BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

PROMPT ENGINEERING TECHNIQUES IN LARGE LANGUAGE MODELS : *A Comparative Study*

[1]Vedika Chaudhari, [2]Prof. Supriya D. Survase

[1]Student, TE (A), PRN 72426653G, [2]Technical Seminar Mentor
[1,2]Department of Artificial Intelligence and Data Science
[1,2]Ajeenkya DY Patil School of Engineering, Pune, Maharashtra, India
[1]Email: vedikachaudhari3105@gmail.com, [2]Email: supriyasurvase@dypic.in

Abstract

Even though LLMs can solve different kinds of tasks without training, the quality of the answer depends on the formulation of the request. Prompt Engineering is the art of formulating a question right. There are many ways to do it, they differ in terms of their design, cost, use cases and computational requirements, thus it may be challenging for a beginner to choose the right one. In this paper, we compare and contrast seven of them: Zero-shot Prompting, Few-shot Prompting, Chain of Thought (CoT), Tree of Thoughts (ToT), Graph of Thoughts (GoT), Self-Refine, and Retrieval-Augmented Generation (RAG). All of them will be described in the same categories: idea, example, pros and cons, uses, computational requirements. No experiments have been conducted, however, the paper is based on the literature review. All the information comes from original sources. They can not be compared directly since different models and methods were used in different papers. The literature suggests that reasoning steps enumeration aids to solve multi-step problems if the model is big enough. The tree and graph approaches prove to be helpful if one thought is not enough; however, they are invoked much more often in that case. Self feedback improves the answer in some cases but not in others. The external knowledge in RAG comes in addition to formulating a prompt. The last part of the paper includes a brief comparison between the techniques, the costs associated with the use of each one and gaps in the research.

Keywords: Prompt engineering, Zero-shot prompting, Few-shot prompting, Chain of thought, Tree of thought, Graph of thought, Self-refine, Retrieval-augmented generation (RAG)

Introduction

There was very little chance that a person would have foreseen a few years back that a computer could write an excellent paragraph. All of this is made possible by the Large Language Models. They are trained on a vast corpus of text and use the Transformer architecture [1]. GPT-3 [2] and InstructGPT [3] can do summarization, translation, question answering, coding, and reasoning, whereas LLaMA [4] can do summarization and translation.

Whatever we give as an input to the LLM is called the prompt. It comprises an instruction, background context, information/data that needs processing and at times even solved instances. Brown et al. showed that it was possible for a trained model to learn a new task based solely on the input prompt without making any adjustments in its weights [2]. This is why the way a prompt is phrased becomes crucial. It is the quality

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

of the prompt which could be the cause of erroneous, inconsistent or even fabricated responses [5]. The art of crafting such prompts without changing the model itself has become a topic of research on its own and is termed Prompt Engineering [6]. There are already plenty of prompting approaches in the literature [7], [8]. We did not try to present all of them and chose only seven. All of them stem from the preceding approach. The distinction between zero-shot and few-shot prompting is the fact that fewer task details are provided for the first one than for the second. CoT, ToT, and GoT exhibit one particular form of reasoning in three different ways – chain, tree, and graph. Self-Refine occurs after an answer has been generated. RAG serves as an external knowledge source. In the case of each of these, there was one paper that used each model, test or score differently. It is not easy for a student to understand the capabilities of each technique, its costs and the situations where it would be best to use it. This paper tries to bridge that gap.

This paper attempts to:

- define the prompt and the importance of its quality;
- explain all seven techniques in a unified fashion;
- compare them in terms of their logic, informational and computational cost;
- summarize the research, the strengths and weakness and research gaps; and
- highlight areas for future work.

Seven techniques are discussed in this paper using the existing literature. There are no new experiments. In addition, the paper provides (i) an explanation of the seven techniques for the benefit of undergraduates in a unified fashion; (ii) a comparison table and analysis of costs; (iii) a summary of results along with the caution necessary; and (iv) research gaps.

The structure of the current paper is as follows. Background and Related Work includes the background. The Seven Techniques covers the seven techniques, and Comparison of Approaches is about their comparison. Methodology provides details on the method, while Results and Discussion provides the results. Gaps in Research and Future Works is about the gaps and future work; Limitations is about the limitations and Conclusion concludes the paper.

