UIT-NTTT at VLSP2025: A Prompt Engineering Approach for Date Arithmetic Reasoning in Vietnamese

Khoa Nguyen-Anh Le[1,2], Dang Van Thin[1,2],
[1]University of Information Technology-VNUHCM, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
23520742@gm.uit.edu.vn, thindv@uit.edu.vn

Abstract

This paper presents our system developed for VLSP2025: Vietnamese Date Arithmetic (date-arith). The primary goal of this task is to parse and manipulate temporal expressions to compute new dates. This study presents a novel approach to date arithmetic computation using Large Language Models (LLMs) enhanced with prompt engineering techniques. Our methodology employs carefully designed prompts to enable LLMs to autonomously identify key temporal features within input data and execute accurate date calculations. To enhance computational precision, we integrate semantic search capabilities that provide contextual information to support more reliable results. This prompt-driven methodology offers a scalable solution for temporal reasoning tasks without requiring extensive fine-tuning, making it particularly suitable for multilingual and domain-specific applications where traditional rule-based systems may prove insufficient. We achieve a very high accuracy of 0.98 on the public test set and 0.99 on the private test set.

1 Introduction

Date arithmetic represents a fundamental challenge in natural language processing, requiring systems to accurately parse temporal expressions and perform mathematical operations on dates and time intervals. This capability is essential for applications ranging from historical research and legal document analysis to automated scheduling systems and temporal question-answering platforms. Our approach leverages large language models through strategic prompt engineering (Chen et al., 2025), enabling the system to autonomously identify temporal features within input data and execute the necessary calculations without relying on external computational tools or predefined algorithmic frameworks (Xiong et al., 2024). To enhance accuracy, we integrated semantic search mechanisms that provide relevant contextual information to guide the model's reasoning process during date computation tasks. Through our participation in this task, we discovered that well-crafted prompts combined with contextual augmentation can significantly improve the model's ability to handle complex temporal reasoning scenarios, particularly when dealing with historical dates and multi-component time intervals that require careful parsing and sequential calculation steps.

Question	Thời gian 1 năm và 2 tháng trước tháng 6, 1297 là khi nào?
Answer	Tháng 4, 1296
Question	Giả sử bạn đang ở tháng 4, 1316, thời gian sau 4 năm 12 tháng, thì là thời điểm nào?
Answer	Tháng 4, 1321
Question	Hãy tính thời điểm 10 năm trước tháng 2, 1088.
Answer	Tháng 2, 1078

Table 1: Example data point from the training dataset.

2 Background

2.1 Task

The date arithmetic task involves processing questions that contain explicit temporal references and mathematical operations on dates. Each input consists of a natural language question specifying a base date and a temporal operation, such as adding or subtracting years, months, or days from the given reference point. The system must parse these temporal expressions, execute the required arithmetic operations, and produce the resulting date as output. The input questions typically follow patterns that combine a temporal reference point with directional operations. For instance, questions may ask for dates that occur before or after a specified time period relative to a given base date. The expected output format consists of precise temporal specifications, commonly expressed in month-year format for this particular dataset. Example inputs and outputs are provided in Table 1, demonstrating the range of temporal calculations required for

■: base date ■: operation ■: interval date
Question 1: Thời gian 1 năm và 2 tháng trước tháng 6, 1297 là khi nào?
Question 2: Giả sử bạn đang ở tháng 4, 1316, thời gian sau 4 năm 12 tháng, thì là thời điểm nào?
Question 3: Hãy tính thời điểm 10 năm trước tháng 2, 1088

Figure 1: Structure of Question with Three Core Components: Base Date, Operation, and Interval Date.

this task. Model performance is assessed using exact match accuracy, where predictions must correspond precisely to the ground-truth answers. This strict evaluation criterion is appropriate for date arithmetic tasks due to their deterministic nature. Unlike text generation or classification tasks, where metrics such as BLEU, precision, or recall capture nuanced performance, date calculations operate on binary correctness—a date is either accurate or incorrect. For example, predicting "Tháng 5, 1296" when the correct answer is "Tháng 4, 1296" represents complete failure regardless of numerical proximity. Alternative metrics, such as semantic similarity or edit distance would be inappropriate, as they could incorrectly reward near-miss predictions that provide no practical value in real-world applications. This binary evaluation ensures the absolute precision required for domains such as historical research, legal document analysis, and automated scheduling systems.

