Prompt Engineering 101
Prompt Engineering Guidelines from a Linguistic Perspective

Wenjuan Han[1]*, **Xiang Wei**[1]*, **Xingyu Cui**[1]*, **Ning Cheng**[1]*,
Guangyuan Jiang[2], **Weinan Qian**[2] **Chi Zhang**
[1] Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing, China
[2] Peking University, Beijing, China
wjhan@bjtu.edu.cn, 22120436@bjtu.edu.cn
22120357@bjtu.edu.cn, ningcheng@bjtu.edu.cn
jgy@stu.pku.edu.cn, ypqwn@stu.pku.edu.cn, chi.zhang@ucla.edu

Abstract

Deploying tuning-free prompting is challenging in engineering practice: it not only requires users to engage in cumbersome trials and errors but is also extremely time-consuming, as even a slight change in wording and phrasing could have a huge impact on the final performance. To further investigate the impact of different prompts, in this work, we perform a systematic inspection of four factors in linguistics involved in prompt engineering: syntax, semantics, lexicon, and pragmatics. The empirical results quantify the sensitivity of the output to small textual perturbations in four linguistic factors of prompts. Based on the analysis of these four factors, we present a series of design guidelines to help human users write effective prompts. Human evaluation on amateurs shows that using the proposed guidelines helps humans produce prompts with significant gains in zero-shot performance in Pre-trained Language Models (PLMs) and hence validates the utility of the guidelines.

1 Introduction

Dramatic gains in Large Language Models (LLMs) have been witnessed in recent years (Brown et al., 2020; Rae et al., 2021; Thoppilan et al., 2022; Chowdhery et al., 2022), accelerated by the discovery of prompting (Liu et al., 2021; Wei et al., 2022; Zhang et al., 2021). Prompts function in the form of natural language to drive LLMs to access a variety (Wei et al., 2023; Wei et al., 2024) of downstream tasks (See Sec. A). However, amateurs must engage in cumbersome trials and errors when the input prompt quality is poor or ambiguous for LLMs to deliver their real intention (Jiang et al., 2020; Lu et al., 2022; Kim et al., 2022; Madaan and Yazdanbakhsh, 2022; Khashabi et al., 2022). Guiding amateurs to clearly present their intention, we refer to education and linguistic theory where linguistics knowledge often be utilized to guide second-language students to effectively express ideas in writing. There is a vast body of research available in the field of linguistics that delves into these topics (Givón, 1978; Karakoç and Köse, 2017).

Inspired by this, we investigate ***linguistics*** factors of prompts. A general overview of how these linguistic factors can be employed to support humans is as follows. **Syntax** studies how words and morphemes combine to form larger phrases and gradually constitute a sentence (Lyons, 1981; Silva and Matsuda, 2012). Central concerns of syntax include word order, verb tense, subject-verb agreement, hierarchical sentence structure, and other syntactic features. Understanding syntax can aid second language students in composing coherent sentences (Ferris, 1999). **Semantics** studies the meaning in natural languages. Negation, denoting the reversal of some meaning groups in a sentence, is one of the most discussed phenomena in semantics (Givón, 1978). Human beings can learn about negation and can benefit from developing their understanding of the appropriate use of words and expressions in specific contexts. **Lexicon** or lexicology, studies the vocabulary of a language. (Johnson et al., 2016) and (Karakoç and Köse, 2017) examine

* Equal Contribution

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 1408-1426, Taiyuan, China, July 25 - 28, 2024.
Volume 1: Main Conference Papers

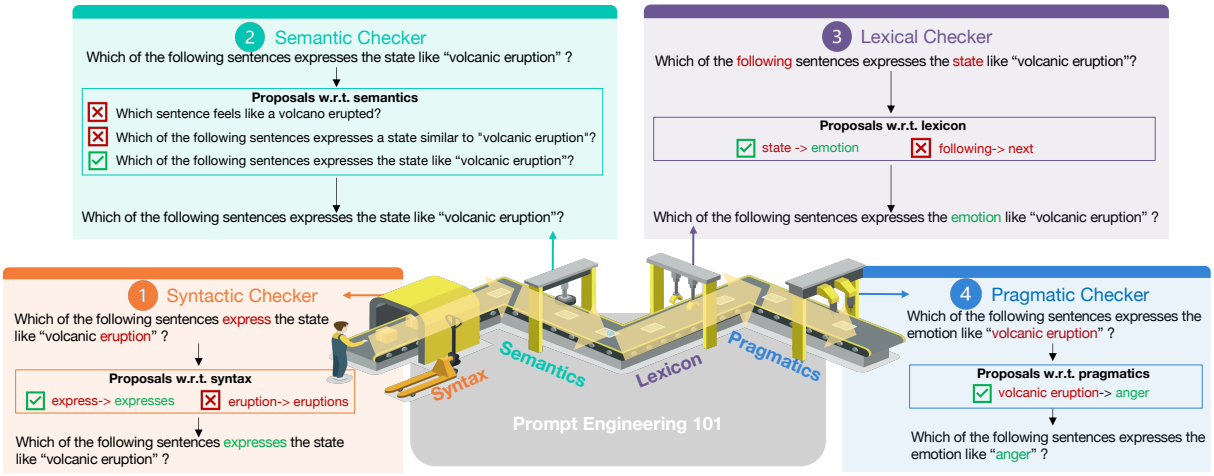

Figure 1: Illustration of our guidelines `Prompt Engineering 101`. The guidelines involve four stages: a syntactic checker to correct grammatical errors, a semantic checker to make proposals without negation, a lexical checker to substitute works, and a pragmatic checker to remove metaphors. `Prompt Engineering 101` is used for an amateur to interact with PLMs via prompting. Following this workflow, the amateur checks carefully each factor and finally writes a prompt. For better checking, in each stage, some suggestions (bullet points in the box) from external automatic software (*e.g.*, grammatical error corrector) will be provided for your reference. The amateur can choose to accept the suggestion or not.

the vocabulary knowledge and suggest effective vocabulary strategies for improving writing proficiency. Similarly, (Razeghi et al., 2022) consistently shows that LLMs are more accurate on samples with words more commonly used. **Pragmatics** is the study of using languages to express what we call "Beyond Saying" in linguistics and cognitive science (Korta and Perry, 2020). Generally speaking, semantics focuses on the literal meanings of linguistic expressions, whereas pragmatics captures the context-dependent implicit factors of meaning (Fried et al., 2022). The study of pragmatics involves metaphors, deceits, indirect speech, irony, Gricean maxims, humor, and coherence inferences (Hu et al., 2022). (Kasper and Roever, 2005) offers insights into how pragmatic knowledge helps produce appropriate sentences. Against this background, we explore the aforementioned four crucial factors: syntax, semantics, lexicon, and pragmatics, since these four factors provide valuable insights that can help humans produce coherent sentences and present desired intentions.