Background and Related Work

A typical prompt consists of an instruction, background information (optional), input data, and recommendation for the output format. The model may get some example prompts so that it could learn the task during the process. The model receives no updated weights. Brown et al. called this approach in-context learning and delineated it and other learning approaches in terms of zero-shot, one-shot and few-shot learning [2]. Instruction tuning and learning from human feedback may help in making models learn how to obey written instructions [3]. The shift in language processing to “pre-train, prompt and predict” is described by Liu et al. [6]. Prompts can also be learned instead of written by hand: continuous prompt tuning [28], prefix-tuning [29] and automatically generated discrete prompts [34] optimise the prompt itself, and language models can even propose and select instructions on their own [27]. Instruction finetuning, as in FLAN [33] and its scaled successors [35], further improves zero-shot behaviour.

This form of prompting is making the model to go straight from the question to the answer without any intermediate steps. Such prompts are called plain prompts. They fail when working on tasks requiring many

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

sequential steps, such as a maths word problem. As revealed by Wei et al., it has been shown that if a large model generates intermediate steps for itself, solving such tasks becomes much simpler [9]. Further research built the chain into a tree [10] and later generalized the chain into a graph of partial results [11].

The language generated by the LLM can be also completely wrong. This effect is called hallucinations [5]. All knowledge of the model is stored in a fixed set of parameters learned at the time of the training process. The approach of RAG consists of the integration of a parametric language generator and a non-parametric memory module which is used during answering a question [12]. Dense retrievers are responsible for retrieving useful text fragments [13].

Liu et al. provide an overview of prompts for language processing [6]. The classification of many methods into several categories is provided by Sahoo et al. [7] and Schulhoff et al. [8], while Gao et al. [14] investigate the research area of RAG. Edemacu and Wu review the issue of privacy in prompt engineering [15]. This literature is meant to be quite general. In our paper, we go more specific since we consider seven methods, introduce them in the same way and discuss them in the same aspects including costs.

The Seven Techniques

The structure of all techniques is as follows: meaning and purpose, concept and principle, first example, advantages, drawbacks, computational requirements, applications, findings in literature.

Zero-shot Prompting

Meaning and purpose. Here the task instruction and input data are provided, but there is no example [2]. The principle is to use pre-training and instruction tuning, which means that prompt generation is not complicated [3].

Working idea. Instruction is used as a condition and the model is asked to generate the most likely continuation based on it. It will depend on the quality of the instruction and what is expected in the output, limits etc.

Example. "What is the sentiment of this review? Positive, Negative or Neutral? Review: The campus Wi-Fi keeps dropping during lab hours. Sentiment:" An ideal model should give the answer "Negative".

Advantages. Requires no examples, prompt is short and easy to try.

Restrictions. The results may vary based on the wording. The output format may also vary and the tasks which consist of many steps or have an unusual format are usually not well done.

Computing need. Only one call with a short prompt (the cheapest of the seven).

Uses. Classification, summarization, translation, drafting and general questions.

Findings. Brown et al. examined the performance of GPT-3 on a variety of benchmarks. For all the settings the results were higher for larger models. In general, zero-shot was outperformed by few-shot and the difference increased with the size of the model [2].

Few-shot Prompting

Meaning and purpose. The prompt includes some example pairs of inputs and outputs before the new input

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

and the model uses those pairs to deduce what the task is for the new input [2]. This way, the task, labels, and output format can be controlled without training.

Working principle. The model understands the examples provided, takes the next question and does the same thing to it. According to Min et al., the key factors for this technique are the label space, the nature of the input and the format of the examples. The correctness of the labels is not as important as one might expect [16]. The choice of the examples [30] and their order [31] also change the results considerably.

Example. Review: "Lab computers are fast." → Positive. Review: "Library closes too early." → Negative. Review: "The seminar was on Tuesday." → Neutral. Review: "The new canteen menu is excellent." → ? The examples provide the labels, as well as the format of the output.

Advantages. Unlike zero-shot, it is easier to control format and style. High accuracy is ensured and there is no need for training.

Chain of Thought (CoT)

Working idea. The model writes down each step of its actions and embeds it into the context for the next step. The task is divided and receives more attention from the model. The solution is linked to the written chain of steps rather than a single guess (Fig. 1b).

