Prompt engineering – effective communication with AI models

Konrad Hoza[1]

[1]Silesian University of Technology
Poland

Abstract— The article presents prompt engineering as a key element of effective communication between the user and artificial intelligence models based on large language models. Current artificial intelligence is already able to create and deliver material tailored to real needs, but the user does not always have the right skills to obtain this material. The article presents the main techniques of prompt engineering. The article aims to identify the mechanisms by which the structure and precision of prompts affect the quality, accuracy, and coherence of generated responses. As a result, knowledge of the principles of designing effective instructions is just as important as understanding how the expected result functions as feedback.

Keywords— Artificial intelligence, AI, Prompt engineering, Machine learning, Code generation

I. INTRODUCTION

Currently, the vast majority of information and knowledge is obtained from the internet. The revolution in the field of artificial intelligence has further intensified this process. However, artificial intelligence models are only a tool that opens up the possibility of better access to this knowledge. Improper use of these tools can lead to the opposite effect – incorrect generation of results, incorrect conclusions, and the spread of misinformation. Prompt engineering emerged with the development of large language models (LLMs) and has become crucial for their effective use in education, industry, and other fields. This field has become a bridge between the humanities related to language and the exact sciences in the field of computer science. The article will discuss the origin and evolution of prompt engineering, its goals, and the differences between "prompting" techniques and traditional user interfaces. The article will also include arguments for recognizing it as a separate research field, whose tools and methods are increasingly needed in interdisciplinary research. Writing prompts, also known as prompt engineering, refers to the design and optimization of inputs to generative models to obtain accurate and useful responses from artificial intelligence. The main problem with multi-dimensional models is their sensitivity to the command, which means that the quality of the result depends largely on the formulation used by the user when issuing the command to the model. In fact, the style of the command, as well as its content, can affect the level of correctness of the response. Therefore, scientists are actively researching how to interact with multidimensional models in order to most effectively formulate and issue commands to the model, so as to obtain the best possible results without wasting time on ineffective interactions.

II. THEORETICAL BASIS OF THE PROMPTS AND THEIR EFFECT ON LLMS

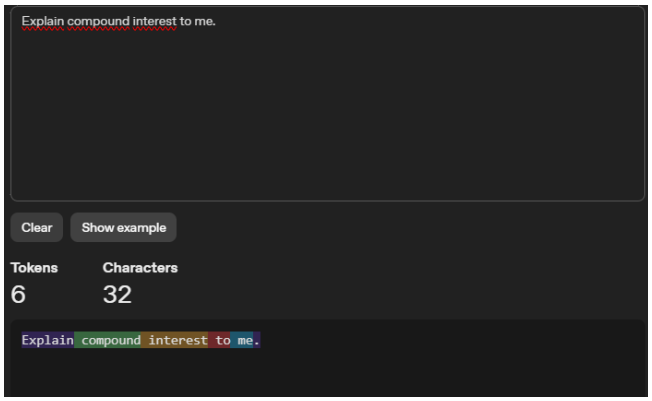
Have you ever used an AI tool and received an answer that didn't meet your expectations? Maybe the logo vision didn't match what the model generated? The problem may not have been with the limitations of the model, but with its lack of understanding of the received command.

The theoretical basis for prompts in the context of large language models (LLMs) stems directly from their architecture and the way they are trained. LLM are probabilistic models trained to predict the next token based on previous ones. This means that they do not "understand" the content like humans do, but they learn statistical relationships between individual elements of the command. The prompt is the input context that conditions the probability distribution of the generated

ASEJ - Scientific Journal of Bielsko-Biala School of Finance and Law
Volume 29, No 4 (2025), pages 6
https://doi.org/10.19192/wsfip.sj4.2025.19
Received: September 2025, Accepted: December 2025
Published: December 2025

responses. The model interprets the command not as an intention, but as a sequence of tokens that should be continued in the most probable way according to learned patterns. (Pawar i in., 2025). The process begins with tokenization, which is the division of the text into smaller units - tokens. Then each token is converted into a vector representation in a multidimensional space. These vectors pass through successive layers until they reach the mechanism that allows the model to calculate which parts of the context are most important when generating the next token. (Picture 1) As a result, each element of the prompt influences the final answer to a different degree. The entire prompt acts as an initial condition for the generation process, modifying the probabilities of subsequent possible tokens.

