the SMART framework has some limitations. First, the use of a structural entropy-based learning model increases computational complexity. Future work could address this by incorporating faster techniques, such as InfoNCE-based methods. Additionally, SMART currently focuses exclusively on textual data. However, the fact abstraction principles, being grounded in information theory, are applicable across modalities. In summary, future work will focus on enhancing computational efficiency and extending SMART to incorporate multimodal data. By addressing these challenges, SMART has the potential to become an even more powerful framework for tackling broader tasks.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Kartik Anand and Ginestra Bianconi. 2009. Entropy measures for networks: Toward an information theory of complex topologies. *Physical Review E*.

Frederick P Brooks Jr. 2003. Three great challenges for half-century-old computer science. *JACM*.

Yuwei Cao, Hao Peng, Angsheng Li, Chenyu You, Zhifeng Hao, and Philip S Yu. 2024. Multi-relational structural entropy. *arXiv preprint arXiv:2405.07096*.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, and 1 others. 2024. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61.

Thomas M Cover. 1999. *Elements of information theory*. John Wiley & Sons.

Matthias Dehmer. 2008. Information processing in complex networks: Graph entropy and information functionals. *Applied Mathematics and Computation*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Shangbin Feng, Taylor Sorensen, Yuhan Liu, Jillian Fisher, Chan Young Park, Yejin Choi, and Yulia Tsvetkov. 2024. Modular pluralism: Pluralistic alignment via multi-llm collaboration. *arXiv preprint arXiv:2406.15951*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation.

Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2024. Grag: Graph retrieval-augmented generation. *arXiv preprint arXiv:2405.16506*.

Yuntong Hu, Zheng Zhang, and Liang Zhao. 2023. Beyond text: A deep dive into large language models' ability on understanding graph data. *arXiv preprint arXiv:2310.04944*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *Preprint*, arXiv:1705.03551.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

A. Li, Qifu Hu, J. Liu, and Yicheng Pan. 2016. Resistance and security index of networks: Structural information perspective of network security. *Scientific Reports*.

Angsheng Li and Yicheng Pan. 2016. Structural information and dynamical complexity of networks. *IEEE Transactions on Information Theory*, 62(6):3290–3339.

Yiwei Liu, Jiamou Liu, Zijian Zhang, Liehuang Zhu, and Angsheng Li. 2019. Rem: From structural entropy to community structure deception. In *NeurIPS*.

Costas Mavromatis and George Karypis. 2024. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv preprint arXiv:2405.20139*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Martin Rosvall, Daniel Axelsson, and Carl T Bergstrom. 2009. The map equation. *The European Physical Journal Special Topics*.

Alireza Salemi, Surya Kallumadi, and Hamed Zamani. 2024. Optimization methods for personalizing large language models through retrieval augmentation. *SIGIR*.

Ruben Sipos, Adith Swaminathan, Pannaga Shivaswamy, and Thorsten Joachims. 2012. Temporal corpus summarization using submodular word coverage. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, page 754–763, New York, NY, USA. Association for Computing Machinery.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *Preprint*, arXiv:2401.15391.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12.

Keheng Wang, Feiyu Duan, Peiguang Li, Sirui Wang, and Xunliang Cai. 2024a. Llms know what they need: Leveraging a missing information guided framework to empower retrieval-augmented generation. *arXiv preprint arXiv:2404.14043*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024b. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

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

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*.

Ethics Statement

All datasets used in this paper are publicly available, carry open licenses, contain no personally identifiable information (PII), and therefore do not require additional IRB approval. Below we list for each dataset its source, license, privacy considerations, and ethical compliance.

- **MultiHop-RAG**
  - *Source:* GitHub repository `https://github.com/yixuantt/MultiHop-RAG`.
  - *License:* Open Data Commons Attribution (ODC-BY).
  - *Privacy:* Underlying documents are from publicly accessible news articles and contain no PII.
  - *Ethics:* Public data only; no IRB approval required.
- **HotpotQA**
  - *Source:* Official HotpotQA site and GitHub (`https://curtis.ml.cmu.edu/datasets/hotpot/`).
  - *License:* Creative Commons Attribution-ShareAlike 4.0 (CC BY-SA 4.0).
  - *Privacy:* All passages sourced from Wikipedia, which is public domain; no PII.
  - *Ethics:* Public data only; no IRB approval required.
- **TriviaQA**
  - *Source:* GitHub repository and TensorFlow Datasets (`tensorflow_datasets.load("trivia_qa")`).
  - *License:* Apache License 2.0.
  - *Privacy:* Questions and evidence drawn from Wikipedia and openly available web sources; no PII.
  - *Ethics:* Public data only; no IRB approval required.