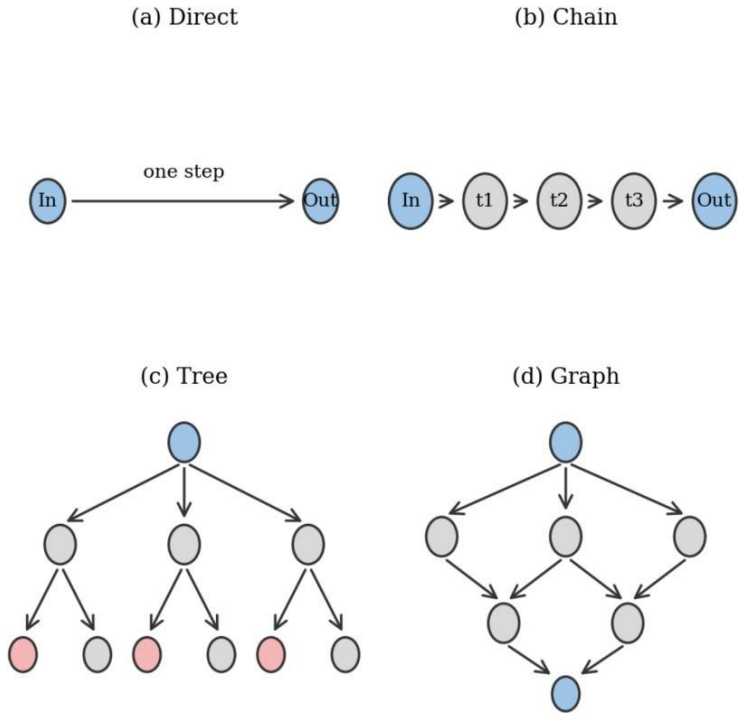


Figure 1: Direct, chain, tree with cut branches (red) and graph with merging shapes of reasoning. Original illustration.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

Example. There are 4 desks in a lab, and each desk has 6 computers on it. 5 computers are being repaired. How many computers work? A: 4 × 6 = 24. 24 − 5 = 19. The answer is 19. Zero-shot variant contains the question and "Let's think step by step" only.

Advantages. Reasoning becomes better, it is possible to verify the chain of actions of the model, and there is no need to train the model. The larger the model, the greater the improvement, as mentioned by Wei et al. [9].

Limitations. Even though the chain might seem convincing, it does not have to be correct. Chain is non-reversible and zero-shot CoT does not have to be reliable in difficult tasks [17]. The two concepts that have some connection to that include self-consistency which samples several chains and keeps the answer that was reached by most of them [22], and Auto-CoT, which automatically creates worked-out examples for automatic instead of hand-written examples [23].

Computational requirements. Normally one request; longer output and larger chain increases cost.

Applications. Mathematical and logic problems, commons sense questions, educational assistance in form of steps for solutions and explanation of how the code works.

Findings. Wei et al. reached state-of-the-art accuracy with PaLM 540B and eight CoT examples on math benchmark GSM8K [9]. For larger InstructGPT, Kojima et al. noted the improvement in accuracy from 17.7% to 78.7% for MultiArith and from 10.4% to 40.7% for GSM8K with zero-shot trigger phrase [17].

Tree of Thoughts (ToT)

ToT is an evolution of CoT. In between, there is an intermediate step called the "thought" which is added to the tree as a node. The model tries various candidate thoughts, evaluates them and then plans ahead and backward [10]. This method is useful in case of complicated problem solving, since one incorrect thought in one chain can destroy the whole process.

Working idea. The algorithm consists of four components. Firstly, the problem is decomposed into stages; then there are multiple candidate thoughts per stage and every state is evaluated (the model labels the state as "promising" or "impossible"). After that, the search procedure (breadth-first or depth-first) selects the branches to grow and the branches to prune (Fig. 1c) [10].

Example. Four speakers should be distributed among three slots under room requirements and availability constraints. There are certain ways of filling the first slot proposed by the model; every thought is verified and good thoughts are selected and bad thoughts (which do not correspond to the slot requirements) are rejected.

Advantages. To plan, compare alternatives, and correct errors; search history is available for review.