PICTURE 1-TOKENIZER



Source: (OpenAI, 2026)

The effect of the prompt on the model can be understood as a shift in the decision distribution. If the prompt contains precise instructions, examples, or a specific style, the model increases the likelihood of generating a response that conforms to that pattern. General or ambiguous commands, on the other hand, lead to more generative freedom, which increases variability and the risk of inaccuracy. The length and structure of the prompt are also important, as models have a limited context window in which they process information. The more ordered and unambiguous the context, the more stable the response. (Fagadau et al., 2024)

As a result, designing prompts, known as prompt engineering, is a practical application of knowledge about the statistical nature of language models and their information processing mechanisms.

III. OVERVIEW OF METHODS AND TECHNIQUES FOR DESIGNING EFFECTIVE PROMPTS

A. Basic methods

1) Zero-shot

The foundation of any interaction with an LLM is the ability to precisely convey intent. In its simplest form, it is about the so-called zero-shot prompts, i.e., single instructions without examples. Their effectiveness is based on transparency: a direct verb ("write", "translate", "summarize"), a clear definition of the expected format, and, often overlooked, revealing the context of why the task is being performed. (Pastor-Merino i in., 2025)

2) Few-shot

When zero-shot is not enough, you should use few-shot prompts, where you add two to five examples illustrating the desired transition from input to output. Examples serve as an implicit pattern: the model infers a rule from them and applies it to a new case. It is crucial to carefully select examples here – they should be representative, diverse, and stylistically consistent. Another important development is the explicit formatting of the response, i.e., indicating the output structure (list, table, JSON, code), which significantly reduces the need for additional data conversion from the given format to the desired one.

3) "Being clear and precise."

The third basic method is simply "being clear and precise." This includes formulating clear and specific commands that can guide the model towards generating the desired result. Most LLM architectures have a broad range of textual data, which is often a collection of observations from many authors. When this enormous amount of data is paired with a broad or imprecise command, the result is usually general in nature, which, although being appropriate in many contexts, may not be appropriately matched for each of them. On the other hand, a detailed and precise instruction allows the model to generate content that is more tailored to the unique requirements of the given scenario, because it reduces the uncertainty of the model and directs it towards specific data containing the required issues, thereby generating more specific and therefore more complete answers. For example, instead of asking vague requirements such as: "I want to understand the latest technological achievements," a more precise instruction would be: "I would like to understand the latest technological achievements, especially those related to artificial intelligence and machine learning."

4) Role-prompting

Role-prompting is another basic method in command engineering. It involves

assigning a specific role to the model, such as a helpful assistant or an individual with knowledge

expert. This method can be particularly effective in directing the model's responses

and ensuring that they are consistent with the desired outcome. For example, if the model is instructed to play the role of a history teacher, it will act as a historian, which can lead to more detailed and contextual descriptions and more precise answers to questions about historical events.

5) Triple quotes

Using triple quotes is a method used to separate different elements of a command part, making it easier for the model to recognize that everything inside the triple quotes is input material, not an additional command. This technique is particularly useful when working with long text passages, source code, tabular data, or content that contains its own quotation marks. This means that the more clearly the instruction segments and data are separated, the more stable and predictable the generation result is. Their effectiveness stems from the fact that in the training data, triple quotes often served as a marker for multiline text, for example in programming code

or documentation. The model learned to treat them as a clear logical boundary between sections. In practice, this reduces the risk of confusing the instructions with the analyzed content and limits ambiguity.

