Evaluating Plan-and-Solve Prompting for Text Formalization: A Comparison with One-Shot Prompting

Kalina Dimova, 196032
Faculty of Computer Science and Engineering
Skopje, North Macedonia
kalina.dimova@students.finki.ukim.mk

Abstract—Prompting techniques have become a powerful means of leveraging large language models (LLMs) for a variety of tasks without fine-tuning. While structured prompting strategies such as Plan-and-Solve (PS) and its enhanced variant PS+ have shown promise in reasoning and symbolic tasks, their effectiveness in open-ended language generation remains underexplored. In this work, we investigate the performance of PS and PS+ prompting for the task of text formalization, where informal input is transformed into a more formal linguistic style. We compare these approaches against a one-shot prompting baseline that provides a single demonstration of the desired transformation. Our experiments, conducted using a curated dataset of informal-to-formal text pairs, evaluate outputs on grammaticality, semantic fidelity, and formality. Results indicate that PS+ prompting outperforms both standard PS and one-shot prompting in producing formalized text that is both accurate and stylistically appropriate. These findings suggest that structured reasoning-based prompting can be effectively applied beyond traditional logical domains and offer new pathways for controllable text generation using LLMs.

Index Terms—Large Language Models, Plan-and-Solve Prompting, One-Shot Prompting, Text Formalization, Prompt Engineering, Style Transfer, Natural Language Generation

I. Introduction

Large language models (LLMs) have demonstrated remarkable proficiency across a variety of natural language processing (NLP) tasks, from question answering to text summarization. A key factor behind their success lies in their ability to adapt to downstream tasks through carefully crafted prompting strategies. Among these strategies, prompting techniques that encourage structured reasoning—such as chain-of-thought (CoT) prompting—have proven particularly effective in multi-step problem solving and reasoning-intensive domains.

Recently, Plan-and-Solve (PS) prompting has been introduced as a method that improves upon standard CoT prompting by explicitly dividing tasks into subtasks and systematically solving them. An extension of this, known as PS+ prompting, incorporates additional instructions to guide LLMs in identifying relevant variables and performing accurate intermediate steps. While PS and PS+ have shown strong performance in mathematical and symbolic reasoning, their potential in more naturalistic and open-ended NLP tasks—such as text transformation or formalization—remains underexplored.

In this paper, we explore the effectiveness of PS and PS+ prompting strategies for the task of text formalization, where the goal is to convert informal or conversational input into its formal equivalent. This task serves as a valuable testbed for evaluating the structural reasoning capabilities of LLMs, as it requires the model to infer context, apply syntactic and stylistic transformations, and preserve semantic intent.

We compare the performance of PS and PS+ prompting against a one-shot prompting baseline, which provides a single demonstration of the task. Unlike prior studies that focused on arithmetic or logic-based challenges, our study evaluates how well these prompting methods handle linguistic nuance and stylistic transformation. We conduct experiments using a curated set of informal-formal text pairs, measuring both the grammatical correctness and fidelity of the generated outputs.

Our results reveal that Plan-and-Solve prompting offers a promising alternative to example-based prompting in text formalization tasks. Specifically, PS+ prompting yields outputs that are not only more grammatically correct but also more aligned with the desired formal tone, suggesting that planning-based prompts can effectively guide LLMs even in stylistically driven tasks. These findings contribute to a growing body of work exploring how prompt engineering can be used to unlock latent capabilities in large-scale language models without relying on fine-tuning or extensive example curation.

II. Related work

Prompting strategies have become essential tools for unlocking the capabilities of large language models (LLMs) across a range of tasks. As researchers move away from fine-tuning toward zero- and few-shot prompting techniques, there has been growing interest in methods that can induce complex reasoning or stylistic transformations with minimal supervision. This section reviews relevant work in three main areas: structured prompting for reasoning, stylistic text generation, and one-shot prompting.