We systematically and quantitatively inspect the impact of the four linguistics factors on prompting LLMs (Sec. 2). For each factor, we study the impact on zero-shot performance via controlled experiments. For syntax, we randomly permute words and introduce grammatical errors in prompts in the experimental group (See Sec. 2.1). For semantics, we explore whether prompts with perturbed semantics but correct syntax can still produce meaningful outputs (See Sec. 2.2). For lexicon, we study prompts with different wording and phrasing. In particular, we replace words with synonyms or translate them into other languages, *e.g.*, Chinese (See Sec. 2.3). Lastly, we investigate the pragmatics of metaphors. We build QA tasks with and without metaphors and study their impact on performance (See Sec. 2.4). The results from these experiments show clear paths for us to navigate in the space of prompt design and steer LLMs toward desirable generations. We, therefore, present a series of design guidelines `Prompt Engineering 101` (Sec. 3) to help humans and the automated toolkit refine prompts. Evaluation on both humans (Sec. 4.2) and automated toolkit (Sec. 4.3) show that using the proposed guidelines subjects produce prompts with significant gains in zero-shot performance in LLMs and hence validates the utility of the guidelines.

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 1408-1426, Taiyuan, China, July 25 - 28, 2024.
Volume 1: Main Conference Papers

2 Analysis of impact of Linguistic Components

This section describes our analysis of the four linguistic components and the proposed guidelines to operationalize them. We list the guidelines by analyzing the impact (namely, sensitivity) of each specific component on the LLMs: if a slight change in terms of a component could make a huge difference in LLMs's performance, then this component is significant as an guideline term. For example, using metaphors for emphasis instead of straightforward statements intended to better describe a task might lead to devastating performance cliff. Hence the guideline should take metaphors into account. Note that our analysis is centered around the zero-shot setting as the few-shot setting is already well-studied by prior arts (Brown et al., 2020; Lu et al., 2022) and the exemplars are not our focus. In this experiment and others in the following, we use *text-davinci-002* of GPT-3 in the zero-shot setting and set the temperature as zero (See Sec. B for more hyper-parameters). For all tasks and datasets, we follow the default setting of (Kojima et al., 2022) and run the experiments on test sets.

2.1 Impact of Syntax

We study the impact of syntax on the zero-shot prompting performance by comparing results from prompts of natural English and ones where word order is perturbed and grammatical errors are introduced.

2.1.1 Grammatical Error Injection

Setup We produce perturbed prompts by artificially injecting grammatical errors into the question. Specifically, we study the problem on the StrategyQA task (Geva et al., 2021). The original question, as well as the options, if any, are used as the prompt. We additionally append "Therefore, the answer is:" to the prompt to induce question answering behaviors in LLMs. Without further notice, we use the same way to create original prompts for other factors. The perturbed prompt is generated by adding rule-based noises (Bryant et al., 2022) to the question and pairing it with the original options: see Tab. 1 for a complete list of the rule-based noises we use.

Error Type	Error Description
Omi_Prep	Omission of the preposition.
Omi_Conj	Omission of the conjunction.
Omi_And	Using two consecutive adverbs of question or pronouns without "and".
Double_Noun	Double nouns.
Repeat_Word	Repeat a word.
Plu_Aft_Card	Using plural noun after cardinal numbers.
Super_Bef_Prep	Using a superlative adjective before preposition "than".

Table 1: The list of the rule-based errors injected in the question.

Results The results in Tab. 2 show the accuracy of original prompts *vs*. perturbed prompts. We observe that grammatical errors do not necessarily have a negative impact on the zero-shot prompting. The robustness of GPT-3 against minor grammatical errors is similar to that of humans. For humans, minor grammatical errors that do not change semantics will not affect understanding either.

2.1.2 Word Permutation

Setup While small perturbations like grammatical errors will not affect the performance, we further study severe perturbation to syntax: eliminating the syntactic structure by randomly shuffling words in a sentence (Kramsch, 2014). We experiment on two tasks: MultiArith (Roy and Roth, 2016) and StrategyQA (Geva et al., 2021). In the perturbed prompts, words are randomly reordered.

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 1408-1426, Taiyuan, China, July 25 - 28, 2024.
Volume 1: Main Conference Papers

Error Type	Original	Perturbed
Omi_Prep	0.570	0.577
Omi_Conj	0.505	0.473
Omi_And	0.580	0.590
Double_Noun	0.537	0.553
Repeat_Word	0.540	0.530
Plu_Aft_Card	0.533	0.500
Super_Bef_Prep	0.286	0.429

Table 2: Comparison of accuracy on StrategyQA from original prompts and perturbed prompts on grammatical errors from Tab. 1.

Results The results are shown in Tab. 3. We observe that shuffling the words brings a notable decrease in zero-shot performance. It clearly illustrates that word order in the prompts affects the performance of LLMs. In summary, the experiments show that prompts should still follow the common word order while minor grammatical errors are negligible.

	Original	Perturbed
MultiArith	0.162	0.035
StrategyQA	0.593	0.473

Table 3: Comparison of accuracy on MultiArith and StrategyQA from original prompts and perturbed prompts on word order permutation.

Few-shot Setting Additionally, we study the effect of syntax on the few-shot setting, feeding GPT-3 with eight examples and the query. The word order of the questions in examples is shuffled for the perturbed prompts. See Tab. 18 for the results.

	Origin	Perturbed
MultiArith	0.327	0.305
StrategyQA	0.614	0.600

Table 4: Comparison of accuracy of original prompts *vs*. perturbed prompts with perturbations to syntax (Few-Shot Setting).

2.2 Impact of Semantics

To what extent can LLMs understand a sentence when a negated word is injected to reverse the semantic meaning? We find the answer based on the comparison between prompts with and without negation.

Setup We experiment on three types of tasks and nine datasets constructed for experiments on semantic negation following (Jang et al., 2022): commonsense reasoning tasks on PIQA (Bisk et al., 2020), ARC-Easy (Clark et al., 2018), and COPA (Gordon et al., 2012), sentence completion tasks on HellaSwag (Zellers et al., 2019), StoryCloze (Mostafazadeh et al., 2016), and Lambada (Paperno et al., 2016), and question answering tasks on WQ (Berant et al., 2013), NQ (Kwiatkowski et al., 2019), and TriviaQA (Joshi et al., 2017). All sample sizes are 300 except for the COPA dataset, which has a sample size of 100. For each dataset, we compare the original prompts with the perturbed prompts. For the perturbed prompt, we change the question to be expressed using negative phrases. As there are two questions, the originally correct choice will be wrong while the originally wrong one will turn correct.

