SUSTech Southern University of Science and Technology

ToT-Prompt Optimization for LLM-based Automatic Summarization

Reporter: Zhao Zhao Xiaoqi Qiu

Date: 2025/12/10

SUSTech Southern University of Science and Technology

CATALOGUE

CATALOGUE

SUSTech Southern University of Science and Technology

01

Introduction

Introduction

SUSTech Southern University of Science and Technology

📌 Core Problem

Select and arrange prompt words, maximize the quality of summaries generated by LLMs

🔍 Problem Nature

- Combinatorial Optimization Problem
- Huge Search Space: n components → O(n!) possible permutations and combinations
- High Evaluation Cost

Introduction

Practical Application Background

- Financial Analysis:

  Efficiently process financial news, corporate reports

- Scientific Research:

  Accelerating knowledge extraction and integration

- Law & Healthcare:

  Quickly condense legal cases, generate medical record summaries

02

Formulization

Formalization of the Problem

Find an optimal ordered sequence from n prompt components within a limited evaluation budget.

Maximize: $Q(\sigma)$

where $Q(\sigma) = f_{comp}(\sigma) + f_{order}(\sigma) - p(\sigma)$ Component Base Score, Order Synergy Effect, Penalty Terms

Decision variables: $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_k) \in \Pi_k(C)$

Let $C = c_1, c_2, \ldots, c_n$ be the set of all available prompt components.

The decision variable is an ordered sequence:

$$\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_k)$$

where:

$k \leq n$: sequence length (variable)

$\sigma_i \in C$: component ID at position i in the sequence

$\Pi_k(C)$: set of all permutations of k elements from C

Constraints

1. Dependency Constraints

For each component $c \in C$, let its dependency set be $D(c) \subseteq C$:

$$\forall c \in \sigma, \forall d \in D(c) : d \in \sigma \text{ and } pos(d) < pos(c)$$

where $\text{pos}(x)$ represents the position index of component x in sequence σ.

2. Mutual Exclusion Constraints

For mutually exclusive component pairs $(c, d) \in X$ (where X is the mutual exclusion relation set):

$$c \notin \sigma \text{ or } d \notin \sigma$$

Constraints

3. Length Constraints

Let $t(c)$ be the token count of component c, and $t_{\text{conn}}(c_i, c_j)$ be the connection token count between components:

$$\sum_{i=1}^{k} t(\sigma_i) + \sum_{i=1}^{k-1} t_{conn}(\sigma_i, \sigma_{i+1}) \leq T_{max}$$

4. Position Constraints

For components with position requirements:

Start components: $c \in C_{\text{start}} \Rightarrow \text{pos}(c) = 1$

End components: $c \in C_{\text{end}} \Rightarrow \text{pos}(c) = k$

Middle components: $c \in C_{\text{middle}} \Rightarrow 1 < \text{pos}(c) < k$

NP-Hard Proof

Theorem: The prompt sequence optimization problem is NP-hard.

Proof by reduction from Directed Hamiltonian Path (DHP):

Given: A directed graph $G = (V, E)$ with $|V| = n$ vertices.

Construct an instance of our problem:

Each vertex $v_i \in V$ corresponds to a prompt component c_i

For each edge $(v_i, v_j) \in E$, set synergy $s(c_i, c_j) = 1$

For non-edges $(v_i, v_j) \notin E$, set $s(c_i, c_j) = -\infty$

Set all component base scores $w(c_i) = 0$

Set length constraint to accept all sequences

Set no dependency or mutual exclusion constraints

NP-Hard Proof

Claim: There exists a Hamiltonian path in G if and only if there exists a sequence σ of length n with $Q(\sigma) = n - 1$.

Proof:

($\Rightarrow$) If G has Hamiltonian path $v_1 \to v_2 \to ... \to v_n$, then sequence $\sigma = (c_1, c_2, ..., c_n)$ has $f_{order}(\sigma) = n - 1$, so $Q(\sigma) = n - 1$.

($\Leftarrow$) If sequence σ has $Q(\sigma) = n - 1$, then all n components appear exactly once and all consecutive pairs have synergy 1, meaning they correspond to edges in G, forming a Hamiltonian path.

Since DHP is NP-hard and this reduction is polynomial-time, our problem is NP-hard.

Corollary: The optimization version (finding max $Q(\sigma)$) is NP-hard.

State Space Complexity

Search Space

The state space consists of all ordered sequences satisfying constraints:

$$S = \{\sigma \in \cup_{k=0}^{n} \Pi_k(C) \mid \sigma \text{ satisfies all constraints}\}$$

State space size:

$$|S| = \sum_{k=0}^{n} P(n,k)$$

where

$$P(n,k) = \frac{n!}{(n-k)!}$$

State Space Complexity

Theoretical Upper Bound

$$|S_{theoretical}| = \sum_{k=0}^{n} P(n,k) = \sum_{k=0}^{n} \frac{n!}{(n-k)!}$$

Asymptotic complexity: $O(n!)$

Practical

$$|S_{practical}| \approx O(\frac{n!}{2^d})$$

SUSTech Southern University of Science and Technology

03

Method Design：ToT Prompt Optimizer

Design Rationale

SUSTech Southern University of Science and Technology

Why heuristic solution?

Problem: Prompt optimization is a High-Dimensional, Discrete Combinatorial Optimization problem.