2.2 Dataset

The evaluation utilizes a Vietnamese-language dataset structured across three standard partitions. The training set contains 3,000 samples, the development set includes 500 samples for validation, and the test set comprises 1,500 samples for final performance evaluation. The dataset focuses specifically on historical dates, with examples spanning medieval time periods, which add complexity due to the extended temporal ranges involved in the calculations.

3 System Overview

In this section, we describe the system in detail. We first generate synthesis data based on the training set. Then, apply semantic search on this new data to increase the information for the LLMs to make predictions. The entire workflow is illustrated in Figure 2.

Question	Thời gian 1 năm và 2 tháng trước tháng 6, 1297 là khi nào?
Reason	Extracted components: Base date = June 1297, Operation = Subtract (-), Interval = 1 year and 2 months. Calculation: Start from June 1297. First, subtract 1 year, which results in June 1296. Then, subtract 2 months from June 1296. Subtracting 1 month from June is May, and subtracting another month is April. Therefore, the final result is April 1296.
Answer	Tháng 4, 1296

Table 2: Example synthesis data have been generated.

3.1 Prompt Engineering

The dataset comprises three primary features that capture temporal characteristics and operational context (described in detail in Figure 1). The **base date** represents the reference time point for each observation. The **operation** feature categorizes the type of activity or process being examined, it can be either "before" or "after". The **interval date** establishes the duration measurement relative to the base date, enabling the temporal analysis of the operational processes. These are the key elements for prompt design.

The prompt architecture follows a structured multi-component design that integrates established prompt engineering techniques for temporal reasoning tasks. The framework begins with expert prompting (Xu et al., 2025) through explicit role definition, positioning the AI as a "temporal calculation expert specializing in precise date arithmetic operations," which leverages role-based prompting to activate domain-specific knowledge and reasoning patterns. The task definition component provides clear operational boundaries by specifying the core function of interpreting and executing date arithmetic calculations. The instructional framework employs a chain-of-thought approach (Wei et al., 2023) through a three-step sequential process: feature extraction (identifying base date, operation type, and time interval), chronological computation (applying temporal

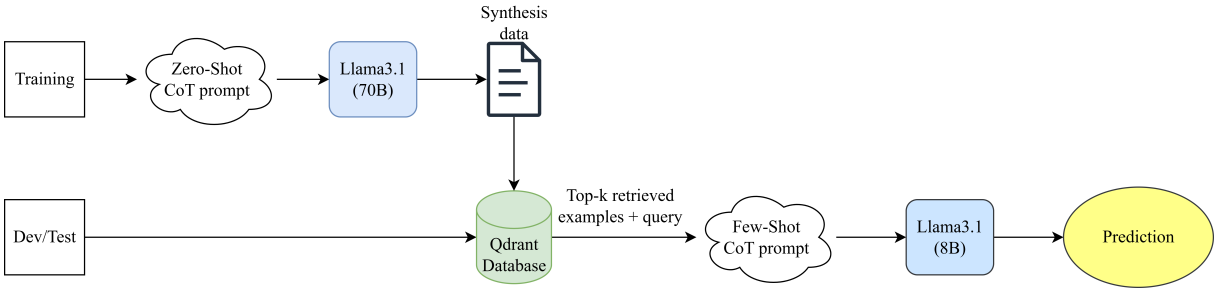

Figure 2: Overview of system pipeline.

logic with calendar rule considerations), and structured response generation. The prompt incorporates comprehensive processing guidelines that address edge cases such as leap years and month boundaries, ensuring robust handling of temporal complexities. Finally, the output specification mandates a standardized JSON format containing both detailed reasoning processes and final calculated results, facilitating a consistent response structure and enabling systematic evaluation of the model's temporal reasoning capabilities across multiple test cases. The entire prompt can be viewed in the section A.