Results From the results shown in Tab. 5, we find that negation brings continued challenges for LLMs on various tasks, and yet human beings handle it very well (Jang et al., 2022). The results suggest that when writing a prompt, users should avoid using negative words.It's better to be clear without ambiguity, and show obvious and concrete goals.

	Original	Perturbed
ARC-Easy	0.960	0.907
PIQA	0.840	0.783
COPA	0.970	0.520
HellaSwag	0.663	0.473
StoryCloze	0.933	0.630
Lambada	0.593	0.440
WQ	0.933	0.803
TriviaQA	0.963	0.973
NQ	0.910	0.880

Table 5: Comparison of accuracy on various datasets from original prompts and perturbed prompts for semantics.

2.3 Impact of Lexicon

Here we focus on which word to choose when multiple words with closely related meanings exist. We approach the problem by comparing the original prompts with perturbed prompts where certain words are replaced with their synonyms or translations, *e.g.*, Chinese, while keeping the semantic meaning.

2.3.1 Synonym Substitution

Setup Experiments are conducted on two tasks: HellaSwag (Zellers et al., 2019) and COPA (Gordon et al., 2012). For the question of a perturbed prompt, we randomly pick up a notional word and replace it with its synonym based on WordNet (Bird et al., 2009). Specifically, the question is tagged using a part-of-speech tagger (Bird et al., 2009) to find out a notional word such as a noun or verb. Then the notional word is substituted with its synonym from WordNet (Bird et al., 2009). We only change one notional word per prompt. If no words in a prompt have synonyms in WordNet, we will keep the prompt as is. To make sure the semantics of the prompt is preserved after the word substitution, BERTScore (F1-score) (Zhang et al., 2019) between the original prompt and the perturbed prompt is calculated to filter out those with low scores. Options remain unchanged.

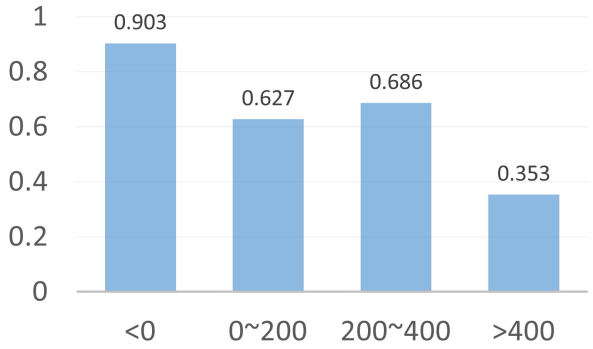

Figure 2: Relationship between the familiarity degree and performance decrease. The experiment is conducted on *HellaSwag*. The x-axis refers to the familiarity score of the original prompt minus the familiarity score of the perturbed prompt. The larger this difference is, the greater the unfamiliarity. The y-axis represents the performance of the perturbed prompt.

Results The experimental results are shown in Tab. 6. Compared with the original prompts, the perturbed prompts lead to inferior performance. Furthermore, we investigate the reasons behind the degradation. We mask the replaced word and calculate the conditional probability of the word masked to measure familiarity and denote it's log value as the familiarity score. As shown in Fig. 2, performance is positively correlated with familiarity. The more familiar the prompts, the better the model performs.

	Original	**Perturbed**
HellaSwag	0.60	0.55
COPA	0.97	0.95

Table 6: Comparison of accuracy on HellaSwag and COPA from original prompts and perturbed prompts on synonym substitution.

2.3.2 Translation Substitution

Setup We further experiment with word substitution with another language. We evaluate translation substitution on a randomly selected set of 300 samples from StrategyQA training dataset. Similar to the word substitution, the perturbed question substitutes a word with its translation in another language. We study English to Chinese translation in this work.

Results The results are shown in Tab. 7. In general, the two substitution methods achieve relatively the same performance compared with the original prompting method. However, translation substitution has a bigger impact than synonym substitution. This is predictable as the perplexity score of the perturbed question with a synonym is higher than the perturbed question with a translated word in another language. It's suggested that users should choose words that are more statistically prevalent in the training corpus of LLMs.

	Original	**Perturbed**
Synonym	0.540	0.541
Translation	0.540	0.530

Table 7: Comparison of accuracy of original prompts *vs*. perturbed prompts with changes on lexicon.

Zero-Shot CoT We also conduct the experiment to study lexicon's effect on the zero-shot CoT via translation substitution. The results are shown in Tab. 19.

	Original	**Perturbed**
Zero-Shot	0.540	0.530
Zero-Shot CoT	0.513	0.497

Table 8: Comparison of accruacy of original prompts *vs*. perturbed prompts with perturbations to the lexicon.

2.4 Impact of Pragmatics

We primarily test metaphor understanding and compare original prompts with and without metaphors.

Setup For experiments on pragmatics, we design a new task based on the Emotion Recognition dataset on SemEval2018 (Mohammad et al., 2018) and TweetEval (Barbieri et al., 2020). We name the task Prompt-based Metaphor Emotion. The task is a multi-choice classification task and aims to recognize four basic emotional expressions: anger, joy, sadness, and optimism. A model is expected to pick one according to the question. We design two types of questions, with and without metaphors. The questions including original prompts and eight perturbed sets are

shown in Tab. 9. The entire dataset contains 1421 samples, covering 558, 358, 382, and 123 samples for anger, joy, sadness, and optimism, respectively.

Prompt Template for Analysis of Pragmatics
Please choose the appropriate answer for the following question from A, B, C, D. Which of the following sentences expresses the emotion like "*¡Emotion Description¿*"? A: You know, she is a disabled figure. But why did you still laugh at her? Do you want to push her buttons? B: The film tells the story of Miguel, a little boy from the shoemaker's family. His family has a mysterious ban —-— No one in the family can touch music performances. C: When the adorable kid met her favorite idol, she was suddenly weeping with joy. D: On the Day of the Dead in Mexico, the dead return to the real world and gather with their loved ones. Leave me alone, ok? I'm in a bad mood now.

¡Emotion Description¿

ID	Emotion of Anger	Emotion of Joy	Emotion of Sadness	Emotion of Optimism
Original	Anger	Joy	Sadness	Optimism
Perturbed_1	Flew off the handle	Jumping up and down with excitement	Down in the dumps	Every cloud has a silver lining
Perturbed_2	Lose your cool	Grinning from ear to ear	Feel blue	When one door closes, another one opens
Perturbed_3	Blow a fuse	In seventh heaven	Heart is broken	There is light at the end of the tunnel
Perturbed_4	Go ballistic	Walking on air	Reduced to tears	When life hands you lemons, make lemonade
Perturbed_5	Hit the ceiling	On cloud nine	Cry one's eyes out	Count your blessings
Perturbed_6	Volcanic eruption	Over the moon	Lump in one's throat	Hope against hope
Perturbed_7	Flip one's wig	On top of the world	Under the weather	There are plenty of fish in the sea
Perturbed_8	Raise the roof	A pig in mud	Get the face of a wet weekend	See the world through rose-colored glasses

Table 9: The eight perturbed sets of prompts used to express anger, joy, sadness and optimism, respectively. The same column indicates the same emotion.