Design Rationale

SUSTech Southern University of Science and Technology

Why heuristic solution?

Problem: Prompt optimization is a High-Dimensional, Discrete Combinatorial Optimization problem.

Problem Challenges:

- **Combinatorial Explosion** ($O(2^N)$): Exhaustive search is intractable, requiring effective **pruning.**
- **High Evaluation Cost**: Objective function evaluation (Full Evaluation) is **costly, time-consuming**, **non-convex, and non-differentiable** (precluding gradient-based methods)
- **Discreteness & Constraints**: Component selection is binary/discrete, with predefined **conflict constraints**, requiring methods that support **step-wise state construction** and **validity checks**.

Design Rationale

SUSTech Southern University of Science and Technology

Why heuristic solution?

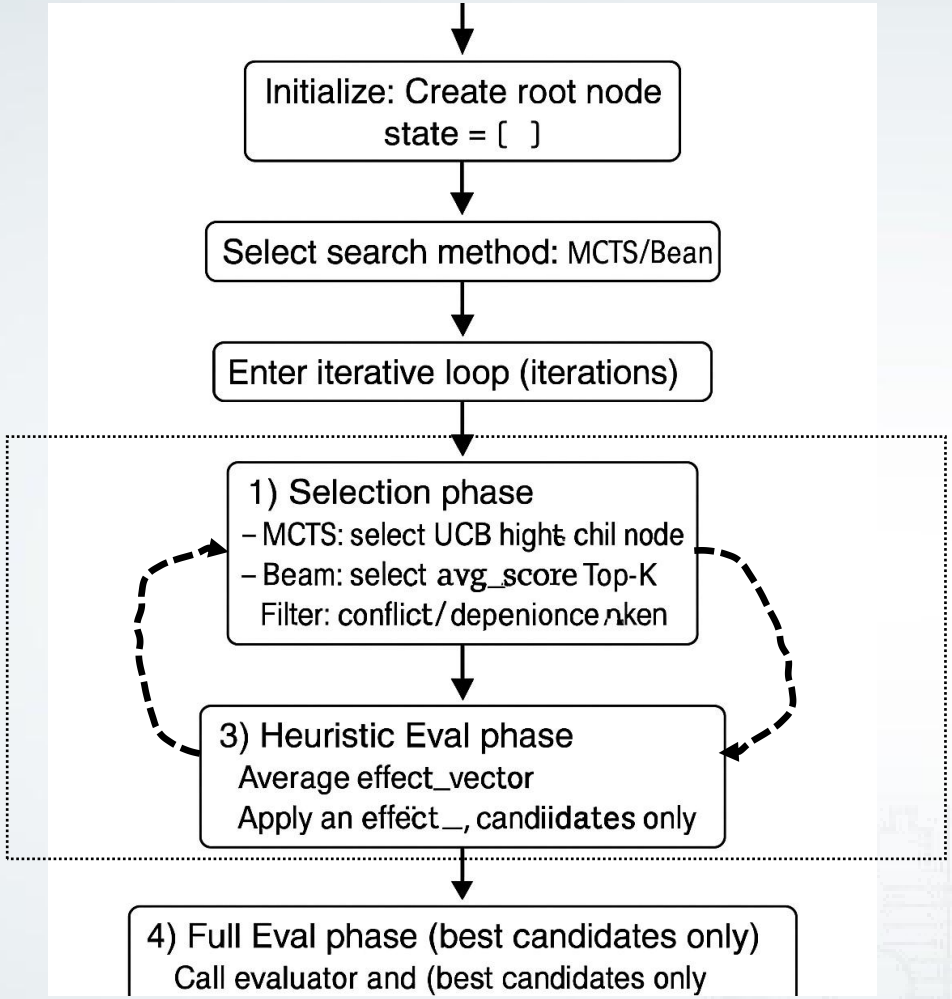
Justification for Metaheuristic Search

- **Monte Carlo Tree Search** (MCTS): Strategically balances exploration (new sets) and exploitation (high-performers) using **upper bound score**, ensuring costly Full Evaluations are allocated to the most promising states
- **Beam Search**: Offers a faster, greedy complement for rapid convergence to local optimums by retaining only the top k states at each step

Key Feature: Learning-to-Optimize (L2O): Incorporates a lightweight Heuristic Evaluation (predictive assessment) to guide the search.

ToT Prompt Optimizer

SUSTech Southern University of Science and Technology



ToT Prompt Optimizer

SUSTech Southern University of Science and Technology

Core Structure and State Representation

- State: A valid set of Prompt components.
- Node: A position within the search tree, storing a state S and search statistics.

```python
@dataclass
class SearchNode:
    """搜索树节点"""
    node_id: str
    state: List[str]  # 组件ID列表
    parent_id: Optional[str]
    depth: int
    visits: int = 0
    total_score: float = 0.0
    heuristic_score: float = 0.0
    full_evaluation_score: float = 0.0
    children_ids: List[str] = None
    is_fully_evaluated: bool = False
    metadata: Dict[str, Any] = None
```

Monte Carlo Tree Search (MCTS)

SUSTech Southern University of Science and Technology

MCTS guides the search through iterative sampling, executing four steps per iteration :

- **Selection**: Starting from the root, nodes are recursively selected based on the highest heuristic score
- **Expansion**: Valid candidate child states (enforcing **conflict avoidance** via generate_candidates) are generated from the selected node. New child nodes are created and immediately undergo Heuristic Evaluation (evaluate_state_heuristic).
- **Evaluation** (Simulation/Rollout): The most promising nodes (e.g., $\text{avg\_score} > \text{best\_score} \times 0.7$) are subjected to a costly Full Evaluation (**evaluate_state_full**) to obtain the accurate full_score.
- **Backpropagation**

Heuristic Scoring Function

SUSTech Southern University of Science and Technology

Purpose: Provide a **lightweight approximation** of prompt quality to guide the search efficiently, avoiding expensive full LLM evaluations at every node.