A. Text Formalization and Stylistic Generation.

Text formalization involves transforming informal or conversational language into a more formal style while preserving meaning. Traditional approaches to this task have relied heavily on parallel corpora and supervised models, often tailored to specific domains. The review in Text Generation (2021) outlines how more recent neural approaches—such as encoder-decoder architectures and unsupervised style transfer methods—have attempted to generalize across text attributes like formality, politeness, and sentiment. However, many of these methods still require fine-tuning or domain-specific datasets. Prompt-based approaches for text style transformation remain underexplored, particularly in terms of leveraging structured reasoning strategies like PS and PS+.

Jain et al. (2019) propose an unsupervised framework for controllable text formalization that avoids the need for parallel corpora. Their system uses an encoder-decoder architecture guided by external scoring modules that assess fluency, semantic similarity, and readability. Users can control the degree of formalization at inference time, making the model highly adaptable. This work highlights the promise of modular, controllable architectures for style transformation—an approach conceptually similar to PS+ prompting in its two-stage design. However, unlike Jain et al.'s system, our work focuses on prompting rather than training, evaluating whether LLMs can be steered using structured instructions alone.

B. Plan-and-Solve prompting.

Yao et al. (2023) introduced the Plan-and-Solve (PS) prompting framework as a zero-shot alternative to chain-of-thought (CoT) prompting for multi-step reasoning tasks. Rather than prompting models to "think aloud" in a free-form manner, PS prompting explicitly separates the task into two phases: first generating a high-level plan and then solving the problem based on that plan. Their experiments showed that PS outperformed standard zero-shot and CoT prompting on various math and symbolic reasoning benchmarks. An enhanced variant, PS+, incorporated a structured two-stage prompt that encouraged decomposition of the task into relevant variables, further improving performance. While effective for logical and mathematical tasks, these approaches have not been extensively tested in language generation domains such as style transfer or formalization.

C. Chain-of-Thought prompting.

Chain-of-thought prompting (Wei et al., 2022) improves performance on complex reasoning tasks by encouraging LLMs to articulate intermediate reasoning steps. It has proven effective in arithmetic, commonsense, and symbolic domains. However, recent work by Wang et al. (2023) highlights a critical nuance: models often benefit from the structure of the reasoning rather than its correctness. Even flawed reasoning paths can lead to correct answers, suggesting that the presence of intermediate scaffolding—rather than logically valid chains—is what drives performance gains. These insights help contextualize the design of PS and PS+ prompting, which seek to offer more interpretable and controlled reasoning structures.

D. One-shot Prompting and Generalization.

Wang (2024) investigates how LLMs perform on inductive reasoning tasks with only a single example, showing that one-shot prompting can achieve strong in-distribution generalization. The thesis demonstrates that LLMs are capable of learning and applying abstract rules even when only a minimal demonstration is provided. This insight positions one-shot prompting as a valuable baseline for low-resource or minimal-example settings. However, its application to open-ended generation tasks such as formalization has not been fully assessed.

In contrast to prior work, this research examines the intersection of these domains by evaluating how structured prompting methods, specifically PS and PS+, perform on a natural language transformation task. By comparing these methods to one-shot prompting for the task of formalizing text, we explore whether planning-based strategies can support LLMs in producing more stylistically appropriate and semantically faithful outputs, even in the absence of explicit task-specific tuning.

III. Methodology

This study evaluates the effectiveness of three prompting strategies—one-shot prompting, Plan-and-Solve (PS) prompting, and PS+ prompting—for the task of text formalization. The objective is to determine whether structured, multi-step reasoning prompts (PS and PS+) yield higher-quality formal text transformations compared to a single-example prompt (one-shot). The following subsections describe the dataset, prompting design, model setup, and evaluation protocol used.

A. Dataset

We use a benchmark corpus from Grammarly consisting of paired informal and formal English sentences. Each data instance contains an input in colloquial or unstructured form (Style 1) and a target output in professionally structured language (Style 2). Input files are formatted as tab-separated values and are preprocessed using pandas to ensure data consistency and accessibility.