Results The results are shown in Tab. 10. The perturbed prompts generally underperform the straightforward ones in each emotion. Therefore, we recommend not using metaphors, but stating ones' needs directly. If metaphors are necessary, it is recommended to choose those that are very common and widely used, such as *Every cloud has a silver lining*.

3 `Prompt Engineering 101`

Since Sec. 2 show clear paths for us to navigate in the space of prompt design, in this section, we discuss good practices for an amateur to interact with LLMs via prompting, following the analytical results aforementioned. In addition to general requirements, *e.g.*, the language choice, human readability, avoiding abstract statement (Khashabi et al., 2022), prompt engineering should follow four main guidelines as `Prompt Engineering 101`:

ID	Anger	Joy	Sadness	Optimism
Original	0.772	0.688	0.696	0.538
Perturbed_1	0.668	0.576	0.626	0.544
Perturbed_2	0.626	0.562	0.608	0.532
Perturbed_3	0.602	0.516	0.582	0.516
Perturbed_4	0.578	0.504	0.582	0.490
Perturbed_5	0.554	0.502	0.522	0.478
Perturbed_6	0.444	0.496	0.484	0.422
Perturbed_7	0.376	0.474	0.458	0.392
Perturbed_8	0.194	0.330	0.432	0.388

Table 10: Comparison of accuracy of original prompts and perturbed prompts on metaphoric pragmatics.

- Checking Syntax: Sentences should be free from grammatical errors and respect proper word order. If possible, a grammatical error checker will be helpful for users to check their language.
- Checking Semantics: Sentences should be clear without ambiguity. Users should describe their goals and intentions directly without using many negations.
- Checking Lexicon: Without changing the meaning, a user should choose words that are prevalent for training LLMs.
- Checking Pragmatics: We recommend not using metaphors, but stating the needs straightforwardly. If metaphors are necessary, it is recommended to choose those that are very common and widely used.

Fig. 1 illustrates the flow diagram of our proposed guidelines. It is a pipeline including four stages. Our prescribed approach first requires grammatical error correction to check the input sentence. With grammar errors free, the prompt will further go through analysis on semantics, removing unnecessary negations, common word substitution, and metaphor removal.

Guidelines Automation We further automated `Prompt Engineering 101` to build an **automatic toolkit** designed based on our guidelines. The specific automation implementation program is as follows. First, we performed spelling corrector [0] and grammar checker [1] on the original prompt. Then, we used ChatGPT [2] for synonym replacement following negation detection [3]. If negations exist, a rephrasing tool was used to rephrase the sentence. Finally, we performed metaphor detection [4]. Prompts designed for ChatGPT is shown in Tab. 11. We used the OpenAI official interface for the experiments, modeled as `gpt-3.5-turbo` [5].

4 Validation of `Prompt Engineering 101`

4.1 Setup

We validate the proposed guidelines with 7 selected tasks from BIG-bench Lite (BBL) (Srivastava et al., 2022): Auto Debugging, Code Line Description, Formal Fallacies Syllogisms Negation, Hindu Knowledge, Operators, Play Dialog Same or Different and Language Identification. Each of these tasks tests a specific aspect of the capabilities of language models. For each dataset, we choose at most 100 instances to conduct our experiment. If the size of the dataset is less than 100, we adopt the whole dataset for our experiment. The hyper-parameters are identical to the analysis experiments in Sec. 2. The experiments are conducted with ten amateurs over these seven tasks. We recruit human subjects from Prolific. More specifically, we create a survey

[0] https://github.com/Rudransh11Kohli/Grammarly
[1] https://github.com/bhattbhavesh91/gramformer-tutorial
[2] https://openai.com/blog/chatgpt
[3] https://github.com/evanmiltenburg/negation-detector
[4] https://huggingface.co/lwachowiak/Metaphor-Detection-XLMR
[5] Following paper (Kojima et al., 2022), temperature=0 and the other parameters are defaulted.

Phase	Prompt
Synonym	`Please replace the sentence "<sentence>" with synonyms so that the semantics of the sentence remain unchanged. Only replace words, do not change the structure and length of sentences, and do not replace if there are no alternatives. Please output the modified sentence, do not output redundant content.`
Negation	`The following sentence "<sentence>" contains negation "<negation>", please rewrite the sentence so that it does not contain negation and keeps the semantics. Note that negation refers to a phrase or word with a negative meaning. Please output the modified sentence, do not output redundant content.`
Metaphor	`The following sentence "<sentence>" contains metaphor, please rewrite the sentence so that it does not contain metaphor and keeps the semantics. Please output the modified sentence, do not output redundant content.`

Table 11: Prompts of ChatGPT.

that allows workers to understand what the task is, given labeled instances as well as the task title. We first provide workers with comprehensive instructions and a set of unlabeled instances to judge if a worker understands the task. During the survey, we show workers an interface for writing down their descriptions of a given task as the prompts. After the initial prompts[6] are written, we recruit another ten workers to refine the initial prompts with and without our guidelines (namely, guidelines for human or automatic toolkit). We compare the performance of the amateurs' initial prompts and the new set of prompts refined with/without the help of our guidelines on the randomly sampled tasks and test instances.[7]

4.2 Results With *vs*. Without Guidelines for Humans

As evidenced by (Mishra et al., 2022; Efrat and Levy, 2020), original prompts (**Row** `initial prompts`) written by amateurs are not easy to follow for LLMs. The same observation can also be found in our experiments. All results are shown in Tab. 12. After humans refine the prompts without the help of our guidelines (**Row** `w/o human`), the performance increase from 54.73% to 57.04%. The guidelines for humans further increase it to 57.46% (**Row** `w human`).

4.3 Results With *vs*. Without Guideline for Automatic Toolkit

As shown in Tab. 12, the average performance improvement achieved by applying our automated toolkits (**Row** `w automatic`) is 1.31% (from 54.73% to 56.04%). More importantly, we further combine guidelines for humans and automatic toolkit to refine the prompts and got a improvement of 3.37% (**Row** `w human + automatic`).

	Perf.	△
Initial prompts	54.73	-
w automatic	56.04	+1.31
w/o human	57.04	+2.31
w human	57.46	+2.73
w human + automatic	**58.10**	+3.37

Table 12: Results with/without our guidelines for humans and automatic toolkit. Perf.: Performance.