Formula: Weighted sum of five metrics $\in [0, 1]$:

$$H = \sum_{m \in M} w_m \cdot S_m$$

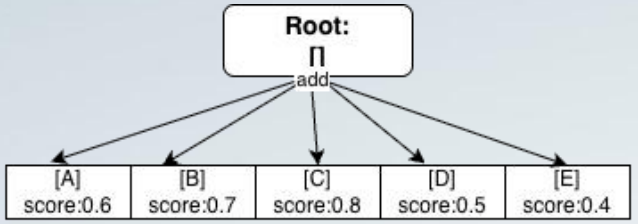
- M = {Faithfulness, Conciseness, Completeness, Readability, Insightfulness}
- Pre-defined Weights

Metric Calculation Strategy: Balances speed and reliability

Rule-Based Metrics (Fast Heuristics): **Conciseness** on word count, **Completeness** on keyword coverage, **Readability** on average sentence length

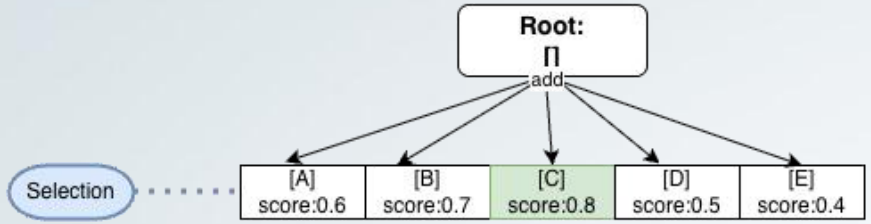
LLM-Based Metrics (Accurate Heuristics): Faithfulness and Insightfulness

Given 5 non-conflict componments: [A B C D E]



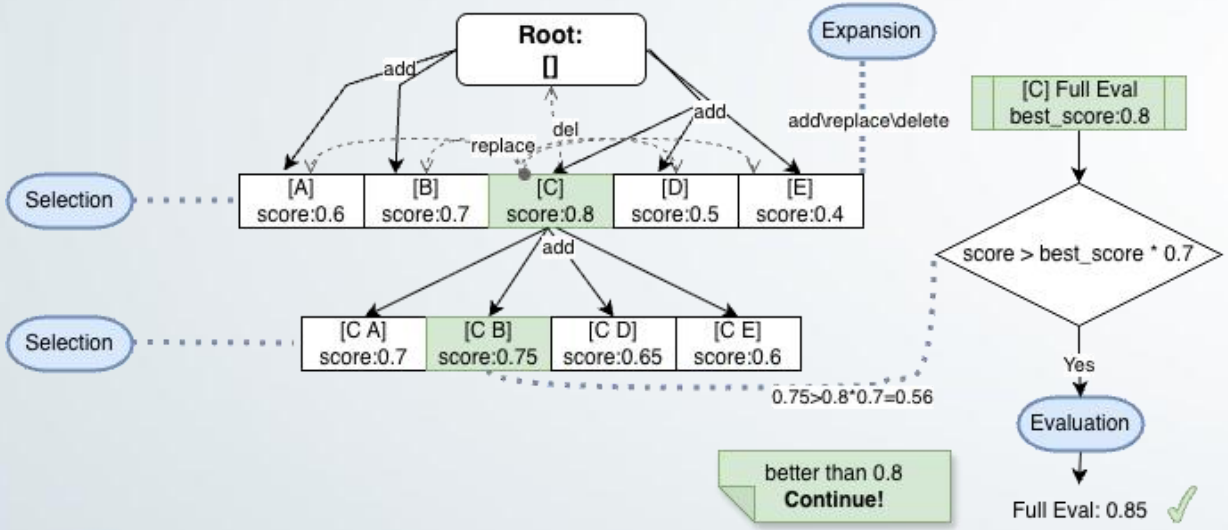
(1)

[] Full Eval
best_score:0

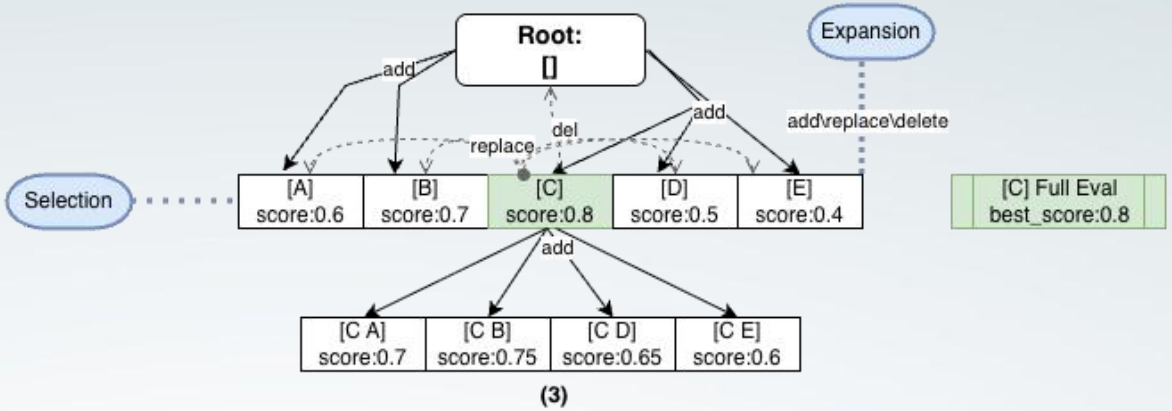


(2)