Limitations. According to Yao et al., the ToT approach requires more resources than the simple sampling approach and in some cases, a good model does not need ToT [10]. For each task, it requires breaking down

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

the task, prompting the judgment, setting up the search and the judgment provided by the model may not be accurate.

Computing requirements. A lot of computing for a single task. The number of computations required depends on the number of candidates generated and evaluated at each node, the width of the search and the depth of the search tree.

Applications. Puzzles, planning, scheduling, constraint satisfaction problems and story planning.

Results. Yao et al. found that GPT-4 using ToT approach was able to solve 74% of the tasks in Game of 24 while CoT solved only 4% [10].

Graph of Thoughts (GoT)

The GoT approach models the reasoning process in a directed graph where nodes correspond to thoughts, and links correspond to the dependency between the thoughts. In contrast to a tree, a graph allows thoughts to be combined, reused, and improved in cycles [11]. It is well-suited for solving a problem that consists of several smaller subproblems whose solutions need to be merged together.

Working idea. Actions performed by a controller: generation of new thoughts, combination of multiple thoughts, refinement of a thought, and scoring of thoughts. Its graph (Fig. 1d) is defined using a plan [11].

Example. Long project report. Split it into four parts, generate summaries of each of them, merge two summaries together, and score and improve the merged summary.

Advantages. It is able to combine the best features of different reasoning paths, reuse intermediate results, and provide feedback loops.

Restrictions. It is harder to construct in comparison with CoT and ToT. Every single task needs to be graphed and the computation cost is high. There is no information about how it was tested on any task besides sorting and merging documents [11].

Computing need. The number of calls depends on the structure of the graph. Merging can avoid repeated work, yet many calls are needed for large graphs.

Applications. Document merging, sorting, set operations, multi-section reports, and task planning with sub-task dependencies.

Observations. Besta et al. reported an improvement in the sorting quality of GoT by 62% over ToT while cutting the cost by over 31% [11].

Self-Refine

Definitions and goals. Self-Refine is iterated for several rounds. The initial answer is produced by one LLM

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

and commented on and improved by it. It does not require any additional training data, supervision, nor reinforcement [18]. The goal is to do better than the first attempt.

Operating principles. The model iterates, providing feedback and making adjustments until some stopping condition is reached, e.g., maximum number of rounds or feedback that indicates there are no more changes to be made (Fig. 2). The prompts are used to provide feedback and make improvements too [18].

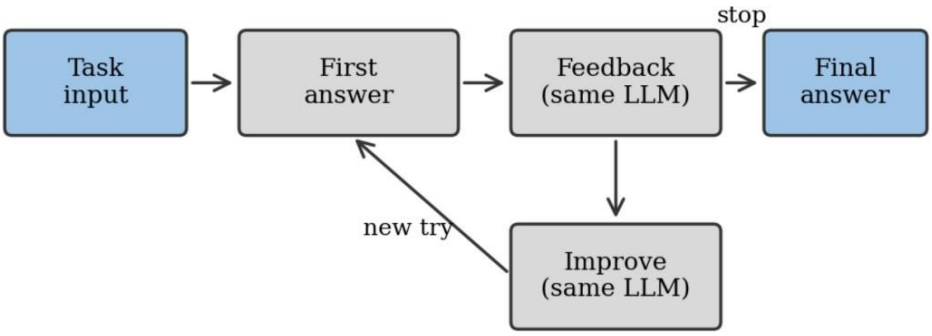


Figure 2: The Self-Refine loop: the same LLM gives the first answer, writes feedback on it and improves it until the stopping condition is met. Original illustration.

Example. A professor is asked for more time in an email. The feedback part observes that the tone is informal and no new date is suggested. In the improvement phase, it revises the email with a formal tone and specifies the date.

Pros. It is easy, uses just one model, and works every time the quality cannot be achieved in one run.

Cons. It takes more time and money every round, and the performance depends on how skilled the model is in responding to the feedback and improvement prompts [18]. A similar technique called Reflexion saves the reflections of the model in its memory for future attempts [24], while ReAct interleaves reasoning steps with actions that consult external tools [26]. Huang et al. demonstrated that it is challenging for large models to self-repair the errors in their reasoning without external intervention, and the results may become worse [19].