4.4 Ablation Study

We conduct an ablation study for Sec. 4.2 to reveal the functions of each checker w.r.t each linguistic factor. As shown in Tab. 13, all checkers play an important role in improving writing

[6]There are 70 initial prompts and about 7000 instances in total.
[7]Performance in this section refers to accuracy.

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 1408-1426, Taiyuan, China, July 25 - 28, 2024.
Volume 1: Main Conference Papers

prompts. They complement each other and contribute to the final improvement. Most errors occur in the stage of lexicon checking, indicating that many prompt writers are unaware of using more common words for LLMs. Our guidelines provide useful advice in this regard. Sometimes, writers are careless with their spelling, word order, and the use of ambiguous words and negation, which should be eliminated according to our guidelines.

	#Mod.	Perf.
`Prompt Engineering 101`	59	57.46
-Pragmatic Checker	58	57.36
-Lexical Checker	24	54.77
-Semantic Checker	15	54.76
-Syntactic Checker	0	54.73

Table 13: #Mod.: The number of modifications for each checker. Perf.: Task performance.

4.5 Significance Analysis

We use the accuracy of the 10 experiments (each set contains initial prompts and refined prompts). The null hypothesis (H0) is that the guidelines are beneficial to the performance. The significance value is 0.05. We conducted the two-sided t-test. The calculated t-value is -2.30, and the corresponding p-value is 0.034, lower than the selected significance value of 0.05. Therefore, the null hypothesis holds, which means the guidelines are helpful for improving the performance of GPT-3.

4.6 Model Generalization Validation

To verify whether our method has model generalization, we perform negation experiments on four different GPT-3 variants of different sizes (`text-ada-001`, `text-babbage-001`, `text-curie-001` and `text-davinci-003`). The results are shown in Tab. 20. We found that the experimental group with negations generally performed worse than the control group. This conclusion is consistent with the experiments on `text-davinci-002`, so we consider our approach to be model generalizable.

	ARC-Easy	PIQA	COPA	HellaSwag	StoryCloze	Lambada	WQ	TriviaQA	NQ
text-ada-001(1.3B)	0.427/**0.467**	**0.533**/0.407	0.400/**0.440**	0.380/**0.400**	0.443/**0.497**	0.247/**0.323**	0.393/**0.453**	0.410/**0.473**	0.403/**0.557**
text-babbage-001(6.7B)	0.463/**0.527**	0.473/**0.523**	**0.500**/0.470	0.197/**0.240**	0.483/**0.497**	**0.500**/0.493	0.477/**0.503**	0.480/**0.523**	0.483/**0.517**
text-curie-001(13B)	0.497/**0.540**	0.480/**0.523**	0.090/**0.290**	0.213/**0.337**	0.490/**0.503**	**0.497**/0.460	0.473/**0.540**	0.453/**0.567**	0.470/**0.540**
text-davinci-003(175B)	0.820/**0.963**	0.763/**0.833**	0.420/**0.960**	0.267/**0.433**	0.677/**0.907**	0.287/**0.680**	**0.977**/0.967	**0.980**/0.978	0.877/**0.943**

Table 14: Negations experiments on four GPT-3 variants of different sizes. Note that metrics representing **neg/pos**.

5 Conclusion and Future Work

We investigate and analyze the sensitivity of prompts from the perspective of linguistics, covering four factors, *i.e.*, syntax, semantics, lexicon, and pragmatics, and their impact on various tasks. Based on the results, we streamline a list of guidelines as `Prompt Engineering 101` for prompt engineering. Via human evaluation, we show the effectiveness of the proposed guidelines. We further automated `Prompt Engineering 101` to build the automatic toolkit designed based on our guidelines. After replacing manual refinement with an automatic toolkit, we still observed an average improvement in performance. This further enhances the superiority of `Prompt Engineering 101`. There are more linguistic factors, for example, phonetics, phonology and typology, for us to study. Even for syntax, there remain aspects that we have not explored. Yet, we take the first step to exploring four of them. A more in-depth analysis for other factors on a wider range of tasks should be favorable. This work also provides another window into prompt engineering that regards prompts as a specific "langauge" to interact with LLMs. This

"language", like a dialect, is a variety of language that is characteristic of LLMs. From the perspective of behaviors, they have their own grammatical and phonological rules, linguistic features, and stylistic aspects, that is potentially different from the natural language. With mounting evidence, LLMs use prompts in a different way from humans. Nevertheless, there is no clear consensus on what kind of linguistic factor plays the most important role. On top of that, we hope that this work would call for future research into the intriguing problem.

6 Acknowledgements

This work is supported by the Talent Fund of Beijing Jiaotong University (2023XKRC006). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *arXiv preprint arXiv:2010.12421*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Gwern Branwen. 2020. Gpt-3 creative fiction. *The Website of Gwern Branwen*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2022. Grammatical error correction: A survey of the state of the art. *arXiv preprint arXiv:2211.05166*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Avia Efrat and Omer Levy. 2020. The turking test: Can language models understand instructions? *arXiv preprint arXiv:2010.11982*.

Dana Ferris. 1999. The case for grammar correction in l2 writing classes: A response to truscott (1996). *Journal of second language writing*, 8(1):1–11.

Daniel Fried, Nicholas Tomlin, Jennifer Hu, Roma Patel, and Aida Nematzadeh. 2022. Pragmatics in grounded language learning: Phenomena, tasks, and modeling approaches. *arXiv preprint arXiv:2211.08371*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Talmy Givón. 1978. Negation in language: pragmatics, function, ontology. In *Pragmatics*, pages 69–112. Brill.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. Semeval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398.

Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. 2022. A fine-grained comparison of pragmatic language understanding in humans and language models. *arXiv preprint arXiv:2212.06801*.

Joel Jang, Seonghyeon Ye, and Minjoon Seo. 2022. Can large language models truly understand prompts? a case study with negated prompts. *arXiv preprint arXiv:2209.12711*.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know. *Transactions of the Association for Computational Linguistics*, 8:423–438.

Mark D Johnson, Anthony Acevedo, and Leonardo Mercado. 2016. Vocabulary knowledge and vocabulary use in second language writing. *TESOL Journal*, 7(3):700–715.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Dilek Karakoç and Gül Durmuşoğlu Köse. 2017. The impact of vocabulary knowledge on reading, writing and proficiency scores of efl learners. *Journal of language and linguistic studies*, 13(1):352–378.

Gabriele Kasper and Carsten Roever. 2005. Pragmatics in second language learning. *Handbook of research in second language teaching and learning*, pages 317–334.

Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to gptk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612.

Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-goo Lee, Kang Min Yoo, and Taeuk Kim. 2022. Ground-truth labels matter: A deeper look into input-label demonstrations. *arXiv preprint arXiv:2205.12685*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Kepa Korta and John Perry. 2020. Pragmatics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2020 edition.