[C] Full Eval
best_score:0.8

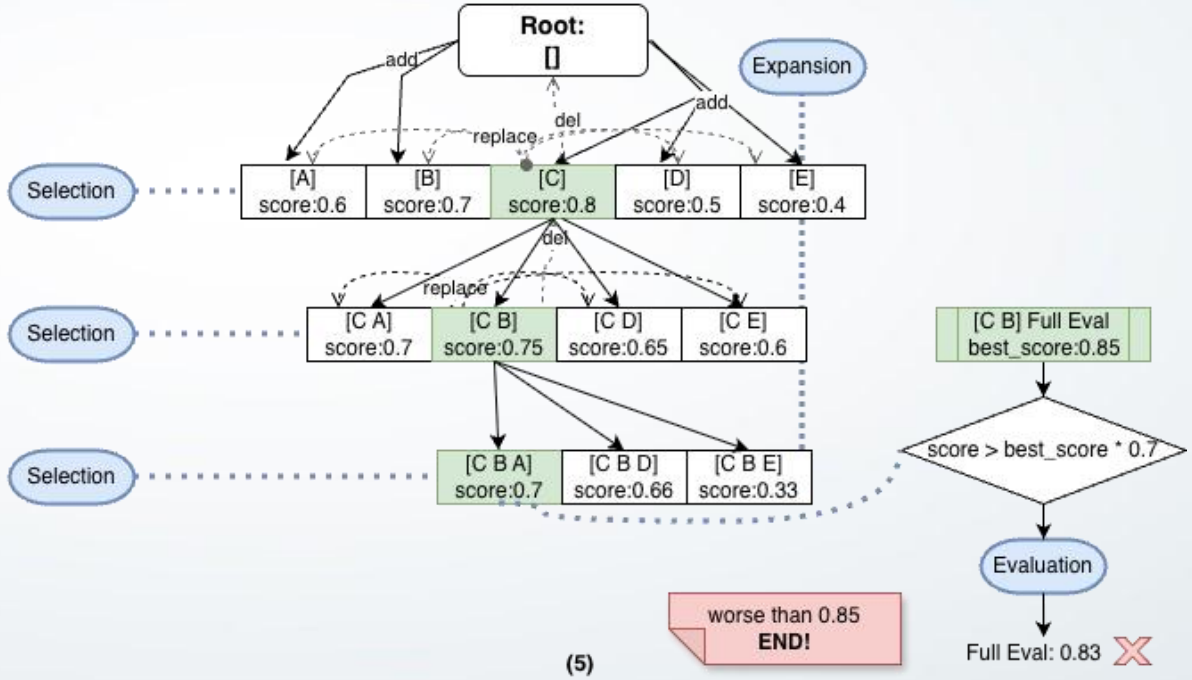


(4)



(3)

[C] Full Eval
best_score:0.8



(5)