Resampling

Due to the large number of variables affecting the nature of multilingual models, it is often beneficial to perform several approaches. The resampling method involves running the model multiple times with the same command and selecting the best result for us. Language models are probabilistic in nature, so even with the same instruction, different processes can lead to slightly different reasoning paths. This method can help overcome the inherent variability of the model and increase the chances of obtaining a high-quality result.

6) One-shot or few-shot prompting

Single and repeated invocations are two important methods in command engineering. A single call refers to a method in which the model receives one command to solve a given problem. Similarly, few-shot prompting will provide a few examples. The choice between one and several calls often depends on the user, who must select the appropriate technique based on the command and the capabilities of the model. For simple tasks, such as face recognition, a model can achieve a result bordering on certainty using only one example, but for more complex tasks or less efficient models, few-shot prompting can provide additional context and guidance, which improves the model's performance.

B. Advanced techniques

The basic methods from the previous chapter can help you achieve satisfactory results. However, research indicates that when using multilingual models for complex tasks such as analysis or inference, the accuracy of the model's results still needs improvement. This section will discuss advanced message engineering techniques to help the model generate more detailed, accurate, high-quality results.

1) Chain-of-thought

The "chain of thought" (CoT) concept in LLM models is a relatively new step in their development, which has significantly improved the accuracy of natural language models in relation to various tasks requiring logical reasoning. The chain of thought is stimulated by providing intermediate steps of reasoning by the model, so as to outline its reasoning step by step, which will make it easier for the user to understand how the given result was achieved. Another advantage of the chain of thought is the organized reasoning of the language model, which can thus "reflect" in time and return to the correct line of reasoning. A command such as "Let's think step by step" or a series of manual demonstrations, each consisting of a question and a chain of reasoning leading to the answer, is sufficient. This seemingly innocuous instruction causes the model to first produce a series of intermediate conclusions that lead to the solution, rather than immediately generating the final answer. (Chen i in., 2025).

2) Golden chain-of-thought

Unlike random or simple examples, a golden Chain-of-Thought is a technique that has been manually developed by a person for a specific model in order to present an ideal, error-free, and most transparent path of reasoning for a given type of problem. The example in the prompt shows not only the correct input and output, but most importantly, the intermediate thought steps that are logical, complete, and free of unnecessary digressions. The golden Chain-of-Thought serves as a model that the model tries to imitate - the better and more representative the example, the greater the chance that the model will generate a high-quality response. This concept stems from the observation that models are extremely sensitive to the quality of examples in the prompt; a flawed or imprecise chain of thought can not only fail to help but even harm by reinforcing bad patterns. The golden Chain-of-Thought is used wherever we want to achieve maximum accuracy and repeatability of results – in mathematical, legal, or medical tasks, or in any context requiring rigorous reasoning.

3) Self-consistency

Self-consistency is a technique that combines Chain-of-Thought and resampling. It is manifested by repeatedly running the same prompt using the Chain-of-Thought, and then selecting the most coherent or most frequently occurring answer among the generated variants. Self-consistency uses this diversity to increase reliability – instead of relying on a single pass, results are aggregated across multiple independent trials, which allows for the elimination of random errors and the selection of the solution that appears most consistently.

4) Generated knowledge

This technique involves enriching the prompt with additional knowledge generated by the model itself before providing the final answer. In the first step, the model is asked to generate facts, hints, or contextual information related to the given query, and then this generated knowledge is added to the proper prompt, providing a solid basis for further reasoning. This technique is based on the assumption that the model can better solve problems when it first activates its explicit knowledge and focuses on the essential aspects. Generated knowledge improves the accuracy of responses in tasks requiring specialized knowledge, as it forces the model to activate the relevant internal resources before attempting to solve the problem.

5) Least-to-most

Least-to-most prompting is the next stage in the development of the "Chain-of-Thought" technique - breaking down a complex problem into simpler sub-problems. The questions are presented sequentially, from the easiest to the most difficult, where the answer obtained forms the basis for the next, more advanced question. This process continues until the main problem is resolved. This way, the model doesn't have to deal with the entire complexity at once, and each step builds on the solid foundations developed earlier. Least-to-most works well for tasks that require multi-step reasoning, planning, or problem-solving, where a hierarchical structure is natural – for example, in instructions, programming, or solving puzzles.