Claire Kramsch. 2014. Language and culture. *AILA review*, 27(1):30–55.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland, May. Association for Computational Linguistics.

John Lyons. 1981. *Language and linguistics*. Cambridge university press.

Aman Madaan and Amir Yazdanbakhsh. 2022. Text and patterns: For effective chain of thought, it takes two to tango. *arXiv preprint arXiv:2209.07686*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland, May. Association for Computational Linguistics.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.

Jonas Oppenlaender. 2022. A taxonomy of prompt modifiers for text-to-image generation. *arXiv preprint arXiv:2204.13988*.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Subhro Roy and Dan Roth. 2016. Solving general arithmetic word problems. *arXiv preprint arXiv:1608.01413*.

Tony Silva and Paul Kei Matsuda. 2012. *On second language writing*. Routledge.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Albert Webson and Ellie Pavlick. 2021. Do prompt-based models really understand the meaning of their prompts? *arXiv preprint arXiv:2109.01247*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Xiang Wei, Yufeng Chen, Ning Cheng, Xingyu Cui, Jinan Xu, and Wenjuan Han. 2024. CollabKG: A learnable human-machine-cooperative information extraction toolkit for (event) knowledge graph construction. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3490–3506, Torino, Italia, May. ELRA and ICCL.

Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *CHI Conference on Human Factors in Computing Systems*, pages 1–22.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Differentiable prompt makes pre-trained language models better few-shot learners. In *International Conference on Learning Representations*.

A Related Work

Successful practices usually require domain expertise and are extremely time-consuming—one needs to spend a significant amount of time on wording since a slight change in the textual input could have a huge impact on performance (Reynolds and McDonell, 2021). As a result, "engineering" these prompts has been one of the major challenges in deploying LLMs in real-world applications. (Webson and Pavlick, 2021). Using the recently introduced In-Context Learning (ICL), a PLM only needs to take in a few training examples and a test instance as context and directly decode the output without any updates to its parameters. However, using open-ended natural language as contextual instruction in ICL is a double-edged sword; while users can input anything and access a variety of generated sentences, they also must engage in cumbersome trials and errors when the resulting language quality is poor (Jiang et al., 2020; Liu et al., 2021). Successful practices usually require domain expertise and are extremely time-consuming—one needs to spend a significant amount of time on wording since a slight change in the textual input could have a huge impact on performance (Reynolds and McDonell, 2021). As a result, "engineering" these prompts has been one of the major challenges in deploying LLMs in real-world applications. (Webson and Pavlick, 2021). The study of prompt engineering can be traced back to T5 (Raffel et al., 2020) and GPT-2 (Radford et al., 2019), where single word task identifiers were utilized to distinguish between different tasks. The term "prompt engineering" was originally coined by (Branwen, 2020) when describing GPT-3's few-shot ICL capabilities. (Branwen, 2020) positioned prompting as an emerging technique for LLMs that were trained from Internet-scale data (Wu et al., 2022). It's recently shown that prompting is an effective method to elicit specific knowledge and abstractions needed to perform well on unseen tasks and bears potential to help end users interact with intelligent language models. Demonstrations in the prompt are well-studied by prior arts. (Lu et al., 2022) and (Kim et al., 2022) reveal the role of providing format clues, the sensitivity of the order of examples, and the importance of ground-truth labels, respectively. This work and many others along the same vein established foundation within the community to study the function of different language aspects in prompting. (Madaan and Yazdanbakhsh, 2022) systematically experiment with symbols, patterns, and text to study these items separately. Another line of research is Chain-of-Thought (CoT) designing (Wei et al., 2022). A manually designed rationale benefit the prompting performance. (Khashabi et al., 2022) share a similar basis with CoT (Wei et al., 2022). They claim that manually decomposing a task instruction into multiple simpler subtasks leads to better performance. Meanwhile, hobbyists and practitioners on the Internet have already engaged in prompt engineering. They discuss fantastic "tricks" and keywords and discover how to steer LLMs toward desirable aesthetic generations (Oppenlaender, 2022). For example, using "unreal engine" as a prompt will help add a hyper-realistic, 3D render quality to the image generation [8].

B Hyper-parameters

Hyper-parameters of the GPT-3 are listed in Tab. 15.

C Data Statistic

We show the data statistics for semantics analysis in Tab. 16 and lexicon analysis in Tab. 17.

[8]Aran Komatsuzaki's findings at https://twitter.com/arankomatsuzaki/status/1399471244760649729.

Hyper-parameter	Value
Max token	64
Top-p	1
Temperature	0
Frequency Penalty	0
Presence Penalty	0

Table 15: Hyper-parameters of the GPT-3 model we use.

Dataset	#Instances
ARC-Easy	300
PIQA	300
COPA	100
HellaSwag	300
StoryCloze	300
Lambada	300
WQ	300
TriviaQA	300
NQ	300

Table 16: Data statistic of experiments for impact analysis of semantics.

Dataset	#Instances
PIQA	50
ARC-Easy	50
HellaSwag	50
HellaSwag	300
SuperGlue	50
SuperGlue	100

Table 17: Data statistic of experiments for impact analysis of lexicon.

D Few-shot Setting for Syntax

Additionally, we study the effect of syntax on the few-shot setting, feeding GPT-3 with eight examples and the query. The word order of the questions in examples is shuffled for the perturbed prompts. See Tab. 18 for the results.

	MultiArith	StrategyQA
Few-Shot Setting		
Original	32.7	61.4
Perturbed	30.5	60.0
Zero-Shot Setting		
Original	16.2	59.3
Perturbed	3.5	47.3

Table 18: Comparison of original prompts *vs*. perturbed prompts with perturbations to syntax.

E Zero-Shot CoT for Lexicon

We also conduct the experiment to study lexicon's effect on the zero-shot CoT via translation substitution. The results are shown in Tab. 19.

	Zero-Shot	Zero-Shot CoT
Original	54.0	51.3
Perturbed	53.0	49.7

Table 19: Comparison of original prompts *vs*. perturbed prompts with perturbations to the lexicon. We show the effect via translation substitution.

F Model Generalization Validation

To verify whether our method has model generalization, we perform negation experiments on four different GPT-3 variants of different sizes (`text-ada-001`, `text-babbage-001`, `text-curie-001` and `text-davinci-003`). The results are shown in Tab. 20. We could find that the experimental group with negations generally performed worse than the control group. This conclusion is consistent with the experiments on `text-davinci-002`, so we consider our approach to be model generalizable.