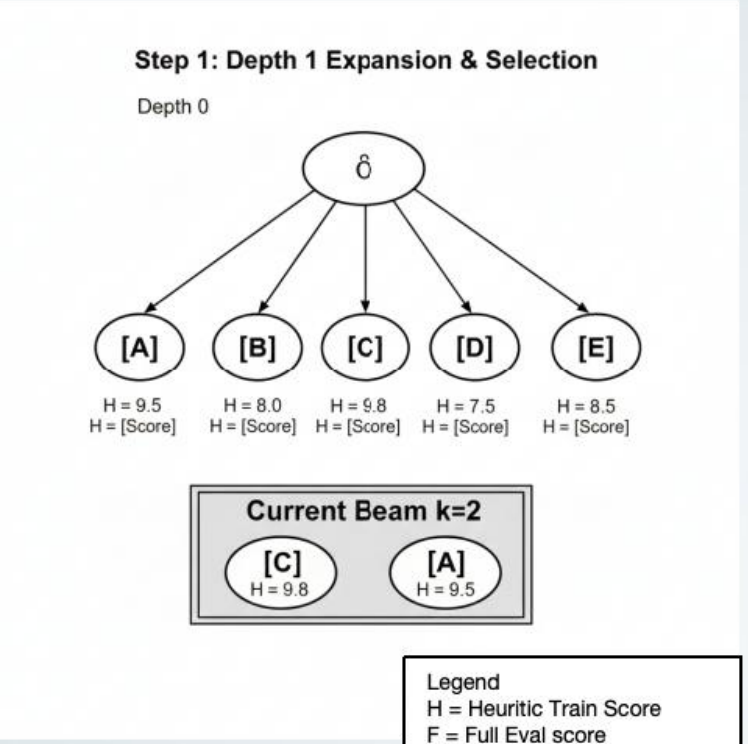
Beam Search

SUSTech Southern University of Science and Technology

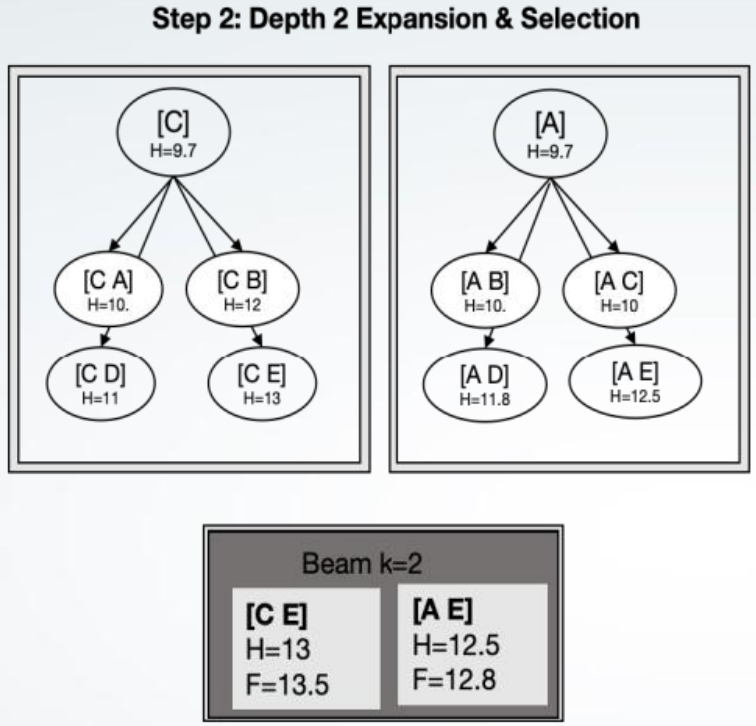
Beam Search implements a more greedy heuristic approach :

- **Breadth-First Pruning:** In each iteration, the **top-k** scoring nodes (the beam) from all existing nodes are selected.
- **Expansion and Heuristics:** All **new candidate states** are generated from the k beam nodes and assessed using Heuristic Evaluation.
- **Evaluation**: The new candidates are ranked by their Heuristic Score, and only the top k are selected to receive a Full Evaluation. This ensures a fast, concentrated search.

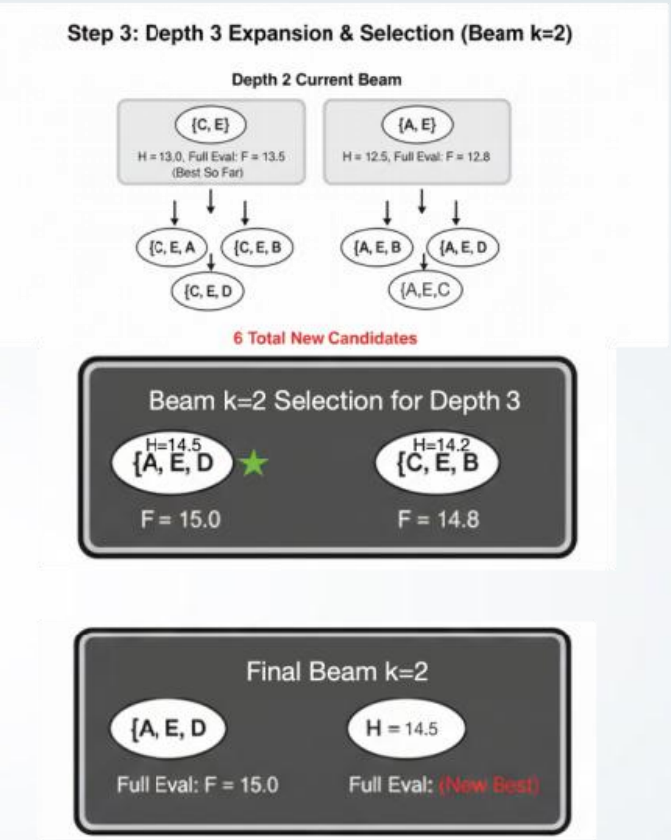
Given 5 non-conflict componments: [A B C D E]



(1)



(2)



(3)

Complexity Analysis

SUSTech Southern University of Science and Technology

Per-State Operations

- Constraint Checking (Dependency, Mutual Exclusion, Length, Position): Total complexity $\mathrm{O(k)}$ with optimization, where k is the sequence length
- Quality Evaluation $O(\sigma)$: $\mathrm{O(k)}$ for component base score, order synergy, and penalty terms

Algorithm	Per Iteration/Layer Complexity (General)	Worst Case Asymptotic Complexity
MCTS	$O(I(d \cdot \log b + b + C_e))$	$\mathcal{O}(I \cdot n)$
Beam Search	$O(d \cdot w \cdot b \cdot \log(w \cdot b))$	$\mathcal{O}(n^2 \log n)$

Complexity Analysis

SUSTech Southern University of Science and Technology

Search Algorithm Complexities

- Variables: I = iterations, d = depth, b= branching factor (avg.), C_e = evaluation cost, w = beam width, n = total components
- Efficiency: Both MCTS and Beam Search are significantly more efficient than the theoretical $\mathrm{O(n!)}$ brute-force search space