3.2 Synthesis Data

For the synthesis data generation phase, training questions were processed through the **Llama3.1-70B-Instruct** (Vavekanand and Sam, 2024) model using zero-shot chain-of-thought prompting (given in Figure 3) to generate predictions on the training set. We only retain correctly predicted samples to create synthesis data enriched with reasoning explanations (as shown in Table 2). These validated samples were subsequently embedded using the Vietnamese-specific embedding model **dangvantuan/vietnamese-embedding** [1] and stored in a Qdrant [2] vector database for efficient retrieval during the few-shot prompting phase of the inference pipeline.

3.3 Few-Shot Inference

For each sample in the development and test sets, we apply semantic search (Monir et al., 2024) to select the **k** most similar examples based on cosine distance from the Qdrant database. These retrieved examples are then incorporated into a few-shot chain-of-thought prompt (as shown in Figure 4) and processed through the **Llama3.1-8B-Instruct** model (Vavekanand and Sam, 2024) to generate the final prediction. This retrieval-augmented approach leverages the synthetic training data to enhance the model's reasoning capabilities on unseen examples through contextually relevant demonstrations.

3.4 Rule-based model

While the competition guidelines prohibited rule-based systems for official submissions, we developed a simple rule-based baseline for comparative analysis. The implementation employs regular expression pattern matching to extract temporal components from Vietnamese questions, including the base date (month and year), directional indicators ("sau" for addition, otherwise subtraction), and interval magnitudes (years and months). The calculation logic applies straightforward arithmetic operations on the extracted components, handling month-year boundary conditions through iterative adjustments when month values exceed twelve or fall below one. The complete algorithmic specification is presented in Algorithm 1. This baseline serves to demonstrate the relative effectiveness of our prompt engineering approach compared to traditional deterministic methods, particularly in handling the linguistic complexity and temporal reasoning requirements inherent in Vietnamese date arithmetic tasks.

4 Experimental setup

To ensure consistent and structured outputs from the large language models, we designed a system prompt (as illustrated in Figure 5) that enforces JSON response formatting. This prompt requires the model to provide detailed reasoning and final answers in a predefined structure, enhancing output reliability and facilitating automated evaluation. For the semantic search component, we selected $k = 3$ based on empirical evaluation (see Section

[1] https://huggingface.co/dangvantuan/vietnamese-embedding

[2] https://qdrant.tech/

System Prompt

You must respond with a valid JSON object that follows this exact structure:
{
"Reason": "Detailed explanation of your calculation process, including extracted features and step-by-step computation",
"Answer": "Final calculated date in the format 'Tháng [month], [year]'"
}
Requirements:
- Include all fields
- Provide clear explanations in the reason field
- Output only valid JSON, no additional text

Figure 5: System prompt setup so that the model always gives JSON structured output.

5.2 for a detailed analysis). Regarding hyperparameter configuration for large language models, we set the temperature to 0 to ensure deterministic outputs and minimize hallucination effects, thereby improving response consistency across multiple runs. Additionally, we configured the random seed to 13 to maintain reproducibility throughout our experimental procedures.

5 Results

System	Accuracy
Ours	0.98
trinhtrantran122	0.98
dg123	0.98
HUET	0.98
Thailevann	0.98
hotuminh	0.98
truong13012004	0.97
Rule-based system	0.91

Table 3: Ranking on the development set based on accuracy.

5.1 Main Results

Our experimental results demonstrate strong performance across both development and private test datasets. On the development set, we observed progressive accuracy improvements: 0.94 with zero-shot prompting, 0.97 with one-shot prompting, and 0.98 with few-shot prompting using three examples, all implemented with chain-of-thought reasoning. As shown in Table 3, our system achieved competitive performance alongside other top-performing teams, with most leading approaches reaching the 0.98 accuracy threshold on the development set. On the private test set, our method achieved exceptional performance with 0.99 accuracy, demonstrating robust generalization capabilities.