	ARC-Easy	PIQA	COPA	HellaSwag	StoryCloze	Lambada	WQ	TriviaQA	NQ
text-ada-001(1.3B)	0.427/**0.467**	**0.533**/0.407	0.400/**0.440**	0.380/**0.400**	0.443/**0.497**	0.247/**0.323**	0.393/**0.453**	0.410/**0.473**	0.403/**0.557**
text-babbage-001(6.7B)	0.463/**0.527**	0.473/**0.523**	**0.500**/0.470	0.197/**0.240**	0.483/**0.497**	**0.500**/0.493	0.477/**0.503**	0.480/**0.523**	0.483/**0.517**
text-curie-001(13B)	0.497/**0.540**	0.480/**0.523**	0.090/**0.290**	0.213/**0.337**	0.490/**0.503**	**0.497**/0.460	0.473/**0.540**	0.453/**0.567**	0.470/**0.540**
text-davinci-003(175B)	0.820/**0.963**	0.763/**0.833**	0.420/**0.960**	0.267/**0.433**	0.677/**0.907**	0.287/**0.680**	**0.977**/0.967	**0.980**/0.978	0.877/**0.943**

Table 20: Negations experiments on four GPT-3 variants of different sizes. Note that metrics representing **neg/pos**.

G Results details

Tab. 21 provides the detailed results obtained using either our guidelines for human or automatic toolkit. The metrics used are micro accuracy and macro accuracy.

- **Micro accuracy** measures the probability of correctly identifying all samples across the seven tasks.
- **Macro accuracy** calculates the accuracy for each of the seven tasks and then averages them.

To obtain the final performance w.r.t. guidelines for humans, we calculate the accuracy for each human and then average their scores.

	micro acc	macro acc
baseline	54.73	54.57
w auto	56.04	55.23
w/o human guidelines	57.04	56.13
w human guidelines	57.46	56.54
w human and auto guidelines	**58.10**	**57.24**

Table 21: Detailed results with/without our human/auto guidelines.

H Human Study Details

Firstly, we released a questionnaire (See Fig. 3 for details) to collect prompts without our guidelines from 10 persons. At the beginning of the questionnaire, there is a general description and two examples of writing prompts. After that, we show the brief introduction of each task. After each introduction we provide 3 examples (the input and the expected output of the models) to help subjects better comprehend what the task aims at. At the end of each task we remind the subjects to write a general instruction of the task rather than a specific prompt based on the examples we provided. The subjects were required to write a prompt of at least 10 characters for each task. Finally we collected 10 valid results, and each result includes 7 valid prompts for all tasks.

Secondly, in order to obtain the prompts with our guidelines, we released another 10 questionnaires (see Fig. 4 for details) to collect refined prompts. The setting of the questionnaire is similar with the questionnaire we designed to collect baseline prompts, except for two exceptions. First, we modified the description and the examples in the beginning in order to collect refined prompts. Second, after the introduction and examples of each task, we also provided the original prompt and our guidelines to make subjects refine the prompts based on our guidelines. For each group of prompts written by the same person, we design a personalized questionnaire and ask one person to refine one group of prompts.

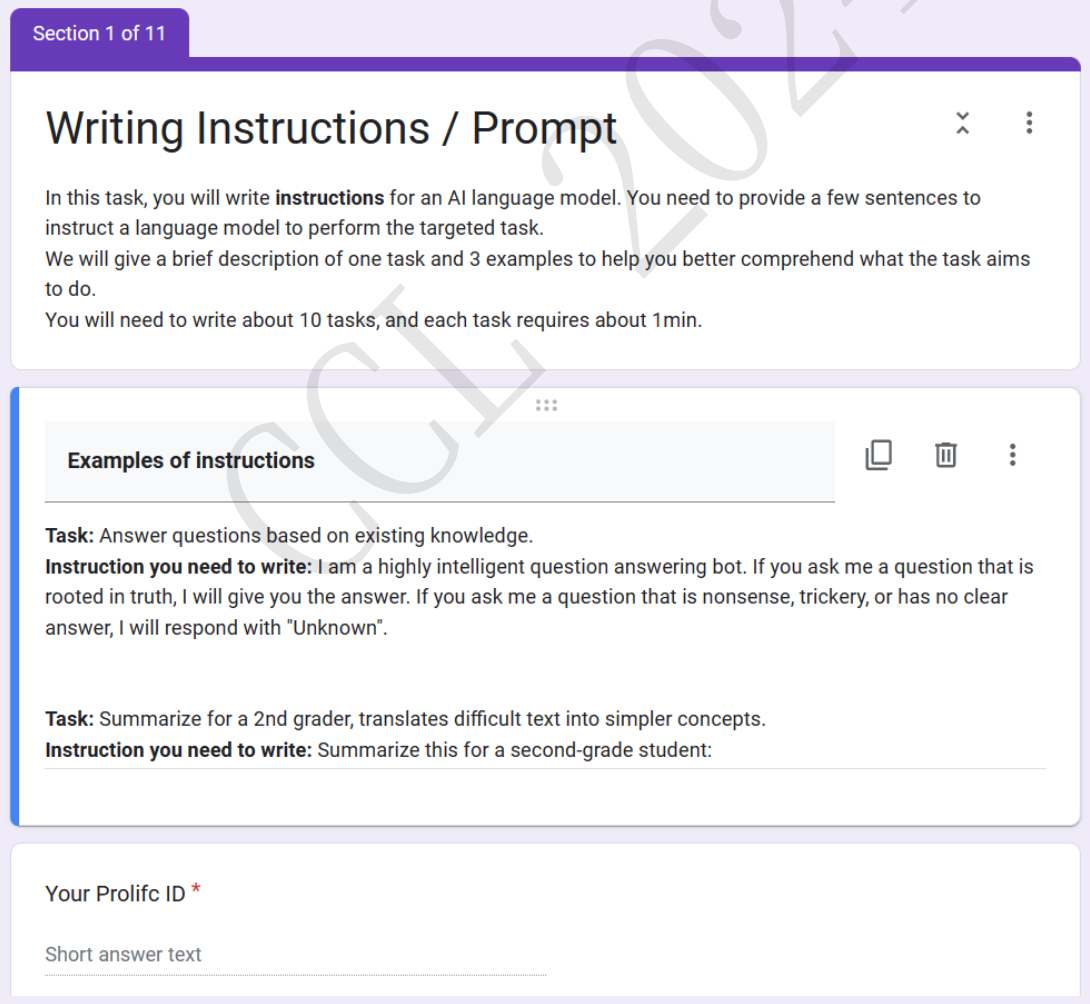

Figure 3: Interface of questionnaires without `Prompt Engineering 101`.