Algorithm	Per Iteration/Layer Complexity (General)	Worst Case Asymptotic Complexity
MCTS	$O(I(d \cdot \log b + b + C_e))$	$\mathcal{O}(I \cdot n)$
Beam Search	$O(d \cdot w \cdot b \cdot \log(w \cdot b))$	$\mathcal{O}(n^2 \log n)$

SUSTech Southern University of Science and Technology

04

Experimental Results and Discussion

Experiments

Dataset → Summarization Task

Instead of complex formats (URLs, PDFs) or dialogue-style content。

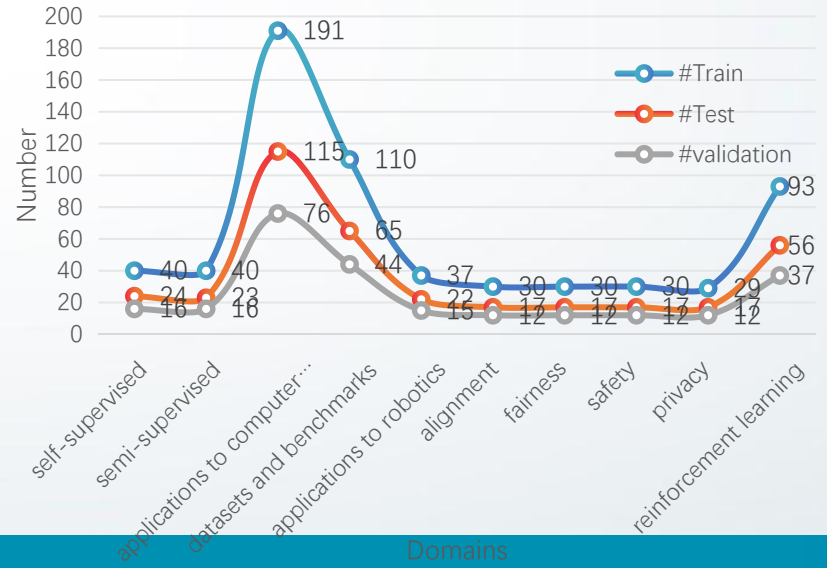
A clean, plain-text dataset was manually constructed for the experiments.

Source:

- Summarization： ICLR 2026 OpenReview platform
- Prompt Components： Human designed 20 Entries

Table 5: Statistics of word length across dataset splits.

Category	Content Length	Summary Length
Training Dataset	182.82	26.94
Test Dataset	184.56	26.50
Validation Dataset	174.90	26.35



```
{
"paper_id":"bU8tRjuanU",
"title":"Low-Rank Attention and Contrastive Alignment for Deep Multi-View Clustering",
"domain":"self-supervised",
"content":"Recent years have witnessed significant advancements in deep multi-view clustering (MVC). However, prevailing methods exhibit three critical limitations: (1) poor scalability for large-scale datasets, (2) neglect of anchor semantic consistency in feature alignment, and (3) inability to capture high-order feature interactions. To overcome these challenges, we propose a Low-Rank Attention and Contrastive Alignment framework (LRACA). Unlike conventional approaches that align sample-level features in shared subspaces, LRACA employs a category-aware anchor generation module to directly align high-level semantic prototypes (i.e., category centers) across views, explicitly enforcing clustering semantic consistency. Furthermore, we devise a dynamic low-rank attention mechanism to enhance feature discriminability, where entropy regularization constrains attention weight distributions to derive clustering pseudo-labels. Finally, a pseudo-label-guided cluster-level contrastive learning module maximizes cross-view mutual information through a feed-forward optimization paradigm. Extensive experiments on six large-scale multi-view datasets demonstrate that LRACA significantly outperforms state-of-the-art methods.",
"gold_summary":"This paper proposes a deep multi-view clustering algorithm equipped with an anchor-based attention mechanism and cluster-based contrastive learning, which achieves good performance, but has many problems, as detailed in the weaknesses section."
},
```


(a) An Example of Data Entry


```
{
"id":"conciseness",
"text":"Please use concise language",
"description":"Requires concise output",
"effect_vector":{
"conciseness":0.8,
"readability":0.6,
"completeness":-0.2
},
"conflicts":[
"detail_oriented",
"comprehensive"
],
"requires":[],
"estimated_tokens":8
},
```


