# Chain-of-Factors: A Zero-Shot Prompting Methodology Enabling Factor-Centric Reasoning in Large Language Models

Musarrat Hussain
*Department of Computer Science*
*UiT The Arctic University of Norway*
Tromsø, Norway
musarrat.hussain@uit.no

Ubaid Ur Rehman
*Dept. of Computer Science & Engineering*
*Kyung Hee University*
Yongin, South Korea
ubaid.rehman@khu.ac.kr

Tri D.T. Nguyen
*Dept. of Computer Science & Engineering*
*Kyung Hee University*
Yongin, South Korea
tringuyendt@khu.ac.kr

Sungyoung Lee
*Department of Computer Science and Engineering*
*Kyung Hee University*
Yongin, South Korea
sylee@oslab.khu.ac.kr

Seong Tae Kim, Sung-Ho Bae, and Jung Uk Kim
*Department of Computer Science and Engineering*
*Kyung Hee University*
Yongin, South Korea
st.kim@khu.ac.kr, shbae@khu.ac.kr, ju.kimm@khu.ac.kr

***Abstract*—Large language models (LLMs) have significantly improved numerous natural language processing tasks. However, their performance relies heavily on the provided instructions or prompts. Recently, several prompting methodologies have been developed to enhance the reasoning abilities of LLMs. Notably, the Chain-of-Thought (CoT) approach provides examples that help break down tasks into sub-steps, resulting in more accurate solutions. However, the process of generating detailed examples may not be user-friendly, as end users prefer providing task descriptions rather than a set of examples. In this study, we introduce Chain-of-Factors (CoF), an innovative zero-shot prompting methodology that incorporates task-specific instructions as a chain of factors into the prompt, aimed at enhancing the factor-centric reasoning abilities of LLMs. Experiments on three LLMs, including ChatGPT-3.5, Gemini, and GPT-4, show performance improvements ranging from 0.01% to 40.2% in accuracy on various symbolic reasoning and logical reasoning tasks compared with zero-shot and few-shot CoT. In summary, CoF enhances LLMs' reasoning abilities by including task-specific steps and instructions, while also decreasing the necessity for fine-tuning specific to each task.**

***Index Terms*—Factor-Centeric Reasoning, Prompt Engineering, Large Language Models.**

## I. Introduction

Large language models (LLMs) have transformed natural language processing (NLP) by achieving state-of-the-art performance on numerous tasks [1]. The success of these models is attributed to their in-context (few-shot or zero-shot) learning capabilities, which enable them to acquire task-relevant instructions in the prompt and generate suitable responses accordingly [2], [3]. These prompts enable LLMs to understand, perform, and adapt tasks without fine-tuning, generating context-aligned outputs [4], [5].

Recently, researchers have proposed various prompting methodologies that enhance the task understanding and reasoning abilities of LLMs [2], [6]–[11]. Broadly, these prompting methodologies are categorized into two types: n-shot and zero-shot prompting. In n-shot prompting, the prompt is supplemented with n examples that aid LLMs in understanding how to approach solving a specific task. One of the most widely used n-shot prompting methodologies is the `Chain-of-Thought (CoT)` prompting [7]. CoT employs a divide-and-conquer approach, breaking down complex tasks into easily solvable sub-steps. The CoT methodology has achieved remarkable improvements in various reasoning benchmarks, particularly excelling in arithmetic reasoning tasks. However, creating step-by-step labeled examples for some reasoning tasks can be difficult. Additionally, users often prefer giving task instructions instead of detailed examples [12]. Therefore, Kojima et al. [2] introduced a zero-shot version of CoT known as Zero-shot-CoT. In this approach, examples are replaced with a single phrase, `Let's think step by step`, instructing LLMs to break down the task into subtasks. Remarkably, Zero-shot-CoT has achieved comparable results to CoT. Nevertheless, relying solely on a generic phrase may not be sufficient to effectively guide LLMs in managing complex tasks. Instead, using task-specific instructions that outline the necessary steps and decision-making process can enhance their comprehension of the task. Specifically, both CoT [7], Zero-shot-CoT [2], and similar n-shot and zero-shot methodologies [8]–[11] encounter challenges when tackling tasks that require implicit reasoning or intuition.

In this study, we introduce a novel zero-shot prompting methodology known as Chain-of-Factors (CoF). CoF incorporates task-specific reasoning steps and intuitions as factors within the prompt. These factors encompass steps, regulations, directives, and decision-making principles aimed at enhancing the factor-centric reasoning capabilities of LLMs. These fac-

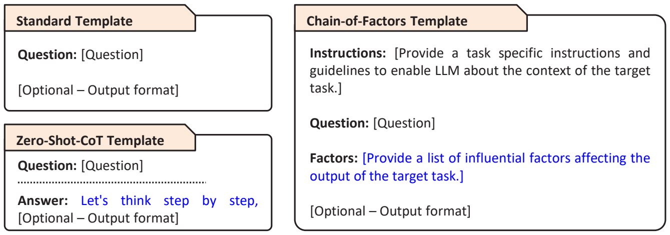


Fig. 1. Templates for prompting in Standard, Zero-shot-CoT, and Chain-of-Factors scenarios.