6) Tree of Thoughts

This technique aims to expand the concept of a chain of thought, allowing the model to explore many alternative paths of reasoning in a branching way, similar to a decision tree.

Instead of following one line of thought, the model generates several different potential lines of thought at each step that are candidates for the next step, and then evaluates them for usefulness or probability. These ratings are used to further develop the most promising branches, also allowing you to go back to previous nodes and choose other paths, as if the default path chosen at any point in the Tree of Thoughts turns out to be incorrect. This process is more complicated because it requires the whole scheme to be given in the command, instead of just "think step by step." Despite this, the Tree of Thoughts technique allows for systematic searching of the space of available solutions, which is extremely useful in tasks requiring planning, creative writing, strategic thinking, or proving theorems, where there are many possible paths to the goal.

7) Decomposed prompting

This technique involves intentionally dividing a complex task into smaller, independent subtasks that are solved separately, often using separate prompts. It differs from the least-to-most technique in that it can be parallel or hierarchical in nature – individual parts can be processed independently, and their results are later combined into a whole. The model can be asked to identify sub-problems and then solve each of them, and the answers to earlier questions are not necessarily needed for the next ones. Decomposed prompting is particularly useful in tasks such as document analysis, report generation, or handling queries that require knowledge from various fields – it allows you to focus on each aspect separately and then merge the conclusions into a coherent response.

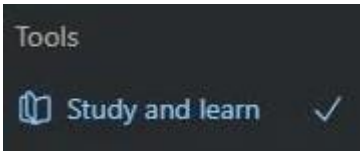
IV. PRACTICAL USE OF PROMPT ENGINEERING

Incorporating large language models (LLMs) into educational, professional, and creative fields has changed the way people perform cognitive and knowledge-based tasks. The impact of large language models, such as ChatGPT, Gemini, DeepSeek, and Claude, is evident not only in speeding up tasks but also in the way people approach and solve complex problems.

A. Educational context - increasing the effectiveness of learning

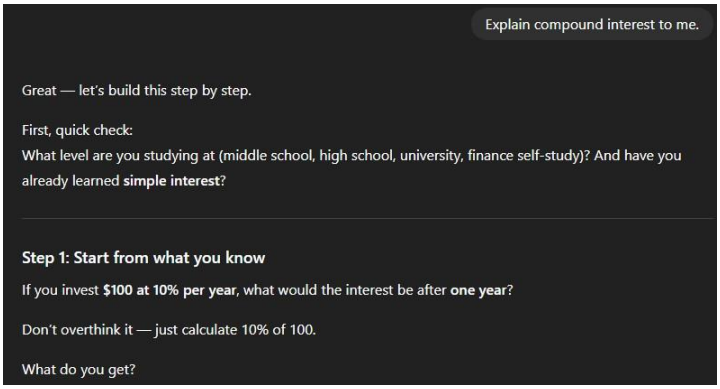
In the education sector, multilingual models have shown great potential in supporting teaching. There are also special modes dedicated solely to developing your knowledge in a chosen subject. They differ from the other modes in that their task is to analyze the level of required knowledge and ask the user questions to verify progress. The nature of AI-assisted learning feedback also supports self-regulation, critical thinking, and adaptive reasoning. This is not a groundbreaking technology, as a similar effect can be achieved by writing prompts yourself, but it is a convenience especially for people unfamiliar with the possibilities of prompt engineering. It is precisely thanks to these simplifications that tools like ChatGPT are increasingly being used as writing assistants, research mentors, and language tutors. They offer immediate explanations, constructive feedback, and translation assistance (Pictures 2 and 3). These features democratize access to knowledge, especially in environments with limited resources or in environments where English is not the native language.