To establish the effectiveness of our prompt engineering approach, we compared it against a rule-based baseline employing regular expression pattern matching and deterministic arithmetic operations. The rule-based system achieved 0.91 accuracy on the development set, seven percentage points lower than our LLM-based approach. This performance gap highlights the advantages of leveraging large language models for temporal reasoning tasks, particularly in handling the linguistic complexity and semantic nuances inherent in Vietnamese date arithmetic questions. Our prompt engineering methodology demonstrates superior adaptability to diverse question structures without requiring manual rule specification for each linguistic variation, thereby offering a more scalable and maintainable solution.

5.2 Ablation Study

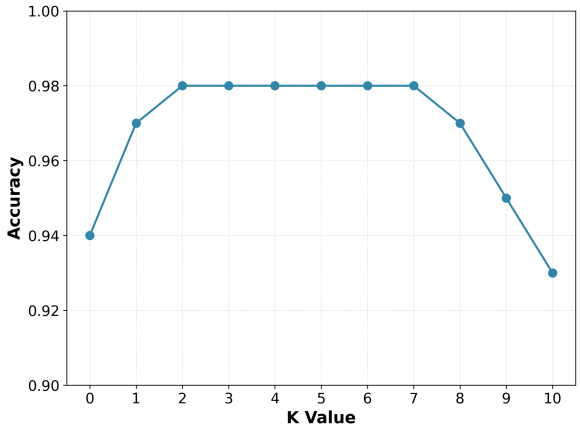

Figure 6: Development set accuracy across different k values.

To determine the optimal number of retrieved examples for few-shot prompting, we conducted an ablation study by varying the parameter k from 0

to 10, as illustrated in Figure 6. The results demonstrate a clear performance trajectory: accuracy increases sharply from 0.94 at $k = 0$ (zero-shot) to 0.97 at $k = 1$, reaching a plateau of 0.98 for k values between 2 and 7. Beyond $k = 7$, performance begins to degrade, dropping to 0.97 at $k = 8$, 0.95 at $k = 9$, and 0.93 at $k = 10$. We selected $k = 3$ as the optimal configuration for our system based on several considerations. First, three examples provide sufficient contextual information to enable accurate temporal reasoning without overwhelming the model with redundant demonstrations. Second, this configuration strikes an effective balance between prediction accuracy and computational efficiency, as limiting the number of retrieved examples reduces prompt length and consequently decreases response latency during inference. Third, the stable performance across k values from 2 to 7 suggests that three examples occupy a robust operating point within the plateau region, offering reliable performance while maintaining processing efficiency. The degradation observed at higher k values likely results from context dilution, where excessive examples introduce noise or exceed the model's effective attention span, thereby compromising reasoning quality rather than enhancing it.

5.3 Error Analysis

Our error analysis reveals important insights about both model performance and dataset quality across different temporal complexity levels. Examining the nine incorrectly predicted samples in the development set (detailed in Table 4), we categorized errors by interval complexity: simple operations (single-unit intervals such as "6 years"), moderate operations (two-unit intervals such as "6 years and 7 months"), and boundary cases (intervals resulting in year transitions). The analysis demonstrates that our model's predictions are mathematically correct for all error cases when independently verified through manual calculations. Notably, the ground truth labels contain systematic inconsistencies, with years differing by several centuries from both our predictions and manual verification results (for example, predicting January 1898 for "6 years and 7 months before August 1904" while the label incorrectly shows May 1406). This pattern persists across all complexity levels, including straightforward single-year calculations and more intricate multi-component intervals, indicating that the errors stem entirely from annotation quality rather than model limitations in handling specific temporal reasoning scenarios. The consistent accuracy of our predictions across varying question difficulties—from simple six-year additions to complex operations involving multiple temporal units—suggests that our prompt engineering approach effectively handles diverse temporal reasoning patterns, with the performance ceiling determined by dataset annotation quality rather than algorithmic capability.

5.4 Discussion

Our results demonstrate that prompt engineering combined with semantic search retrieval offers a viable alternative to traditional rule-based systems for temporal reasoning tasks in Vietnamese. The progressive improvement from zero-shot (0.94) to few-shot prompting (0.98) underscores the value of retrieval-augmented learning, where contextually similar examples enable the model to generalize temporal reasoning patterns without explicit fine-tuning. The seven-percentage-point advantage over the rule-based baseline highlights the superior adaptability of large language models in handling linguistic variations and complex temporal expressions.