tors can be manually generated, through machine learning and AI techniques, or a combination of both. The experimental results, conducted with CoF across various datasets (including Last Letter, Coin Flip, Tracking Shuffled Objects, and Semantic Textual Similarity Benchmark), showcase performance enhancements ranging from 0.01% to 75.8% compared to zero-shot and n-shot standard prompts. Additionally, there is an improvement of 0.01% to 40.2% when compared to zero-shot and n-shot CoT prompting methods, all using the `ChatGPT3.5` model. For the `Gemini-Pro` model, performance gains range from 16.7% to 55.8% against standard prompts and from 3.6% to 37.6% against zero-shot and n-shot CoT prompts. Moreover, utilizing the `GPT-4` model yields an improvement of 0.01% to 80.0% against standard prompts and a minimal enhancement of 0.02% to 1.7% against zero-shot and n-shot CoT prompts. These results illustrate that CoF prompting is especially effective for tasks that can be described in terms of factors. Our key contributions are summarized as follows:

- We introduce a novel prompting methodology, incorporating task-specific instructions, rules, constraints, steps, decision making principles or other influential factors empowering factor-centric reasoning abilities of LLMs.
- CoF allows LLMs to acquire new skills by using predefined factors that represent task-specific decision-making logic and abilities. This reduces the necessity for fine-tuning LLMs specifically for each task.
- We demonstrate that CoF is highly appropriate for tasks involving symbolic reasoning or those that can be expressed in terms of factors.

## II. Chain-of-Factors

We present Chain-of-Factors (CoF), a zero-shot template-based prompting technique for factor-centric reasoning. CoF draws inspiration from the concept of a chain-of-thought [7], but generating step-by-step examples can be challenging and may not be user-friendly, as end users are more likely to prefer providing task descriptions rather than a set of examples. Additionally, language models may encounter difficulties in comprehending the reasoning logic from the provided example. Therefore, this research investigates the impact of directly providing instructions as factors across various tasks. Currently, we have manually crafted factors specific to each task to assess the efficacy of LLMs in factor-centric reasoning. Below are the steps we followed to design task-specific factors:

- Randomly selected instances of the given task.
- Manually evaluated each instance, and task description to gain a comprehensive understanding of the task.
- Outlining generic steps, based on our human intuition and approach to task-solving, as factors for tackling the task.

Figure 1 compares the CoF prompt template with standard and Zero-shot-CoT templates. The CoF template primarily comprises three components: instructions, questions, and factors, along with an optional description of the desired output format. The instruction section provides task-related contextual information, the question defines the task, and the factors elaborate on the approach to the task, which can involve either sequential or non-sequential steps. As mentioned earlier, factors can be designed manually, using AI and LLMs, or a combination of both. In this study, we chose manual design to show the CoF's ability to input human decision logic into LLMs for various tasks. A concrete example of the CoF prompting compare to standard, and Zero-shot-CoT is shown in Figure 2 for **a)** Last Letter **b)** Coin Flip **c)** Shuffled Objects and **d)** Semantic Textual Similarity (STS) tasks.

## III. Experimental Setup

**Tasks and Datasets:** We evaluated We tested our CoF method on five datasets from symbolic and logical reasoning categories. For symbolic reasoning, we used two datasets: Last Letter Concatenation and Coin Flip [2]. In Last Letter Concatenation, LLMs had to find a four-word phrase, extract the last character of each word, and concatenate them into a new string. In the Coin Flip task, LLMs determined whether a coin landed heads up after a series of four flips or no flips. For logical reasoning, we used the Tracking Shuffled Objects and STS-Benchmark datasets [2], [13]. The Tracking Shuffled Objects task tests LLMs' ability to determine an object's final state after shuffling three objects. The STS-Benchmark task requires LLMs to rate the semantic similarity between two sentences, scoring from 0 to 5 representing completely different and semantically identical, respectively.

All the tasks mentioned are ideal for CoF since they can be easily described by factors, leading to expected strong performance. We also tested CoF on a less suitable task, `Date Understanding`, from Common Sense Reasoning, where LLMs choose the correct date from multiple options [2], [7]. Accuracy was used as the performance metric for all datasets except STS, which was evaluated using the Pearson correlation coefficient.

**Models:** To demonstrate the concept, we used three well-known language models in our experiments: `ChatGPT3.5`, `Gemini-Pro` and `GPT-4`. We interacted with `ChatGPT3.5` and `GPT-4` through API provided by OpenAI[1], and for `Gemini-Pro`, we utilized the Google API[2] to execute prompts and extract results.