B. Prompting strategies

Each of the three prompting conditions is defined by a different instruction format embedded into the input text. The only variation between conditions lies in the prompt structure itself; model architecture and inference settings are held constant.

1) One-Shot Prompt: This baseline prompt includes a single input-output example demonstrating the transformation of an informal sentence into formal language. The model is then asked to perform the same transformation on a new sentence using the pattern:

One-Shot Prompt

Rewrite these sentences to make them more formal.

Example Input: "yahoo music, you pay like $10 a month for unlimited downloads"

Example Output: "On Yahoo Music, you pay approximately $10 for unlimited downloads."

Now rewrite this sentence: "{informal_text}"

2) Plan-and-Solve Prompt: This variant introduces task instructions and multiple examples, outlining the refinement steps without requiring intermediate reasoning to be made explicit:

Plan-and-Solve Prompt

You are an expert in text refinement.
Your task is to convert informal text into professional, grammatically correct, and well-structured English while preserving meaning.

Follow these steps:
1. Identify informal words, slang, abbreviations, and grammatical errors.
2. Replace informal words with their formal equivalents.
3. Correct any grammar, punctuation, and spelling mistakes.
4. Ensure proper capitalization and sentence structure.
5. Maintain the original intent of the message.

Example Input: "Famil t Guy, The Simpons, Futurama, and South Park!!!!"
Example Output: "The best cartoons are ""The Family Guy"", ""The Simpsons"", ""Futurama"", and ""South Park""."

Example Input: "yahoo music, you pay like $10 a month for unlimited downloads"
Example Output: "On Yahoo Music, you pay approximately $10 for unlimited downloads."

Example Input: "(or w/e) p.s gurl how old r u ?"
Example Output: "How old are you?"

Now process this sentence: "{informal_text}"

3) PS+ Prompt: This enhanced version extends PS by explicitly requiring a planning stage before text rewriting. The prompt includes structured breakdowns of each transformation decision and then synthesizes the final output:

PS+ Prompt

You are an expert in text refinement.
Your task is to convert informal text into professional, grammatically correct, and well-structured English while preserving meaning.

Follow these structured steps:

Step 1: Extract Key Elements
- Identify slang, abbreviations, grammar mistakes, and informal words in the input text.

Step 2: Plan the Formalization
- Determine the best formal equivalents for identified informal elements.
- Ensure grammatical correctness and proper sentence structure.

Step 3: Transform & Verify
- Rewrite the text in a formal and polite manner.
- Double-check capitalization, punctuation, and clarity.
- Ensure that the final sentence is fluent and natural.

Example Conversion:
Input: "(or w/e) p.s gurl how old r u ?"
Plan:
- "w/e" → "whatever" (remove as unnecessary)
- "p.s" → Remove (not needed in direct speech)
- "gurl" → "girl" → Adjust to "you" for clarity
- "how old r u?" → "How old are you?"
Final Output: "How old are you?"

Second Example Conversion:
Input: "Famil t Guy, The Simpons, Futurama, and South Park!!!!"
Plan:
- "Famil t Guy" → correct spelling to "Family Guy"
- "The Simpons" → correct spelling to "The Simpsons"
- Excess punctuation ("!!!!") → reduce for clarity
- Add structure: rephrase as a statement
- Add quotation marks around titles
Final Output: "The best cartoons are 'The Family Guy', 'The Simpsons', 'Futurama', and 'South Park'."