Notably, our error analysis reveals that all nine incorrect predictions on the development set were mathematically correct, with discrepancies arising from systematic annotation errors in the ground truth labels rather than model reasoning failures. This finding suggests that the actual performance ceiling of our approach may exceed the reported metrics, limited primarily by dataset quality rather than algorithmic capability. The exceptional performance on the private test set (0.99 accuracy) further validates the robustness and generalization capacity of our methodology across unseen temporal reasoning scenarios.

6 Conclusion

This paper presents a prompt engineering approach for Vietnamese date arithmetic reasoning that achieves exceptional performance without requiring extensive model fine-tuning. Our system leverages strategic prompt design combined with semantic search-enhanced few-shot learning to enable Large Language Models to accurately parse temporal expressions and execute complex date calculations. The methodology demonstrates high performance, achieving 0.98 accuracy on the pub-

lic test set and 0.99 accuracy on the private test set, with progressive improvements from zero-shot (0.94) to few-shot prompting (0.98). The integration of synthesis data generation through chain-of-thought reasoning and retrieval-augmented inference provides a scalable solution for temporal reasoning tasks across multilingual contexts. Future work should explore extending this approach to more complex temporal relationships, incorporating additional temporal units such as days and hours, and evaluating the methodology's effectiveness across diverse languages and cultural calendar systems to establish broader applicability in real-world temporal processing applications.

Acknowledgements

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

References

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2025. Unleashing the potential of prompt engineering for large language models. *Patterns*, 6(6):101260.

Solmaz Seyed Monir, Irene Lau, Shubing Yang, and Dongfang Zhao. 2024. Vectorsearch: Enhancing document retrieval with semantic embeddings and optimized search.

Raja Vavekanand and Kira Sam. 2024. Llama 3.1: An in-depth analysis of the next-generation large language model. Preprint, Datalink Research and Technology Lab. Uploaded to ResearchGate on July 24, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. 2024. Large language models can learn temporal reasoning.

Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. 2025. Expertprompting: Instructing large language models to be distinguished experts.

A Prompt Design

This appendix shows the prompts used for the system, including the zero-shot prompt (Figure 3) to create synthesis data and the few-shot prompt (Figure 4) to make predictions.

Zero-Shot with Chain-of-Thought prompt

You are a temporal calculation expert specializing in precise date arithmetic operations. Your expertise includes parsing temporal expressions, identifying mathematical operations on dates, and performing accurate chronological calculations.

Perform date arithmetic calculations by interpreting questions that involve adding or subtracting time intervals from given dates. Calculate the resulting date based on the specified temporal operations and present your reasoning process clearly.

Following these instruction step by step:
1. Extract the following key components from the input question:
- Base date: Identify the reference date/time mentioned in the question
- Operation: Determine whether to add (+) or subtract (-) the time interval
- Interval: Extract the specific time period to be added or subtracted (years, months)
2. Using the extracted features, perform the chronological calculation:
- Apply the specified operation to the base date
- Account for calendar rules and month/year boundaries
- Ensure the resulting date follows proper temporal logic
3. Present your response in the following JSON structure:
{
"Reason": "Detailed explanation of your calculation process, including extracted features and step-by-step computation",
"Answer": "Final calculated date in the format 'Tháng [month], [year]'"
}

** Processing Guidelines **
- Parse temporal expressions carefully to avoid misinterpretation
- Handle edge cases such as leap years and month boundaries appropriately
- Maintain consistency in date format presentation
- Provide clear reasoning that demonstrates your calculation methodology

Input: #Question: ...
Output:

Figure 3: Zero-Shot with Chain-of-Thought prompt to generate synthesis data.

Few-Shot with Chain-of-Thought prompt

You are a temporal calculation expert specializing in precise date arithmetic operations. Your expertise includes parsing temporal expressions, identifying mathematical operations on dates, and performing accurate chronological calculations.

Perform date arithmetic calculations by interpreting questions that involve adding or subtracting time intervals from given dates. Calculate the resulting date based on the specified temporal operations and present your reasoning process clearly.