For the existing datasets (MultiHop-RAG, HotpotQA, TriviaQA), we confirm that their licenses (ODC-BY, CC-BY-SA 4.0, Apache 2.0) permit our offline, research-only use and result redistribution.

For the artifacts we produce (the SMART codebase and the multi-perspective QA samples), we specify in our repository's README that they are released under a CC-BY-4.0 license for research and non-commercial purposes only, in accordance with the original datasets' access conditions.

We used ChatGPT (OpenAI GPT-4) to polish the language of the manuscript. All text suggestions generated by ChatGPT were reviewed, edited, and approved by the authors to ensure technical accuracy and clarity.

A Technical Appendix

A.1 Entity and Relation Extraction Using LLM

Let $\mathcal{V}_D$ and $\mathcal{E}_D$ denote the nodes and edges extracted from a document D:

Algorithm 2 Entity and Relation Extraction Using LLM

Require: Documents $\mathcal{D}$, LLM $\mathcal{M}$, Entity Extraction Prompt (including Entity Types), Relation Extraction Prompt.

Ensure: Extracted entities $\mathcal{V}$ and relationships $\mathcal{E}$

Initialize $\mathcal{V} \leftarrow \emptyset, \mathcal{E} \leftarrow \emptyset$
for each document $D \in \mathcal{D}$ **do**
  Extract entities $\mathcal{V}_D$ using $\mathcal{M}$:
  $\mathcal{V}_D \leftarrow \mathcal{M}(D \oplus \text{Entity Extraction Prompt})$
  Extract relationships $\mathcal{E}_D$ using $\mathcal{M}$ and extracted $\mathcal{V}_D$:
  $\mathcal{E}_D \leftarrow \mathcal{M}(D \oplus \mathcal{V}_D)$
  Update the global entity and relationship sets:
  $\mathcal{V} \leftarrow \mathcal{V} \cup \mathcal{V}_D, \mathcal{E} \leftarrow \mathcal{E} \cup \mathcal{E}_D$
end for
return $\mathcal{V}, \mathcal{E}$

A.2 Dataset Details

MultiHop RAG (Tang and Yang, 2024) evaluates multi-hop reasoning by requiring questions to integrate implicit facts from various documents. Correct answers are derived by combining information from multiple sources, making this dataset a valuable benchmark for RAG systems. **HotpotQA** (Yang et al., 2018) is a Wikipedia-based dataset containing 113,000 question-answer pairs. It requires multi-hop reasoning over multiple documents and provides sentence-level supporting facts for each question. This feature necessitates the ability to perform complex analyses and generate explainable answers. **TriviaQA** (Joshi et al., 2017) comprises over 650,000 question-answer-evidence triples, with each question associated with six supporting documents. This dataset challenges models to synthesize information from multiple sources of evidence, pushing the limits of their reasoning capabilities.

A.3 Baseline Details

Naive RAG (Lewis et al., 2020): This baseline uses a traditional RAG model that retrieves documents based on explicit facts and generates answers from these retrieved documents without considering implicit facts. While it effectively handles simpler questions, it may struggle with complex multi-hop reasoning and integrating information from diverse perspectives. **GRAG** (Edge et al., 2024) is a RAG variant that introduces a structured, hierarchical approach to retrieval-augmented generation. Instead of relying on flat document representations, it extracts a knowledge graph from raw text, builds a community hierarchy, and generates summaries for these communities. By considering the relationships between documents and abstracting hidden facts within these structures, GRAG aims to improve the quality of generated answers, especially for multi-hop and multi-perspective questions. **LightRAG**(Guo et al., 2024) incorporates implicit facts into the retrieval process through a more efficient, dual-level retrieval system. By combining low-level and high-level knowledge discovery, LightRAG uses graph structures integrated with vector representations, allowing for efficient retrieval of related entities and their relationships. This method significantly improves retrieval accuracy and response times while reducing computational costs, making it suitable for large-scale, real-time applications. These baselines were chosen to demonstrate the impact of explicitly considering implicit facts, which is a key feature of our SMART framework.

A.4 Implementation

For the implementation of our baseline models, we use the settings and hyperparameters provided in their respective papers. Specifically, for **Naive RAG** (Lewis et al., 2020), we follow the setup described in the original RAG paper. For **GRAG**, the implementation follows the settings in the paper by Edge et al. (Edge et al., 2024). For **LightRAG**, we use the parameters recommended by Guo et al. (Guo et al., 2024).