[1]https://openai.com/index/introducing-chatgpt-and-whisper-apis/
[2]https://blog.google/technology/ai/gemini-api-developers-cloud/

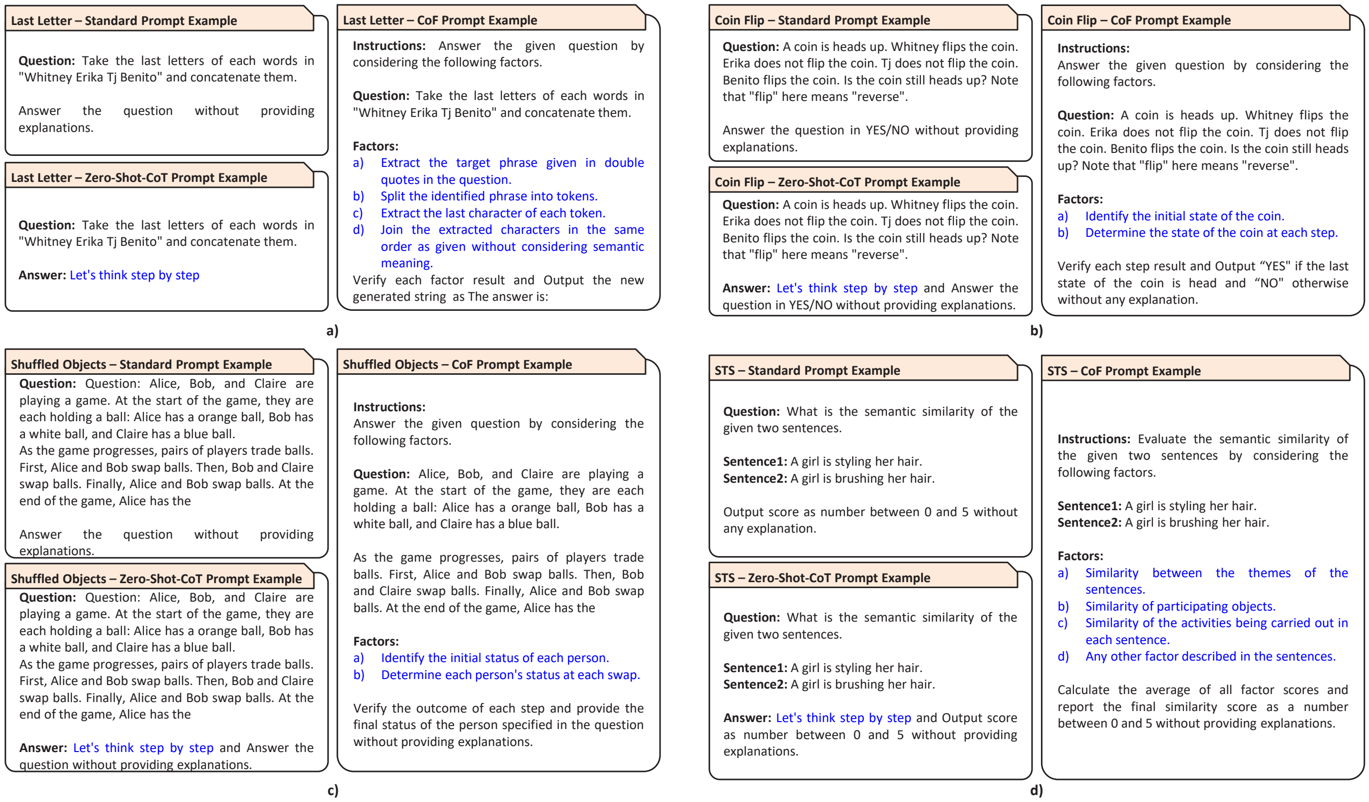


Fig. 2. Example prompts for Standard, Zero-shot-CoT, and Chain-of-Factors scenarios for a) Last Letter b) Coin Flip c) Shuffled Objects d) Semantic Textual Similarity tasks.

**Baselines:** As previously mentioned, the CoF draws significant inspiration from the work presented in CoT [7]. However, it's essential to emphasize that CoF operates within the realm of zero-shot prompting. Consequently, we evaluate CoF in contrast with the vanilla prompt as the Standard prompt, and the zero-shot version of CoT as Zero-shot-CoT as our baseline [2]. Additionally, we also compared CoF with few-shot-CoT supplementing 2, 4, and 8 CoT examples with the prompt. For Zero-shot-CoT, we obtained the results using the methodology outlined in [2]. In their manuscript [2], the Zero-shot-CoT approach uses two prompts for each question: one for reasoning extraction and one for answer extraction. In contrast, we accomplished both reasoning and answer extraction with a single prompt.

## IV. Results and Discussion

Table I displays the results achieved by CoF compared to the Standard, Zero-shot-CoT, and few-shot-CoT prompting approaches. CoF performed significantly better in both symbolic reasoning datasets (`Last Letter` and `Coin Flip`) as well as in both logical reasoning datasets (`Shuffled Objects` and `STS-Benchmark`). However, when it came to the Common Sense Reasoning task (`Date Understanding`), CoF could not outperform its competitors. This outcome was expected because CoF is particularly effective for tasks that can be tackled using factors. Since common sense reasoning tasks are challenging to represent as factors, CoF encounters difficulties in providing appropriate solutions for such tasks.