Need for computing. $1 + 2T$ requires T turns: one for the first attempt, T for feedback, and T for improvement. In proportion as more attempts are made, the context can grow larger, too.

Uses. Written text, replies in conversation, and quality of code or readability.

Findings. Madaan et al. conducted seven tasks using some very strong LLMs. The results were considered better than those obtained in one step, with an average gain in task performance of about 20% [18].

Retrieval-Augmented Generation (RAG)

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

Meaning and goal. RAG combines information retrieval with text generation. External knowledge source is utilized, the retriever finds relevant excerpts, and the generator produces a response to the query as well as to the excerpts returned by the retriever [12]. For that reason, it is not just a prompt formulation technique. It is a system consisting of a knowledge source, a retriever, and a generator, but when the LLM is used, the retrieved text becomes part of the prompt. It should provide the most recent and domain-specific or private knowledge and reduce ungrounded responses [14].

Basic principle. First, the documents are chunked, embedded, and indexed in the vector space. Then, once the query is received, it is embedded and the corresponding chunks are selected via dense retriever [13]. The prompt consisting of the question and passages is passed to the LLM and the instructions require to generate an answer based on the provided text (see Fig. 3). At the initial stage, RAG consisted of two types of models trained jointly, sequence-level and token-level [12]. In other approaches, multiple passages are independently read by the model and then fused in the decoder [25]. Later surveys mention naive, advanced, and modular RAG architectures [14].

Illustration. A student asks a college chatbot about the attendance requirement for the end-semester examination. The relevant passage is retrieved and added to the prompt.

Advantages. The knowledge base is updated by editing the index rather than retraining [12]. Responses are cited and hence more credible.

Limitations. The results are dependent on the retrieval process; an erroneous document could give misleading information to the model. Liu et al. noted that models depend less on information when it occurs in the middle of a lengthy text [20]. The usage of the RAG needs some configuration; using a privately managed corpus may cause privacy concerns [15].

Computational requirement. There is an initial computational cost involved in indexing the documents. After this, there is a cost for each query consisting of the query embedding, the search and the prompt containing the passages.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

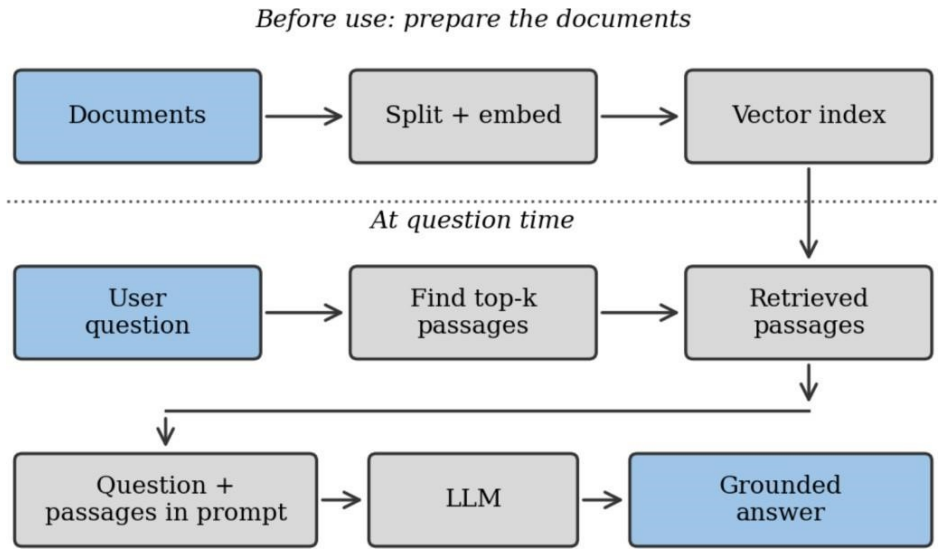


Figure 3: Pipeline of RAG: indexing before using it and retrieval and generation per question. Original illustration.

Applications. Student or customer support assistants, question answering from documents, and literature from the field of healthcare and education.

Findings. Lewis et al. obtained state-of-the-art results on three open-domain question answering benchmarks. They also found that the RAG provided more specific, diverse and factual answers as compared to a baseline model trained only on parametric information [12].