Third Example Conversion:
Input: "yahoo music, you pay like $10 a month for unlimited downloads"
Plan:
- "yahoo music" → Capitalize to "Yahoo Music"
- "like $10" → Approximate language adjusted to "approximately $10"
- Minor cleanup for grammar and tone
Final Output: "On Yahoo Music, you pay approximately $10 for unlimited downloads."
—

Now, transform this text: "{informal_text}"

C. Models

We evaluate all prompting strategies using three open-source LLMs of varying size and architecture:

1) tiiuae/falcon-7b-instruct

- Type: Decoder-only LLM
- Parameters: 7B
- Training: Instruction-tuned on mixed datasets for conversational tasks
- Key Features: Optimized for inference efficiency (FlashAttention, multiquery); strong instruction-following ability
- Use Case: Ideal for instruction-based prompting and dialogue generation

2) google/flan-t5-large

- Type: Encoder-decoder transformer (T5)
- Parameters: 783M
- Training: Fine-tuned on 1000+ tasks including multilingual data
- Key Features: Excels in zero-shot and few-shot prompting; strong generalization across NLP tasks
- Use Case: Suitable for translation, summarization, QA, and other text-to-text tasks

3) facebook/opt-1.3b

- Type: Decoder-only LLM
- Parameters: 1.3B
- Training: Pretrained via causal language modeling
- Key Features: Open weights; suitable for text generation benchmarks; basic instruction following
- Use Case: Baseline model for evaluating prompting techniques and generation tasks

Each model is loaded using the HuggingFace Transformers library and executed in deterministic inference mode on GPU. Prompted inputs are tokenized and processed in batches, with a maximum output length of 60 tokens.

D. Evaluation Metrics

Model outputs are evaluated using multiple standard metrics for text generation:

1) BLEU (sentence-level): Measures n-gram overlap between a candidate sentence and one or more reference sentences.

2) ROGUE-1: Measures overlap of unigrams (single words) between candidate and reference.

3) ROGUE-2: Measures overlap of bigrams (two-word sequences).

4) ROGUE-L: Measures the longest common subsequence (LCS) to capture sentence-level structure.

5) METEOR: Evaluates translation quality based on unigram matches, weighted by precision, recall, and alignment.

6) BERTScore: Measures semantic similarity between candidate and reference using contextual embeddings from pretrained language models (like BERT).

For each test sample, we compute metric scores between the model's formalized output and the human-authored formal reference. Results are aggregated across the test set for each model and prompting strategy.

E. Implementation Details

The experiments are implemented in Python using HuggingFace Transformers, torch, and common NLP evaluation libraries (nltk, rouge, bert-score). A unified script handles data loading, prompt construction, generation, and evaluation. To ensure comparability, all hyperparameters—such as batch size, decoding temperature, and beam width—are held constant across all conditions. The script outputs both prediction-level results and per-split aggregated summaries for further analysis.

IV. Results

We evaluated three prompting strategies—one-shot, Plan-and-Solve (PS), and PS+—on the task of text formalization using Falcon-7b-instruct, Flan-T5-large, and OPT-1.3b. While quantitative results favored one-shot prompting across nearly all metrics, qualitative analysis revealed deeper insights into how these strategies shaped model behavior and why the outputs diverged.

A. Quantitative results

1) One-shot prompting: The one-shot prompt consistently outperformed both PS and PS+ across most

metrics and models. For Flan-T5-large, one-shot achieved the highest overall scores, including BLEU (0.2016), ROUGE-1 (0.4932), ROUGE-2 (0.3022), and BERTScore (0.7191). Similar trends were observed for Falcon-7b-instruct and OPT-1.3b, where one-shot prompting yielded better ROUGE and BERTScore values relative to PS variants.

ONESHOT	tiiuae/falcon-7b-instruct	google/flan-t5-large	facebook/opt-1.3b
BLEU	0.1439	**0.2016**	0.0411
ROUGE-1	0.3993	**0.4932**	0.2826
ROUGE-2	0.2248	**0.3022**	0.1386
ROUGE-L	0.3815	**0.4749**	0.2742
METEOR	0.3886	**0.4672**	0.2928
BERTScore	0.6478	**0.7191**	0.4676

TABLE I
ONESHOT evaluation metrics across models. Best scores are highlighted in bold.