### A. Last Letter Dataset Results and Analysis

As shown in Table I, in the `Last Letter` task, the `ChatGPT3.5` model, when using the zero-shot standard prompt, predicted only 18 out of 500 instances correctly, resulting in a 3.60% accuracy rate. In the 2-shot, 4-shot, and 8-shot scenarios, the model accurately predicted 16, 10, and 11 instances, yielding accuracies of 3.2%, 2.0%, and 2.2% respectively. When using the zero-shot-CoT prompt, the model accurately predicted 389 instances (77.80%). However, with 2-shot, 4-shot, and 8-shot CoT prompts, the model's accuracy significantly decreased, correctly predicting only 25, 14, and 22 instances respectively, resulting in less than 5% accuracy in all cases. With the CoF prompt, the model correctly predicted 397 instances (79.40%) and placed CoF at the top position. Similarly, the `Gemini-Pro` model failed to predict any instances correctly with the zero-shot, 2-shot, and 4-shot standard prompts and only predicted 2 instances correctly in 8-shot setting. However, the same model was able to achieve 263 (52.60%), 137 (27.4%), 172 (34.4%), and 123 (24.6%) correct predictions with the 0-shot, 2-shot, 4-shot, and 8-shot settings. While, with CoF, the model achieved an accuracy of 56.20% with 281 correct predictions. Furthermore, the `GPT-4` model was able to correctly predict 59, 61, 458, and 43 instances producing 11.8%, 12.2%, 91.6%, and

TABLE I

Accuracy comparison of Standard, Zero-shot-CoT, and CoF on each task. All datasets use accuracy as an evaluation metric except STS-Benchmark, which utilizes Pearson correlation coefficient as an evaluation metric.

| Models | Prompts | Symbolic Reasoning | | Logical Reasoning | | Common Sense Reasoning |
|---|---|---|---|---|---|---|
| | | **Last Letter** | **Coin Flip** | **Shuffled Objects** | **STS-Benchmark (Pearson)** | **Date Understanding** |
| ChatGPT3.5 | 0-shot Standard | 3.6 | 51.2 | 34.0 | 0.76 | 39.8 |
| | 2-shot-Standard | 3.2 | 53.6 | 34.3 | 0.78 | 64.2 |
| | 4-shot-Standard | 2.0 | 51.8 | 30.9 | 0.78 | **67.8** |
| | 8-shot-Standard | 2.2 | 52.0 | 29.2 | 0.78 | 64.2 |
| | 0-shot-CoT | 77.8 | 54.0 | 36.4 | 0.71 | 56.4 |
| | 2-shot-CoT | 5.0 | 48.6 | 26.5 | 0.69 | 50.4 |
| | 4-shot-CoT | 2.8 | 47.8 | 25.5 | 0.78 | 54.5 |
| | 8-shot-CoT | 4.4 | 49.4 | 29.6 | 0.78 | 52.6 |
| | CoF | **79.4** | **94.2** | **71.3** | **0.79** | 38.5 |
| Gemini-Pro | 0-shot Standard | 0.0 | 56.6 | 26.8 | - | 29.0 |
| | 2-shot-Standard | 0.0 | 48.4 | 29.6 | - | 54.5 |
| | 4-shot-Standard | 0.0 | 52.6 | 26.4 | - | 61.2 |
| | 8-shot-Standard | 0.4 | 47.8 | 26.7 | - | **61.8** |
| | 0-shot-CoT | 52.6 | 46.6 | 31.2 | - | 36.6 |
| | 2-shot-CoT | 27.4 | 51.2 | 29.9 | - | 56.9 |
| | 4-shot-CoT | 34.4 | 49.60 | 29.7 | - | 60.7 |
| | 8-shot-CoT | 24.6 | 52.4 | 32.5 | - | 60.2 |
| | CoF | **56.2** | **90.0** | **46.3** | - | 29.3 |
| GPT-4 | 0-shot Standard | 11.8 | 50.2 | 40.4 | 0.83 | 83.5 |
| | 4-shot Standard | 12.2 | 77.0 | 38.1 | - | **87.5** |
| | 0-shot CoT | 91.6 | 57.4 | 41.2 | 0.82 | 83.5 |
| | 4-shot CoT | 8.6 | 97.6 | 39.7 | - | **87.5** |
| | CoF | **92.2** | **97.8** | **42.9** | **0.84** | 82.9 |

8.6% accuracies with 0-shot-standard, 4-shot-standard, 0-shot-CoT, and 4-shot-CoT, respectively. Whereas the same model achieved 92.2% accuracy with CoF by correctly predicting 461 out of 500 instances. The performance of all three models showed improvement in CoF, suggesting that specific factors play a role in guiding the reasoning abilities of LLMs.