PICTURE 2- STUDY AND LEARN MODE



Source: ChatGPT, 2026

PICTURE 2-INRERACTION IN STUDY AND LEARN MODE



Source: ChatGPT, 2026

B. Workplace efficiency - automation and expanded capabilities

In professional environments, natural language models can act as virtual coworkers, able to create documents, generate reports, prepare presentations, or code simple functions in many programming languages. Companies are also beginning to recognize the value of "artificial intelligence competence" as a workplace skill. In-house workshops and training programs are becoming more common to improve staff's qualifications in the effective use of AI. (Anam, 2025).

C. Creative productivity and idea generation

In creative fields such as marketing, media, and design, natural language models have proven effective in reducing the time needed to come up with an idea and overcoming creative blockages. Experts use artificial intelligence to generate lists of ideas, transform slogans, simulate brand voices, and even sketch out narrative arcs. According to research, companies that have implemented AI have found that they have approximately 53%–55% higher revenues and 48% higher added value than comparable companies that have not adopted this technology. Furthermore, increasing the intensity of AI use is associated with an even greater productivity premium of 74–76% in sales and up to 67% in added value, which highlights the significant benefits of deeper AI integration. In a dynamic view, AI adoption is associated with 9%–13% faster annual sales growth (Lee and Palmer, 2025). In combination with human originality, multilingual models act as catalysts - they suggest alternatives, explore extreme cases, and reveal insights that might have been overlooked. The iterative process between human creativity and AI suggestions expands the horizon of what is possible within limited time frames.

The impact of multi-dimensional models on human productivity is broad, multi-layered, and constantly evolving. From personalized learning and language proficiency to corporate effectiveness and creative development, these tools have proven transformative. However, their success depends on how effectively users formulate commands and interact with them. Therefore, continuous education and popularization of knowledge in the field of prompt engineering are required to achieve harmony and enable the full use of the potential of artificial intelligence to increase productivity.

V. ETHICAL ASPECTS OF PROMPT ENGINEERING

As prompt engineering becomes more widespread, ethical and socio-technical issues related to its use, including result manipulation, blurring of responsibility, and cognitive error, also arise. This section aims to emphasize that prompt engineering is not only a technical challenge, but also an area of social and ethical significance. The conclusions will show what research directions are available and what conclusions they can lead to.

A. Stereotypes based on data

Stereotypes in language models stem directly from the nature of the training data on which they were learned. The models learn statistical relationships present in large text collections, which reflect both knowledge and social prejudices. If historical data contain certain narrative patterns, inequalities, or biased representations of social groups, the model may reproduce them. A well-crafted prompt can limit or unconsciously reinforce these tendencies. As a result, prompt engineering requires cultural and ethical awareness, because the way a question is asked affects which patterns are activated in the response generation process (Vallverdú, Rzepka, and Sans Pinillos, 2025).

B. Manipulation

Precisely constructed prompts can be used not only to improve the quality of answers, but also for deliberate manipulation. The model generates content that is consistent and convincing stylistically, even when the content itself is questionable. Skillful control of context can lead to the creation of one-sided, misinforming narratives or narratives that reinforce specific beliefs (Hickerson and Perkins, 2025). In this context, prompt engineering becomes a tool with dual use: it can increase transparency, precision of communication, and democratization of knowledge, or be used to create persuasive content of questionable credibility and serve to create a narrative defined by its creator. The ethical use of AI therefore requires control over the purpose and consequences of the generated responses.

C. Responsibility for decisions and suggestions

In situations where language models support decision-making processes, the problem of distributed responsibility arises. The model's response is the result of both its training and the specific prompt. This means that responsibility lies not only with the creators of the system, but also with those designing the queries and the institutions implementing LLM-based solutions. It is necessary to introduce mechanisms of critical thinking, audit, and human supervision over the decisions implemented by the model. It is especially important to remember that the model does not have its own intentions or a real understanding of the consequences of its answers.