Comparison of Approaches

Table 1 gives a side-by-side comparison of the seven approaches considered, their key idea and application domain. These approaches vary on four factors, so no approach could be preferred over another. Zero-shot and few-shot approaches differ by the amount of task-specific information present in the prompt. CoT, ToT, and GoT approaches differ in the nature of reasoning that takes place, chain, tree, and graph, correspondingly. Self-Refine adds one more process of refinement of the answer. Finally, RAG uses different source of information.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

Table 1: Comparison of Seven Techniques

Approach	Key Idea	Types of Tasks and Reasoning
Zero-shot	Instruction and input alone; utilizes pre-trained model	Simple categorization, summarization, translation; no examples required
Few-shot	Instruction plus several examples; captures the pattern from context	Restricted multi-step reasoning, fixed labels, format, or style
CoT	Example or trigger instruction makes the model generate steps	Mathematical, logical, and commonsense reasoning; one way forward
ToT	Candidate thought tree generation with evaluation and backtracking search	Planning, puzzles, scheduling; explores many future possibilities ahead of time
GoT	Generation and combination of thoughts through graph operations	Combination of documents, sorting, multi-step tasks; makes use of intermediates
Self-Refine	Text generated by one model is reviewed and revised iteratively	Text, conversation, code writing; self-refinement of its own output
RAG	External passages are retrieved and answers are formed from them	Fact questions; from private or newly created documents

Methodology

It is a qualitative study of literature arranged in a framework and no new experiments were carried out for any of the models. All the methods were originally verified independently on their own models and benchmarks. Replication of the same would need access to large models which an undergraduate class does not have.

The research was based on the literature matrix constructed for the seminar synopsis and presentation. The original paper of the technique served as the primary source [2], [9], [17], [10], [11], [18], [12].
The survey papers [6], [7], [8], [14] and critical studies [16], [19], [20], [5], [21]–[25] served as further sources. We selected sources that either introduced, applied or criticized one of the seven techniques. The papers about other prompting techniques were excluded from the comparison; [26]–[35] are cited only to give background on prompt learning, example selection and prompt security.

Each of these techniques was described using the format of the Seven Techniques section. These techniques

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

were assessed on the basis of six criteria: (C1) core principle; (C2) type of reasoning; (C3) origin of the knowledge; (C4) number of model invocations per query; (C5) additional requirements like examples and setup; and (C6) tasks and reliability concerns.

We recorded the model, the benchmark, the baseline, the obtained result and the upper bound for each source. All numbers are reproduced from the source and attributed as such. The results of separate papers cannot be compared because of different models, tasks and metrics. Cost evaluation conducted in the Results and Discussion section is done according to how the method is described in its paper, i.e., in terms of the number of model calls necessary to implement it. Hence, it cannot be treated as a benchmark.

Accuracy and validity of the research depend completely on the accuracy of the selected sources since the models and prompting practices develop rapidly and what works for one model does not have to work for another. This topic is discussed in more details in the Limitations section.

Results and Discussion

Table 2 below contains a brief summary of the findings reported in the literature. They are taken from various experiments and models, hence the rows in the table do not represent any order.

Table 2: Summary of Key Findings Reported in the Literature

Method	Reference	Finding Reported
Zero-shot	Brown et al. [2]	GPT-3 family architecture; performs better for larger model sizes; typically below few-shot
Few-shot	Brown et al. [2]	GPT-3 175B comparable to fine-tuning for various benchmarks
Few-shot	Min et al. [16]	Example label space and input format affect performance; correct labels do not matter as much as expected
CoT	Wei et al. [9]	PaLM 540B with eight examples: state-of-the-art on GSM8K
Zero-shot CoT	Kojima et al. [17]	InstructGPT accuracy rose from 17.7% to 78.7% (MultiArith) and from 10.4% to 40.7% (GSM8K) with a trigger phrase
ToT	Yao et al. [10]	GPT-4 with ToT solved 74% of Game of 24 tasks; CoT solved 4%

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

Method	Reference	Finding Reported
GoT	Besta et al. [11]	Sorting quality improved by 62% over ToT with cost reduced by over 31%
Self-Refine	Madaan et al. [18]	Better than one-step results on seven tasks; average gain of about 20%
Self-Refine	Huang et al. [19]	Large models struggle to self-correct reasoning without external feedback; results may become worse
RAG	Lewis et al. [12]	State-of-the-art on three open-domain QA benchmarks; more specific, diverse and factual answers than a parametric-only baseline
RAG	Liu et al. [20]	Models use information less when it occurs in the middle of a long context

It is clear from Table 3 that the number of model calls and preparation requirements before the calls are listed for each technique.