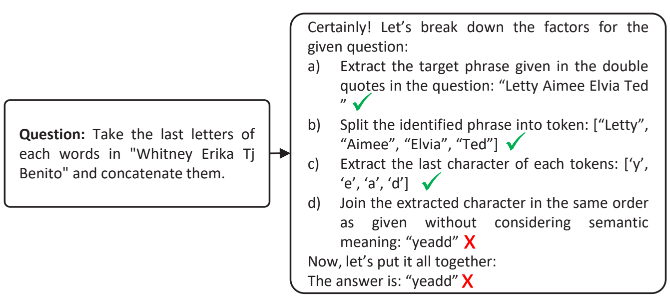


Fig. 3. Character concatenation error example from the last letter dataset in the ChatGPT 3.5 model with a CoF prompt.

The main errors across all three models occurred when combining extracted characters from each word, as shown in Figure 3. The provided question was entered into the prompt (Figure 2, CoF Prompt), and the result was generated by `ChatGPT 3.5`. All Models often added, deleted, or reordered characters to form a meaningful word. Additionally, they occasionally failed to extract the correct number of tokens. In general, CoF performed better compared to the other two counterparts with all three models.

### *B. Coin Flip Dataset Results and Analysis*

In the `Coin Flip` task, CoF performed the best compared to all other tasks, achieving an increase in accuracy of 0.2% to 40.6%. This resulted in an overall accuracy of 94.20% when used with the `ChatGPT 3.5` model. A similar improvement of 33.4% to 37.6% was observed when using the `Gemini-Pro` model with CoF prompting. While an improvement of 0.2% to 20.8% was recorded with `GPT-4` model utilizing CoF Prompt. This significant improvement can be attributed to the factor that guides the LLM to track the state of the coin at each step. As a result, the LLMs can accurately determine the coin state at each flip or no flip, leading to better results.

The manual analysis of errors showed that all three models face two main issues in the `Coin Flip` task (Figure 4). First, as seen in question 1, models often misjudge the coin's state after a flip or no-flip, leading to incorrect predictions. Second, as shown in question 2, models frequently skip steps, resulting in errors. Question 2 illustrates both problems: a missing step and an incorrect coin state.

As highlighted in [2], the Last Letter and Coin Flip datasets are examples of symbolic reasoning tasks. In both of these datasets, the CoF prompting demonstrated effective performance, supporting our assertion that CoF can significantly enhance the performance of LLMs in symbolic reasoning tasks when provided with the correct sequence of factors.

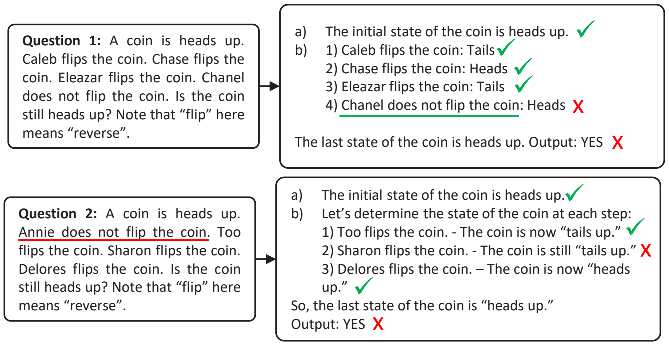


Fig. 4. Missing step and incorrect state error examples from the coin flip dataset in the ChatGPT 3.5 model with a CoF prompt.

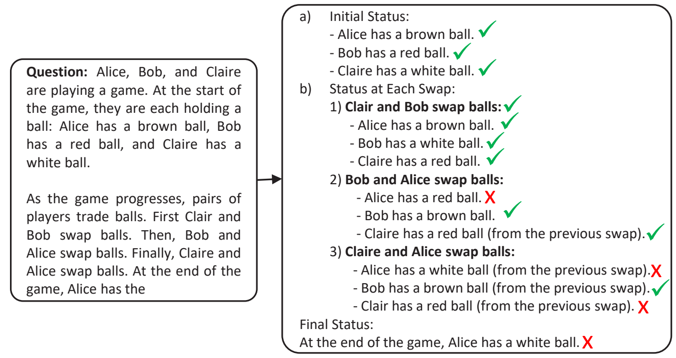


Fig. 5. An example of a shuffled objects tracking error in the ChatGPT 3.5 model with a CoF prompt.

### C. Shuffled Object Dataset Results and Analysis

Continuing from symbolic reasoning, CoF achieved a remarkable improvement of 37.0% and 34.9% compared to standard and CoT, respectively, in the `Shuffled Objects` task with the `ChatGPT3.5` model. Similarly, there was an enhancement of 16.7% and 13.8% observed with the `Gemini-Pro` model compared to standard and CoT, respectively. Furthermore, an advancement of 2.5% and 1.7% was observed with the `GPT-4` model along with CoF compared to standard and CoT prompting. Once again, this significant performance improvement can be attributed to the perfect alignment of the task with the relevant factors. We can easily provide clear and concise instructions to language models regarding object tracking, enabling them to better track objects and thus leading to an overall improvement in performance.