For our **SMART model**, we use the following settings:

- κ: The number of trees (head count) in our model. The default value is 3, which corresponds to three heads or three trees.
- m: The number of layers in each tree. The default value is 3.
- λ: The weight for the NMI (Normalized Mutual Information) term. The default value is 1.

- μ: The parameter for the Parent-Children balance in a tree, set to 1e-3 as the default value.

We use **ChatGPT-4o mini** (Achiam et al., 2023) for both entity extraction and answer generation tasks. **ChatGPT-4o mini** is the same model used for entity extraction and generating answers, ensuring consistency across tasks. The **multilingual-e5-large** (Wang et al., 2024b) model is used for document embeddings. The prompt settings for entity extraction, report generation, and answering tasks are detailed in Table 6.

For the hardware, we use a high-performance system equipped with a **NVIDIA GeForce RTX 4090** GPU for efficient training and inference. The system is powered by a **16-core CPU** and is equipped with **80 GB of RAM**, ensuring smooth processing and fast computation for large-scale models and datasets.

A.5 Evaluation Metrics

To evaluate the performance of the models, we use the following metrics: **Word Coverage** (Sipos et al., 2012): Measures the percentage of words in the ground truth answer that are also present in the generated answer.

Acc@0.25 and **Acc@0.5** (Raffel et al., 2020): These metrics measure the accuracy of the generated answer compared to the ground truth, with thresholds of 0.25 and 0.5, respectively. These thresholds represent the percentage of words in the generated answer that must match the ground truth answer to be considered correct.

Perspective Coverage: This metric evaluates how well the model captures multiple perspectives in the generated answers. A higher coverage indicates that the model is considering multiple angles and contexts when generating the answer.

Average Modularity (Traag et al., 2019) and **Average NMI (Normalized Mutual Information) (Cover, 1999)**: These metrics are used to evaluate the effectiveness of the multi-perspective approach in our SMART framework. **Average Modularity** is used to assess the quality of clustering by measuring how well the model groups related facts or perspectives together. A higher modularity score indicates more meaningful and coherent clusters. **Average NMI** is used to evaluate the diversity between the multiple heads generated by the SMART framework. It measures how distinct the perspectives captured by the model are, with higher values indicating greater diversity and separation between the generated heads.

Table 6: Prompt Settings for Different Tasks.

Task	Prompt
Entity and Relationship Extraction	**Goal:** Given a text document and a list of predefined entity types (as a hyperparameter, which can be passed externally), identify all entities of the specified types from the text and the relationships between the identified entities. **Steps:** 1. Identify all entities. For each identified entity, extract the following information: - **entity_name:** Name of the entity (capitalized). - **entity_type:** One of the entity types from the provided list (e.g., [Person, Organization, Location, Event, Product, Concept, Time]). - **entity_description:** A comprehensive description of the entity's attributes and activities. Format each entity output as a JSON entry, following the structure: {{"name": <entity name>, "type": <type>, "description": <entity description>}} 2. From the entities identified in step 1, identify all pairs of (source_entity, target_entity) that are clearly related. For each related entity pair, extract the following details: - **source_entity:** The name of the source entity as identified in step 1. - **target_entity:** The name of the target entity as identified in step 1. - **relationship_description:** An explanation of why the source entity and the target entity are related. - **relationship_strength:** An integer score between 1 and 10, indicating the strength of the relationship between the source and target entity. Format each relationship as a JSON entry, following the structure: {{"source": <source_entity>, "target": <target_entity>, "relationship": <relationship_description>, "relationship_strength": <relationship_strength>}} 3. Return the results as a single list of JSON entities and relationships identified in steps 1 and 2, in English.
Report Generation	**Goal:** Write a comprehensive assessment report of a community as a community analyst. The report should provide an overview of the community's key entities and their relationships. **Report Structure:** The report should include the following sections: - **TITLE:** The community's name representing its key entities. The title should be short but specific and, when possible, include representative named entities. - **SUMMARY:** An executive summary of the community's overall structure, how its entities are related to each other, and key points associated with these entities. - **DETAILED FINDINGS:** A list of 5-10 key insights about the community. Each insight should include a short summary, followed by explanatory paragraphs grounded with concrete examples.
Answer Generation	**Goal:** Generate a response to the following question based on the provided report: Question: {question} **Instructions:** - Use the data provided in the report below as the primary context for generating the response. - If you don't know the answer or if the input report does not contain sufficient information, respond with: "Information not found in the report." - Provide the **id** of the findings used to generate your response.