Here are some examples:
Example 1: #Question:{...} #Reason:{...} #Answer:{...}
Example 2: #Question:{...} #Reason:{...} #Answer:{...}
Example 3: #Question:{...} #Reason:{...} #Answer:{...}

Following these instruction step by step:
1. Extract the following key components from the input question:
- Base date: Identify the reference date/time mentioned in the question
- Operation: Determine whether to add (+) or subtract (-) the time interval
- Interval: Extract the specific time period to be added or subtracted (years, months)
2. Using the extracted features, perform the chronological calculation:
- Apply the specified operation to the base date
- Account for calendar rules and month/year boundaries
- Ensure the resulting date follows proper temporal logic
3. Present your response in the following JSON structure:
{
"Reason": "Detailed explanation of your calculation process, including extracted features and step-by-step computation",
"Answer": "Final calculated date in the format 'Tháng [month], [year]'"
}

** Processing Guidelines **
- Parse temporal expressions carefully to avoid misinterpretation
- Handle edge cases such as leap years and month boundaries appropriately
- Maintain consistency in date format presentation
- Provide clear reasoning that demonstrates your calculation methodology

Input: #Question: {...}
Output:

Figure 4: Few-Shot with Chain-of-Thought prompt to make final predictions.

B Pseudocode for rule-based systems

Algorithm 1 Parse Question and Calculate Time

function PARSEQUESTION($question$)
  Extract month and year from pattern "tháng (\d+),?\s*(\d+)"
  if no match found **then**
    return NULL
  end if
  $month \leftarrow$ extracted month value
  $year \leftarrow$ extracted year value
  $direction \leftarrow 1$ if "sau" in $question$ else -1
  $years \leftarrow 0$
  Extract years from pattern "(\d+)\s*năm"
  if match found **then**
    $years \leftarrow$ extracted value
  end if
  $months \leftarrow 0$
  Extract all months from pattern "(\d+)\s*tháng"
  for each match **do**
    if $match \leq 12$ **and** $match \neq month$ **then**
      $months \leftarrow match$
      break
    end if
  end for
  return $\{month, year, years\_delta : years \times direction,$
    $months\_delta : months \times direction\}$
end function

function CALCULATETIME($month, year, years\_delta, months\_delta$)
  $year \leftarrow year + years\_delta$
  $month \leftarrow month + months\_delta$
  while $month > 12$ **do**
    $month \leftarrow month - 12$
    $year \leftarrow year + 1$
  end while
  while $month < 1$ **do**
    $month \leftarrow month + 12$
    $year \leftarrow year - 1$
  end while
  return $(month, year)$
end function

C Wrong Predictions

Question	Predict	Label
Hãy tính thời điểm 6 năm và 7 tháng trước tháng 8, 1904	Tháng 1, 1898	Tháng 5, 1406
Giả sử bạn đang ở tháng 11, 1062, thời gian trước 3 năm 4 tháng, thì là thời điểm nào?	Tháng 7, 1059	Tháng 7, 1430
Ngày tháng nào sẽ là 9 năm 2 tháng sau tháng 5, 1868?	Tháng 7, 1877	Tháng 5, 1735
Giả sử bạn đang ở tháng 11, 1718, thời gian sau 6 năm, thì là thời điểm nào?	Tháng 11, 1724	Tháng 6, 1665
Thời gian 5 năm và 5 tháng sau tháng 4, 1733 là khi nào?	Tháng 9, 1738	Tháng 1, 1945
Hãy tính thời điểm 10 năm và 2 tháng sau tháng 11, 1946	Tháng 1, 1957	Tháng 5, 1375
Giả sử bạn đang ở tháng 9, 1338, thời gian sau 2 năm 11 tháng, thì là thời điểm nào?	Tháng 8, 1341	Tháng 9, 1013
Hãy tính thời điểm 4 năm và 7 tháng trước tháng 1, 1155	Tháng 6, 1150	Tháng 8, 1026
Giả sử bạn đang ở tháng 12, 1457, thời gian sau 10 năm 12 tháng, thì là thời điểm nào?	Tháng 12, 1468	Tháng 6, 1165

Table 4: Wrong predictions on the development set.