For error analysis, we randomly selected erroneous instances from all three models and manually analyzed the output. The analysis showed that while all models accurately track the initial status and the first swap, they struggle after the first swap. They either fail on the second swap or make errors on the third. Instead of using the latest status, the models update from either the initial status or an unrelated result, leading to incorrect predictions. An example from the `ChatGPT3.5` model is shown in Figure 5. This issue is more prominent in the `Gemini-Pro` and `GPT-4` models, despite CoF performing better overall.

### D. STS-Benchmark Dataset Results and Analysis

The nature of the aforementioned tasks is somewhat similar. Therefore, we assessed CoF using a completely different task: semantic textual similarity (STS). The goal of this task is to have the model evaluate the semantic similarity of two given textual phrases. The STS task is particularly challenging because the semantic similarity of two text snippets can vary with different contexts. Therefore, the model needs to comprehend the context and evaluate the text accordingly. To enhance the understanding of language models regarding the context of semantic text pairs, we provided LLMs with three key factors for STS, in addition to one extra factor to handle variations. These provided factors instruct LLMs to identify and compare themes, actors, and activities performed in a given text, along with the additional factors. This resulted in a slight performance improvement of 0.01 in Pearson correlation scores compared to both standard and CoT prompting, with the `ChatGPT3.5` model. Similarly, The `GPT-4` model showed comparable Pearson correlation improvements of 0.01 and 0.02 compared to standard and CoT prompting methods, respectively. Interestingly, the `Gemini-Pro` model returns an empty string for every instance, indicating that the model is unable to produce a score between 0 and 5.

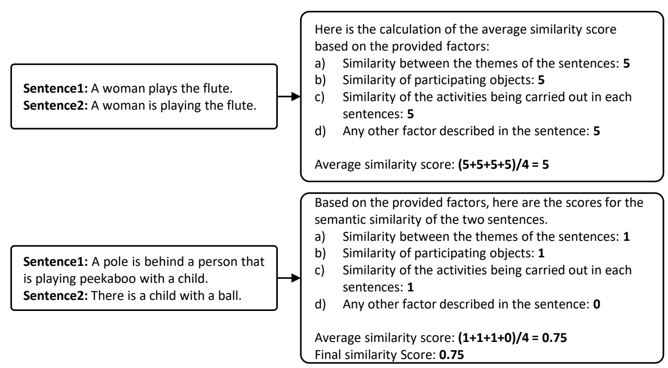


Fig. 6. Examples of ChatGPT 3.5 model generated output for CoF prompt of STS task.

An example of a semantically similar sentence pair, along with a dissimilar pair, is depicted in Figure 6. In both cases, the `ChatGPT3.5` model correctly evaluated the semantic equivalence of the given sentences with CoF Prompting. The model was able to produce the exact same score of 5 as assigned by a human expert. However, in the second sentence pair, the model assigned a score of 0.75, in contrast to the human-assigned score of 0.2. In most cases with standard and CoT prompts, the model compares syntactic similarity, such as common elements, and differences in verb tense and comparisons between the object and subject resulted in a significant difference compared to human-assigned scores. While consistently, CoF utilized the provided factors for each instance. In the presented evaluation, the final score was the average of all individual factors' scores. However, the factors are allowed to be tuned by assigning different weights to each factor, resulted in a more focused and customized evaluation.

### *E. Date Understanding Dataset Results and Analysis*

Now, we should present a task that is not suitable for effective handling by CoF. This task, called `Date Understanding` falls into the category of common sense reasoning. This task is a multiple choice questions and requires LLMs to select a correct date based on a reference date, with the addition or subtraction of days, months, or years. We used the CoF prompt as shown in Figure 7 to query LLMs. The factors presented in Figure 7 were finalized after several improvements. Although these factors were the best ones identified, the performance still did not surpass that of their counterparts. The 4-shot-standard prompt produced the best results of 67.8%, compared to 38.5% accuracy of CoF, recording a lead of 29.3% when using the `ChatGPT3.5` model. Similarly, with the `Gemini-Pro` model. 8-shot-standard prompt achieved the best accuracy of 61.8% compared to 29.3% of the CoF. For the `GPT-4` model, both the 4-shot-standard and 4-shot-CoT approaches achieved an accuracy of 87.5%, whereas the CoF Prompting method achieved 82.9% accuracy.