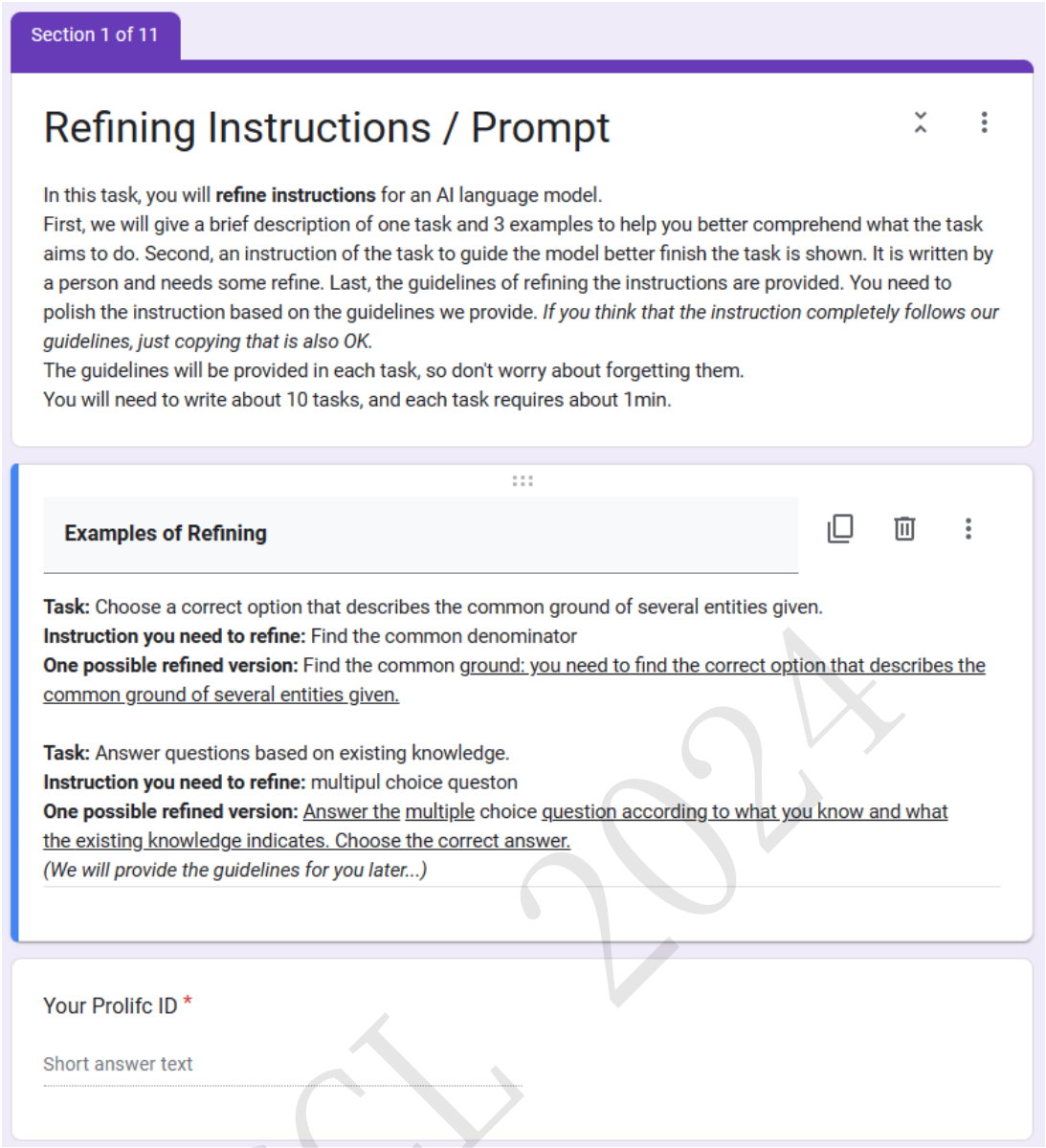

Figure 4: Interface of questionnaires with guidelines following `Prompt Engineering 101`.

I Case Study

Examples of original prompts *vs*. perturbed prompts with changes on syntax, semantics, lexicon, and pragmatics are shown in Tab. 22.

	Original Prompt	Perturbed Prompt
Syntax	***Prompt:*** For Halloween Deddy and her sister combined the candy they received. Deddy had 32 pieces of candy while her sister had 42. If they ate 35 pieces the first, how many pieces do they have left? A: Therefore, the answer (arabic numberals) is	***Prompt:*** Deddy the have the many they pieces Debby left? of combined her sister had how do candy Halloween candy while pieces If night, they 32 ate For 35 received. had 42. pieces and sister first her they A: Therefore, the answer (arabic numberals) is
	Expected Output: 39 ***Output:*** 39	***Expected Output:*** 39 ***Output:*** 74
Semantics	***Prompt:*** Generate the correct answer to the following question. Question: Earth, along with the other planets, revolves around the Sun. The planets revolve in a counterclockwise direction. The cause of the revolution is mostly due to which force? A. Gravitational B. Magnetic Answer:	***Prompt:*** Generate the incorrect answer to the following question. Question: Earth, along with the other planets, revolves around the Sun. The planets revolve in a counterclockwise direction. The cause of the revolution is mostly due to which force? A. Gravitational B. Magnetic Answer:
	Expected Output: A ***Output:*** A	***Expected Output:*** B ***Output:*** A
Lexicon	***Prompt:*** Are Christmas trees dissimilar to deciduous trees?	***Prompt:*** Are Christmas 树 dissimilar to deciduous trees?
	Expected Output: Yes. ***Output:*** yes, Christmas trees are dissimilar to deciduous trees.	***Expected Output:*** Yes. ***Output:*** no.
Pragmatics	***Prompt:*** Please choose the appropriatee answer for the following question from A, B, C, D. Which of the following sentences expresses the emotion like "anger"? A. I've used almost half of my printing money and it's the first day of the semester. #pissed B. When I chirp, shawty, chirp back. C. U know what's very pathetic? The fact that I dearly miss my professors but they probably forgot about me already. #sad #:(D. @user said, "Surely the #bitterness of #death is past." [2/2] Answer:	***Prompt:*** Please choose the appropriate answer for the following question from A, B, C, D. Which of the following sentences expresses the emotion like "volcanic eruption"? A. I've used almost half of my printing money and it's the first day of the semester. #pissed B. When I chirp, shawty, chirp back. C. U know what's very pathetic? The fact that I dearly miss my professors but they probably forgot about me already. #sad #:(D. @user said, "Surely the #bitterness of #death is past." [2/2] Answer:
	Expected Output: A ***Output:*** A	***Expected Output:*** A ***Output:*** B

Table 22: Examples of original prompts *vs*. perturbed prompts with changes on syntax, semantics, lexicon, and pragmatics. The text following ***Prompt*** represents the prompt. Original and perturbed prompts are denoted with red and green respectively.