Table 3: Structural Cost of Each Technique

Technique	Model calls per query	Extra need
Zero-shot	1	None
Few-shot	1 (with longer input, *k* examples)	Set of examples
CoT	1 (with longer output)	Examples or trigger phrase
ToT	Many (proportional to width × depth × candidates)	Search controller, judging prompts

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

GoT	Many; depends on the graph of operations	Graph controller, scoring prompts
Self-Refine	$1 + 2T$ for T rounds	Feedback and improvement of prompts, stopping rule
RAG	1 generation + 1 retrieval	Documents, embedding index, retriever

CoT and Self-Refine both depend on the power of the model to create useful intermediate steps or feedback [9], [18]. Hence, the method which works well for the huge model will most likely fail for the small one.

The transition from CoT to ToT to GoT allows increasing the field of actions of the model and enables reflection, generalization, and integration of results. However, there is one drawback - the increase in the number of model calls (Table 3). The big success of ToT in solving the Game of 24 problem [10] and the success of GoT in sorting problem [11] is possible due to the special nature of the problems which allows searching and decomposition. However, the authors of ToT claim that such techniques may not be necessary for simpler problems [10].

The inconsistency problem is solved in each method in its own way. RAG uses generated answers which are based on the retrieved text [12], but is dependent on the quality of retrieval and handling of the long prompt [20]. Self-Refine intends to detect errors via self-feedback, however, empirical results demonstrate that self-feedback itself does not improve the reasoning skill [19]. CoT helps detect errors easily; however, it does not prevent them.

It is obvious that all approaches may not be applicable in every case. The following is just one of many possible ways to choose the technique. Start with zero-shot approach for trivial tasks. Use examples for formatting and labeling. Apply CoT to perform step-by-step reasoning. If one chain fails and the cost is acceptable, then apply ToT or GoT. Use Self-Refine in case when revision improves the result. Apply RAG if there is a necessity to utilize the information that is unknown to the model. Since the data are provided by various models and benchmarks, the discussion is about trade-offs, not measurements.

Gaps in Research and Future Works

- Seven techniques are rarely compared in a single paper using the same models, tasks and metrics.
- In papers, accuracy is always provided without corresponding calls, tokens, and delay, although ToT and GoT are quite costly.
- The results are valid for specific models and tasks, while outputs are extremely dependent on the prompt formulation.
- Each task requires a manual annotation of ToT and GoT with judging methods and plots for each.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

- We still do not know the criteria for when the model can adequately judge its own output for using Self-Refine and ToT judging [19].

- Hallucination is reduced but not eliminated; even RAG cannot get rid of retrieval errors and long-context usage issues [20].

- Prompting and document collections may contain sensitive information, therefore privacy-aware prompting requires further investigation [15]. Prompt injection, where inserted text makes the model ignore its original instructions [32], is a related security concern.

Promising directions include benchmarking techniques that report both accuracy and cost; selecting a method based on the task difficulty; cost-sensitive reasoning that performs search only when necessary; analysis of combinations such as retrieval with step-by-step reasoning that are widely used in practice but rarely analyzed; experiments with smaller models; and evaluation with human experts from education and healthcare.

Limitations

The research is conducted based on previous papers and involves no experiments at all. The information comes from previous papers and reflects the variety of different models, benchmarks, and metrics which makes a comparison between them impossible. Also, there is a limit in the amount of literature analyzed and it was collected in an unstructured way, which means that some pieces of literature can be missed. The most of the results come from the models available until 2020-2024 and newer LLMs can perform differently. Finally, the cost analysis is structural and not measured at all.