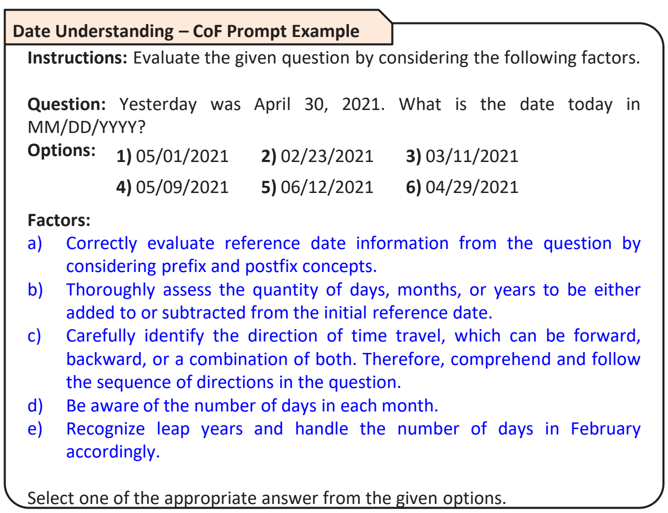


Fig. 7. CoF prompt used for date understanding task.

The error analysis of Date Understanding task has shown that the model primarily makes two types of errors: reference date identification errors and quantity of days, months and years calculation errors. Both of these errors can be traced back to a common root cause, which is the model's tendency to ignore prefix and post-fix phrases during calculations.

The reference date identification is one of the primary errors observed during the task of understanding dates by models. As depicted in Figure 8, the question includes a prefix phrase "Yesterday" along with a reference date. The `ChatGPT3.5` model correctly identifies the reference date phrase, highlighted by the green box in Figure 8. However, the model mistakenly considers the reference date to be "April 30, 2021," disregarding the prefix "Yesterday," even though it is explicitly mentioned in the provided factors to consider both prefix and post-fix concepts. Most of the other factors are assessed correctly despite the incorrectly extracted reference date, leading to an inaccurate final result of "05/01/2021." This issue was consistently observed in all cases, including standard prompts, CoT, and CoF prompting. The majority of errors made by the models were due to the incorrect identification of the reference date, ignoring the prefix and post-fix phrases.

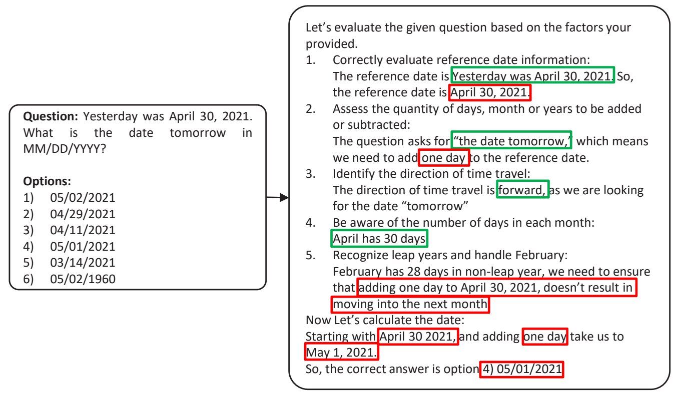


Fig. 8. An example of reference date calculation error from date understanding dataset in the ChatGPT 3.5 model with a CoF prompt.

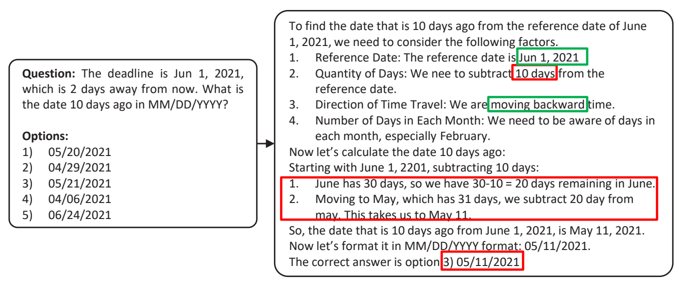


Fig. 9. An example of an error in calculating the number of days from date understanding dataset in the ChatGPT 3.5 model with a CoF prompt.

The calculation error related to days, months, or years quantity refers to inaccuracies in determining the number of days, months, and years to add or subtract from the reference date. This error, as illustrated in Figure 9, often occurs when the model disregards the given prefix and postfix phrases. In the example shown in Figure 9, the model correctly identified the reference date as "Jun 1, 2021." However, when calculating the number of days, it subtracts 10 days from the reference date while ignoring the phrase "which is two days away from now." The model, as indicated by the red box, performs arbitrary calculations, leading to an incorrect final result of 05/11/2021. Interestingly, the model associates the identified date "05/11/2021" with option 3, whereas the actual option is "05/21/2021." Consequently, the model's answer does not align with the provided options.

The results shown in Table I indicates that CoT's performance with few-shot examples lags behind zero-shot CoT, suggesting inappropriate example selection. To address this, we extend our methodology by employing K-Means clustering, inspired by Auto-CoT [8]. By clustering instances and selecting examples from each cluster, we ensured broader task coverage. We also incorporated Self-Consistency evaluation, creating three paths per instance and utilizing majority voting for the final result. In cases of varied LLM responses for the same instance, we adopted the first answer. Results for 2-shot-CoT, 4-shot-CoT, and 8-shot-CoT with Self-Consistency show slight improvement (1-5%), still trailing the proposed CoF method, which is ideal for symbolic and logical reasoning tasks due to its factor-based approach. However, CoF is less effective for tasks involving mathematical calculations, as describing them in terms of factors is challenging, and their complexity may challenge LLMs [14], [15].