(b) An Example of Prompt Component Entry

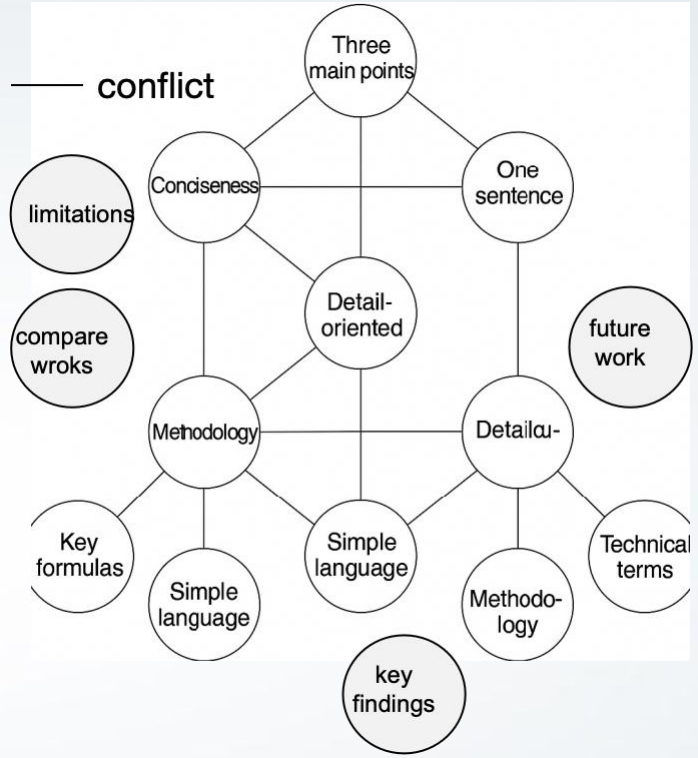


Fig. 4: Data Entry Statistic.

Setup

SUSTech Southern University of Science and Technology

Models:

- Base Model (for ToT Training): Qwen3-1.7B.
- Scoring Model (Discriminator): Qwen3-8B

Baselines：

- **SimplePrompt**: “Summary this paper”
- **DetailedPrompt**: “A prompt asking for core content, methodology, and contributions”
- **StructuredPrompt**: “A prompt asking to summarize from four aspects: background, methodology, experiments, and conclusions”

Metrics:

- **N-gram Static Metrics**: ROUGE-2 F1 and ROUGE-L F1
- **Semantic Similarity Metrics:** BERTScore F1 (using "bert-base-multilingual-cased").
- **LLM Evaluation Score**: Heuristic Scoring

Experimental Results and Discussion

SUSTech Southern University of Science and Technology

Overall Performance

- **ToT Prompt-Beam's Success:** This strategy consistently achieved the highest scores across all reported metrics
- MCTS vs. Beam Search:

  ToTPrompt-MCTS < **ToTPrompt-Beam**

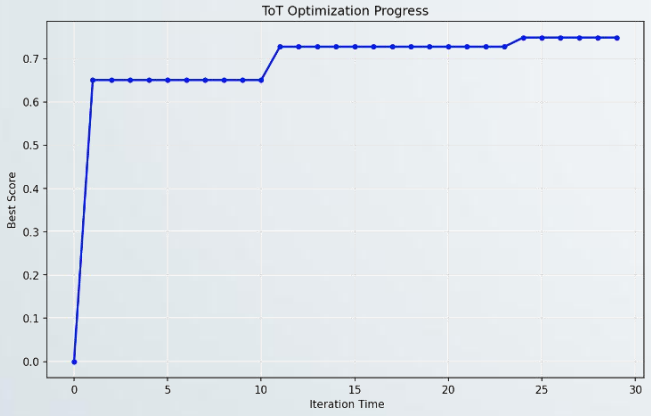
Prompt Strategy	Heuristic Score ↑	ROUGE-2 ↑	ROUGE-L ↑	BERTScore ↑
SimplePrompt	0.51 ± 0.10	0.03 ± 0.02	0.03 ± 0.02	0.05 ± 0.13
DetailedPrompt	0.58 ± 0.06	0.02 ± 0.01	0.07 ± 0.02	-0.01 ± 0.04
StructuredPrompt	0.58 ± 0.04	0.01 ± 0.01	0.06 ± 0.01	-0.18 ± 0.20
ToT-MCTS(ours)	$0.61 + 0.00$	0.03 ± 0.02	0.09 ± 0.01	$0.06 + 0.04$
ToT-Beam(ours)	$\mathbf{0.67 \pm 0.02}$	$\mathbf{0.07 \pm 0.05}$	$\mathbf{0.13 \pm 0.05}$	$\mathbf{0.16 \pm 0.06}$

Convergence Analysis

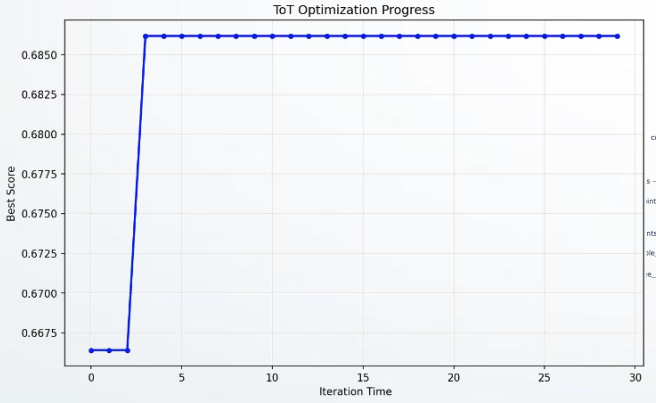
Scores quickly converged in both strategies. Beam Search converged significantly faster than MCTS due to its greedy-like selection at each step