Conclusion

In this paper, zero-shot prompting, few-shot prompting, chain-of-thought, tree of thoughts, graph of thoughts, self-refine, and retrieval-augmented generation have been compared based on one format and one criterion. In all the studies, it is evident that each of these methods is designed to address a specific problem – the amount of information available in the prompt, reasoning structure, the need for refining of the generated text, and the source of the information respectively. The more structured the method is, the more times the model is called, and most of the methods depend heavily on the base model. However, RAG is distinguished by its retrieval and the source of information. The main problems of this area are the lack of unified benchmarks, inaccurate costs calculation, and poor generalization and reliability. In order to solve this problem, researchers should compare the methods directly.

References

[1] A. Vaswani et al., “Attention is all you need,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.
[2] T. B. Brown et al., “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

[3] L. Ouyang et al., "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 27730–27744.
[4] H. Touvron et al., "LLaMA: Open and efficient foundation language models," arXiv:2302.13971, 2023.
[5] Z. Ji et al., "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, Art. no. 248, 2023.
[6] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, Art. no. 195, 2023.
[7] P. Sahoo et al., "A systematic survey of prompt engineering in large language models: Techniques and applications," arXiv:2402.07927, 2024.
[8] S. Schulhoff et al., "The Prompt Report: A systematic survey of prompting techniques," arXiv:2406.06608, 2024.
[9] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 24824–24837.
[10] S. Yao et al., "Tree of Thoughts: Deliberate problem solving with large language models," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.
[11] M. Besta et al., "Graph of Thoughts: Solving elaborate problems with large language models," in *Proc. AAAI Conf. Artificial Intelligence*, vol. 38, no. 16, 2024, pp. 17682–17690.
[12] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474.
[13] V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," in *Proc. EMNLP*, 2020, pp. 6769–6781.
[14] Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," arXiv:2312.10997, 2023.
[15] K. Edemacu and X. Wu, "Privacy preserving prompt engineering: A survey," *ACM Computing Surveys*, vol. 57, no. 10, 2025.
[16] S. Min et al., "Rethinking the role of demonstrations: What makes in-context learning work?" in *Proc. EMNLP*, 2022, pp. 11048–11064.
[17] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 22199–22213.
[18] A. Madaan et al., "Self-Refine: Iterative refinement with self-feedback," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.
[19] J. Huang et al., "Large language models cannot self-correct reasoning yet," in *Proc. ICLR*, 2024.
[20] N. F. Liu et al., "Lost in the middle: How language models use long contexts," *Trans. Assoc. Computational Linguistics*, vol. 12, pp. 157–173, 2024.
[21] Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate before use: Improving few-shot performance of language models," in *Proc. Int. Conf. Machine Learning (ICML)*, 2021, arXiv:2102.09690.
[22] X. Wang et al., "Self-consistency improves chain of thought reasoning in language models," in *Proc. ICLR*, 2023, arXiv:2203.11171.
[23] Z. Zhang, A. Zhang, M. Li, and A. Smola, "Automatic chain of thought prompting in large language models," in *Proc. ICLR*, 2023, arXiv:2210.03493.
[24] N. Shinn et al., "Reflexion: Language agents with verbal reinforcement learning," in *Advances in*

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

Neural Information Processing Systems, vol. 36, 2023, arXiv:2303.11366.

[25] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in *Proc. EACL*, 2021, arXiv:2007.01282.
[26] S. Yao et al., "ReAct: Synergizing reasoning and acting in language models," in *Proc. ICLR*, 2023, arXiv:2210.03629.
[27] Y. Zhou et al., "Large language models are human-level prompt engineers," in *Proc. ICLR*, 2023, arXiv:2211.01910.
[28] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proc. EMNLP*, 2021, pp. 3045–3059.
[29] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proc. ACL-IJCNLP*, 2021, pp. 4582–4597.
[30] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What makes good in-context examples for GPT-3?" in *Proc. DeeLIO Workshop, ACL*, 2022, pp. 100–114.
[31] Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp, "Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity," in *Proc. ACL*, 2022, pp. 8086–8098.
[32] F. Perez and I. Ribeiro, "Ignore previous prompt: Attack techniques for language models," *arXiv*:2211.09527, 2022.
[33] J. Wei et al., "Finetuned language models are zero-shot learners," in *Proc. ICLR*, 2022, arXiv:2109.01652.
[34] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "AutoPrompt: Eliciting knowledge from language models with automatically generated prompts," in *Proc. EMNLP*, 2020, pp. 4222–4235.