2) Plan-and-Solve prompting: By contrast, PS prompting underperformed in comparison to one-shot across all three models. For example, with Falcon-7b-instruct, PS achieved a BLEU of only 0.0253 compared to 0.1439 for one-shot. This drop in performance extended across ROUGE and METEOR as well. While PS produced outputs that adhered more closely to step-wise instructions, this rigid structure appeared to reduce fluency and semantic fidelity, as reflected in the lower automatic scores.

PS	tiiuae/falcon-7b-instruct	google/flan-t5-large	facebook/opt-1.3b
BLEU	0.0253	**0.1433**	0.0398
ROUGE-1	0.1413	**0.3987**	0.2924
ROUGE-2	0.0647	**0.2363**	0.1489
ROUGE-L	0.135	**0.3831**	0.2823
METEOR	0.1741	**0.4022**	0.283
BERTScore	0.4217	**0.5856**	0.4623

TABLE II
PS evaluation metrics across models. Best scores are highlighted in bold.

3) PS+ prompting: PS+ prompting yielded results slightly better than PS but still below one-shot prompting. With Flan-T5-large, PS+ produced a BERTScore of 0.6401, which was higher than PS (0.5856) but still substantially below one-shot (0.7191). This indicates that the explicit planning stage in PS+ may contribute to marginal improvements over PS, but does not surpass the effectiveness of simpler example-driven prompting.

PS+	tiiuae/falcon-7b-instruct	google/flan-t5-large	facebook/opt-1.3b
BLEU	0.0241	**0.1458**	0.0412
ROUGE-1	0.1629	**0.404**	0.2907
ROUGE-2	0.0702	**0.2397**	0.1434
ROUGE-L	0.157	**0.3883**	0.2823
METEOR	0.1458	**0.4056**	0.2712
BERTScore	0.4093	**0.6401**	0.4904

TABLE III
PS+ evaluation metrics across models. Best scores are highlighted in bold.

B. Qualitative results

1) One-shot Prompting: One-shot prompting provided a direct exemplar, allowing the models to imitate the stylistic transformation rather than explicitly reason about it. For instance, when given the input "and on the other hand you know ur husband well", the outputs closely matched the desired formalization: "And on the other hand, you know your husband well."

The strength of one-shot prompting is that the models used the provided example as an anchor, applying surface-level edits such as fixing capitalization, spelling, or contractions. This preserved semantic fidelity while producing fluent outputs.

However, some outputs carried over informal traces (e.g., retaining redundant punctuation or filler words) because the transformation was guided by analogy rather than rules.

The models here are relying on pattern matching. By seeing a single example of informal → formal, they infer the "style" to imitate and extend it to new cases. This aligns well with LLMs' training on parallel text examples (formal/informal pairs in web data).

2) Plan-and-Solve Prompting: PS prompting decomposed the task into explicit steps (identify slang, replace informal words, correct grammar). While this worked well in reasoning-heavy domains in prior work, our experiments showed frequent misalignment in formalization tasks. For example, the input "That's what I want in a man!" was transformed into "I am looking for a man who is mature, responsible, and reliable."—a semantically plausible but invented rewrite.

This prompting technique occasionally introduced richer paraphrasing and a more polished phrasing, but frequently over-edited, drifting from the original meaning. Models sometimes misinterpreted the step-wise instructions, generating verbose or off-task content.

Because PS prompts emphasize reasoning through steps, models sometimes treat the instructions as a license to generate explanations rather than outputs. This encourages hallucination: instead of simply rewriting, the model "rationalizes" what a formal version could mean, even when unnecessary.

3) PS+ Prompting: PS+ added an explicit planning stage (extract, plan, transform). While this yielded slightly more systematic handling of slang and spelling, outputs were often cluttered with reasoning traces. In some cases, the model echoed parts of the planning steps directly in the output, rather than producing a clean sentence. For example, a "plan" list with substitutions (e.g., "ur → your") sometimes leaked into the final response.