Potential Reason for Rapid Convergence

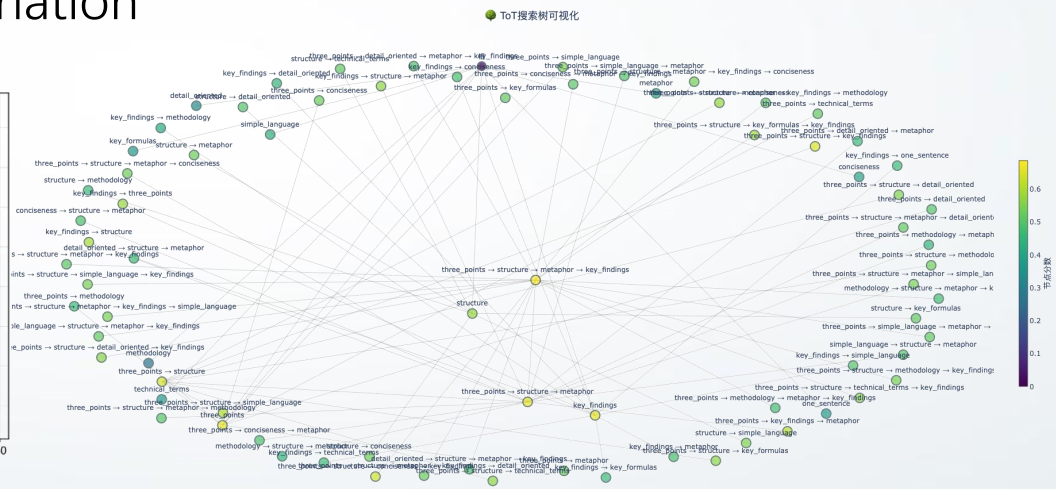
- an insufficiently rich set of prompt components,
- a small candidate set
- a small training batch size leading to minimal score variation



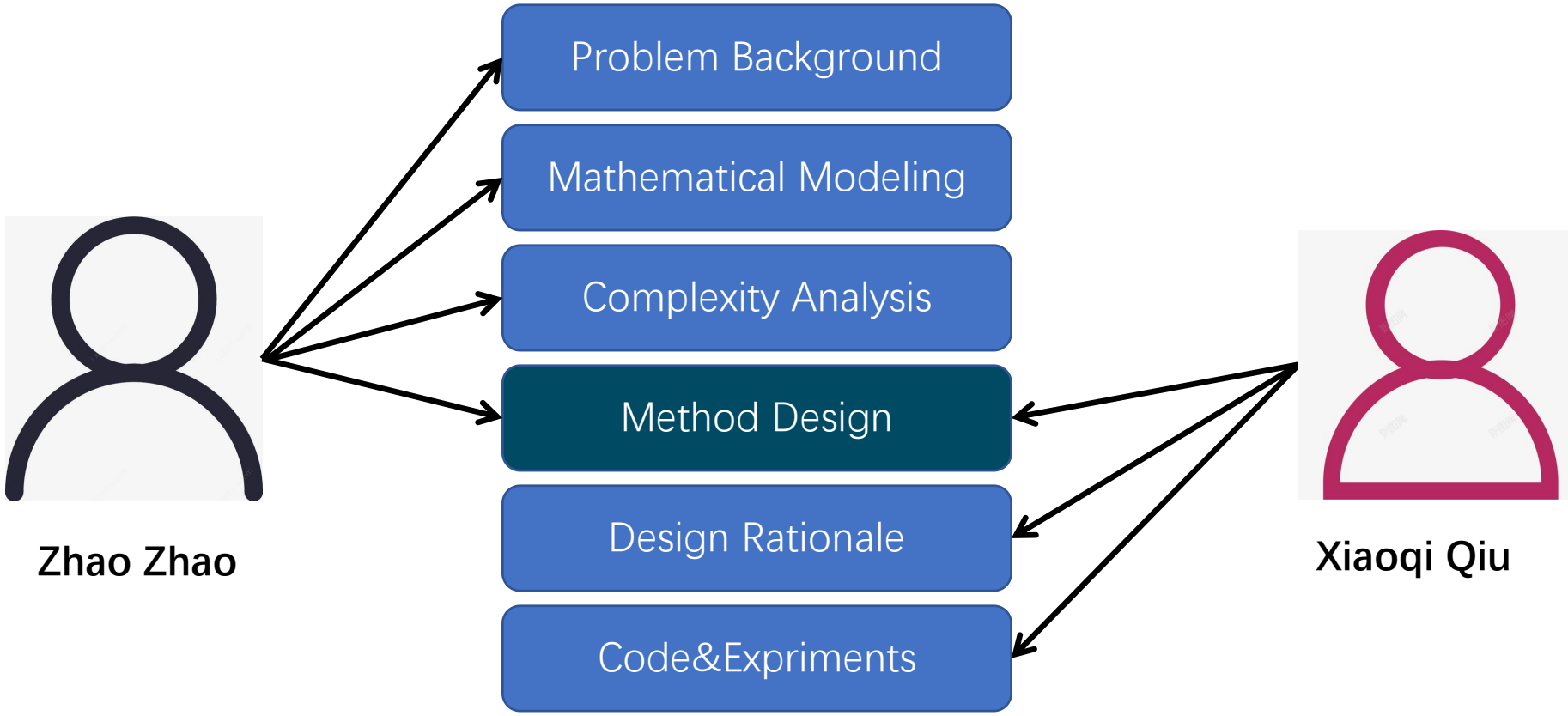
(a) Optimization Progress of MCTS Search



(b) Optimization Progress of Beam Search



Division of Lablor



Code&Dataset:
git@github.com:Siki-cloud/ToT_Optimizer_for_LLMSum.git

SUSTech Southern University of Science and Technology

“ToT-Prompt Optimization for LLM-based Automatic Summarization”

Thank you

Reporter: Zhaozhao Xiaoqi Qiu

Date: 2025/12/10