Despite significant performance improvements, the proposed CoF prompting has two primary limitations:

The first limitation is related to the importance of identifying the factors that drive the CoF in different tasks. To effectively and efficiently identify these task-specific factors, one needs task-related knowledge, expertise, and manual effort. The quality of the identified factors directly impacts the performance of the CoF, meaning that the same task with different factors may yield different results.

The second limitation pertains to the generation of explanations. Despite explicit instructions not to generate explanations, there are cases where the model still generates output with explanations. Interestingly, this phenomenon is more noticeable when using the CoF prompt compared to the other two prompting methods. This necessitate to post-processing for appropriate response extraction.

## V. Conclusion and Future Work

This research introduced Chain-of-Factors (CoF), a novel approach to enhance zero-shot prompting. CoF embeds task-relevant instructions, steps, regulations, and decision-making principles to enhance factor-centric reasoning. Experimental results demonstrate a significant accuracy improvement of 75.8%, 40.6%, and 37.0% compared to the standard prompt for the Last Letter, Coin Flip, and Shuffled Objects tracking datasets with the `ChatGPT3.5` model, respectively. Additionally, a minor 0.01-point Pearson correlation improvement was observed in a semantic textual similarity task on the STS-Benchmark dataset. When compared to CoT, CoF showed marginal improvements of 1.6%, 40.2%, and 34.9% in the same tasks with the `ChatGPT3.5` model, with a Pearson correlation of 0.79 for CoF compared to 0.78 for the best CoT. Similar improvements were also observed with the Gemini-Pro and GPT-4 models. However, CoF did not perform better in the common sense reasoning task of Date Understanding. These results suggest that CoF has the potential to guide LLMs more effectively in symbolic reasoning, logical reasoning, and other tasks that can be approached with factors.

In the future, we intend to automate the process of extracting factors using LLMs. We will have two or more LLMs collaborate on a specific task to generate a definitive list of agreed-upon factors. This automation will significantly reduce the need for manual intervention, shifting the role of human experts to verifying the identified factors.

## Acknowledgment

This research was supported by the Ministry of Science and ICT (MSIT), Korea, under the Grand Information Technology Research Center (ITRC) support program (IITP-2024-2020-0-01489) and was supported by the MSIT, Korea, under the ITRC support program (RS-2023-00259004) supervised by the Institute for Information & communications Technology Planning & Evaluation (IITP) and by IITP grant funded by the Korea government(MSIT) (IITP-2022-0-00078, Explainable Logical Reasoning for Medical Knowledge Generation), (IITP-2017-0-00655, Lean UX core technology and platform for any digital artifacts UX evaluation).

## References

[1] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–40, 2023.

[2] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[3] J. Wang, E. Shi, S. Yu, Z. Wu, C. Ma, H. Dai, Q. Yang, Y. Kang, J. Wu, H. Hu *et al.*, "Prompt engineering for healthcare: Methodologies and applications," *arXiv preprint arXiv:2304.14670*, 2023.

[4] L. Giray, "Prompt engineering with chatgpt: A guide for academic writers," *Annals of Biomedical Engineering*, pp. 1–5, 2023.

[5] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, and Z. Sui, "A survey for in-context learning," *arXiv preprint arXiv:2301.00234*, 2022.

[6] B. Zhang, B. Haddow, and A. Birch, "Prompting large language model for machine translation: A case study," *arXiv preprint arXiv:2301.07069*, 2023.

[7] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.

[8] Z. Zhang, A. Zhang, M. Li, and A. Smola, "Automatic chain of thought prompting in large language models," *arXiv preprint arXiv:2210.03493*, 2022.

[9] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, O. Bousquet, Q. Le, and E. Chi, "Least-to-most prompting enables complex reasoning in large language models," *arXiv preprint arXiv:2205.10625*, 2022.

[10] T. Khot, H. Trivedi, M. Finlayson, Y. Fu, K. Richardson, P. Clark, and A. Sabharwal, "Decomposed prompting: A modular approach for solving complex tasks," *arXiv preprint arXiv:2210.02406*, 2022.

[11] L. Wang, W. Xu, Y. Lan, Z. Hu, Y. Lan, R. K.-W. Lee, and E.-P. Lim, "Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models," *arXiv preprint arXiv:2305.04091*, 2023.

[12] R. Lou, K. Zhang, and W. Yin, "A comprehensive survey on instruction following," 2024.

[13] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation," *arXiv preprint arXiv:1708.00055*, 2017.

[14] A. Efrat and O. Levy, "The turking test: Can language models understand instructions?" *arXiv preprint arXiv:2010.11982*, 2020.

[15] P.-N. Kung and N. Peng, "Do models really learn to follow instructions? an empirical study of instruction tuning," 2023.