This technique was better at catching obvious surface-level issues like "ur" → "your" or "alot" → "a lot.", yet added verbosity and repetition; occasionally failed to exit reasoning mode, producing hybrid outputs that combined analysis with text.

This reflects the model's difficulty in separating instruction following from task execution. When told to "plan first, then transform," models sometimes interleave the plan with the output. This behavior arises from training: LLMs are optimized to continue text sequences, so they

often treat the plan as part of the target completion rather than an intermediate step.

4) Cross-Model Analysis: Flan-T5-large consistently produced the most fluent and accurate outputs. Its encoder-decoder structure, which is designed for conditional sequence transformation, aligns naturally with formalization tasks.

Falcon-7b-instruct often generated verbose or incomplete outputs under PS and PS+, suggesting it struggled more with separating reasoning scaffolding from final answers.

OPT-1.3b frequently defaulted to near-verbatim repetition of the input text, particularly under PS and PS+, showing limited capacity to apply transformations without direct exemplar guidance.

V. Conclusion

The results highlight a key difference between reasoning-oriented tasks and stylistic transformation tasks like text formalization. While PS and PS+ prompting have shown strong performance in reasoning-heavy domains (e.g., mathematics, logic puzzles), their structured decomposition appears less effective for stylistic rewriting. In this setting, one-shot prompting provides the model with a clear stylistic exemplar, which better guides the transformation into formal text.

One possible explanation is that text formalization relies heavily on implicit stylistic cues—such as tone, register, and lexical choice—that are more effectively demonstrated through direct examples than through abstract planning instructions. Structured prompts like PS and PS+ may overemphasize step-wise reasoning, leading to outputs that are grammatically correct but less natural, as reflected in lower BLEU, ROUGE, and BERTScore values.

Across models, Flan-T5-large consistently achieved the strongest results, indicating that encoder-decoder architectures may be particularly well-suited to formalization tasks. Falcon-7b-instruct and OPT-1.3b showed weaker performance overall, though they followed the same general trend of one-shot outperforming PS and PS+.

In summary, our experiments suggest that example-based prompting remains superior for text formalization, while reasoning-oriented approaches like PS and PS+ may not generalize effectively to stylistic generation tasks. This underscores the importance of tailoring prompting strategies to task requirements rather than assuming generalizability across domains.

References

[1] Wang L., Xu W., Lan Y., Hu Z., & Lan Y., Ka-Wei Lee R. & Lim E. (2023). Plan-and-solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models.

[2] Wang B., Min S., Deng X., Shen J., Wu Y., Zettlemoyer L. & Sun H. (2023). Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters.

[3] Wang, J. (2024). LLMs one-shot learning from human demonstration on inductive reasoning tasks (Master's thesis, Rice University).

[4] Jain P., Mishra A., Prakash Azad A. & Sankaranarayanan K. (2019). Unsupervised Controllable Text Formalization. In The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

[5] Becker, J., Wahle, J. P., Gipp, B., & Ruas, T. (2024). Text Generation: A Systematic Literature Review of Tasks, Evaluation, and Challenges. University of Göttingen, Germany.

[6] Li Y., Hui B., Xia X., Yang J., Tang M., Zhang L., Si Sh., Chen L., Liu J., Liu T., Huang F. & Li Y. (2024). One-Shot Learning as Instruction Data Prospector for Large Language Models.

[7] Wang X., Li C., Wang Z., Bai F., Luo H., Zhang J., Jojic N., Xing E. & Hu Zh. (2023). PromptAgent: Strategic Planning with Language Models Enables Expert-Level Prompt Optimization.

[8] Raman S. S., Cohen V., Rosen E., Idrees I., Paulius D. & Tellex S. (2022). Planning with Large Language Models via Corrective Re-prompting.