D. Professional competence

As generative models become more widely used, the ability to design effective and responsible prompts is becoming a new professional skill. It involves not only knowledge of query formulation techniques, but also understanding the limitations of models, the risk of hallucinations, and potential biases. In sectors such as education, law, finance, or medicine, it is necessary to develop training standards that will enable critical interpretation of the results generated by the system. Lack of proper competence can lead to overconfidence in the model and incorrect decisions.

E. Cognitive bias – asking leading questions

One of the significant risks is the phenomenon of confirming one's own beliefs through the way the prompt is formulated. Asking questions based on a preconceived thesis can lead to answers that reinforce a particular narrative, rather than subjecting it to critical analysis. The model, conditioned by the content of the question, generates an answer that is consistent with its interpretive framework. As a result, the user may receive an apparent confirmation of their own beliefs, even if they are incomplete or incorrect. Responsible design of objective prompts should therefore include formulating questions in an open manner and encouraging the presentation of alternative perspectives.

VI. SUMMARY

AI models have a huge potential for generating content, but utilizing these capabilities requires precise and well-constructed commands. The methods and techniques of communication with artificial intelligence mentioned here will certainly help in achieving more satisfactory results. Practical applications in education, professional work, and creative fields confirm the transformative impact of skillful writing prompts on productivity, learning quality, and creative processes. At the same time, the ethical analysis reveals significant challenges: the risk of reproducing stereotypes, susceptibility to manipulation, the problem of distributed responsibility, and the threat of confirming one's own beliefs through tendentiously formulated questions. In summary, prompt engineering is an interdisciplinary field that combines knowledge from linguistics and computer science, and its development and popularization are necessary for the full and responsible use of the potential of artificial intelligence in various spheres of life.

VII. REFERENCES

Anam, R.K. (2025) 'Prompt Engineering and the Effectiveness of Large Language Models in Enhancing Human Productivity'. OSF Preprints. Available at: https://doi.org/10.31219/osf.io/ad9y5_v1

Chen, B., Zhang, Z., Langrené, N. and Zhu, S. (2025) 'Unleashing the potential of prompt engineering for large language models', Patterns, 6(6), 101260. Available at: https://doi.org/10.1016/j.patter.2025.101260

Fagadau, I.D., Mariani, L., Micucci, D. and Riganelli, O. (2024) 'Analyzing Prompt Influence on Automated Method Generation: An Empirical Study with Copilot', arXiv preprint arXiv:2402.08430. Available at: https://doi.org/10.48550/arXiv.2402.08430

Hickerson, D. and Perkins, M. (2025) 'A Peek Behind the Curtain: Using Step-Around Prompt Engineering to Identify Bias and Misinformation in GenAI Models'. arXiv preprint. Available at: https://doi.org/10.48550/arXiv.2503.15205 (

Lee, D. and Palmer, E. (2025) 'Prompt engineering in higher education: A systematic review to help inform curricula', International Journal of Educational Technology in Higher Education, 22, 7. Available at: https://doi.org/10.1186/s41239-025-00503-7 (

OpenAI (no date) ChatGPT. Available at: https://chatgpt.com 2025).

OpenAI (no date) Tokenizer. Available at: https://platform.openai.com/tokenizer

Pastor-Merino, A., Martínez-Barbero, X., Vicente, M.R. and Domenech, J. (2025) 'Does AI boost firm productivity? A web scraping and LLMs approach', Telecommunications Policy, 50(2), 103138. Available at: https://doi.org/10.1016/j.telpol.2025.103138

Pawar, S., Apte, M.M., Jadhav, K., Palshikar, G.K. and Ramrakhiyani, N. (2025) 'Broken Words, Broken Performance: Effect of Tokenization on Performance of LLMs'. arXiv preprint. Available at: https://arxiv.org/abs/2512.21933

Vallverdú, J., Rzepka, R. and Sans Pinillos, A. (2025) 'Editorial: Prompts: the double-edged sword using AI', Frontiers in Artificial Intelligence, 8, 1756343. Available at: https://doi.org/10.3389/frai.2025.1756343