

NirDiamant /

Prompt_Engineering

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

This repository offers a comprehensive collection of tutorials and implementations for Prompt Engineering techniques, ranging from fundamental concepts to advanced strategies. It serves as an essential resource for mastering the art of effectively communicating with and leveraging large language models in AI applications.

View license

12 stars

1 fork

1 watching

Branches

Tags

Activity

Public repository

1 Branch

0 Tags

Go to file

Go to file

+

Add file

<> Code

NirDiamant

updated CONTRIBUTING

0594d88 · 1 hour ago

.github	copy files from private repo	1 hour ago
all_prompt_engineering_tec...	copy files from private repo	1 hour ago
.gitignore	copy files from private repo	1 hour ago
CONTRIBUTING.md	updated CONTRIBUTING	1 hour ago
LICENSE	copy files from private repo	1 hour ago
README.md	updated README	1 hour ago
requirements.txt	copy files from private repo	1 hour ago

README

License

PRs welcome

LinkedIn Connect

Follow @NirDiamantAI

Discord

Join our community

Support This Project:

Your sponsorship fuels innovation in prompt engineering development.

[Become a sponsor](#) to help maintain and expand this valuable resource!

Prompt Engineering Techniques: Comprehensive Repository for Development and Implementation

1/9

Welcome to one of the most extensive and dynamic collections of Prompt Engineering tutorials and implementations available today. This repository serves as a comprehensive resource for learning, building, and sharing prompt engineering techniques, ranging from basic concepts to advanced strategies for leveraging large language models.

Stay Updated!

Don't miss out on cutting-edge developments, new tutorials, and community insights!

[Subscribe to DiamantAI's Newsletter of Gen AI](#)


Introduction


Prompt engineering is at the forefront of artificial intelligence, revolutionizing the way we interact with and leverage AI technologies. This repository is designed to guide you through the development journey, from basic prompt structures to advanced, cutting-edge techniques.

Our goal is to provide a valuable resource for everyone - from beginners taking their first steps in AI to seasoned practitioners pushing the boundaries of what's possible. By offering a range of examples from foundational to complex, we aim to facilitate learning, experimentation, and innovation in the rapidly evolving field of prompt engineering.

Furthermore, this repository serves as a platform for showcasing innovative prompt engineering techniques. Whether you've developed a novel approach or found an innovative application for existing techniques, we encourage you to share your work with the community.

Related Projects

 Explore my [comprehensive guide on RAG techniques](#) to learn how to enhance AI systems with external knowledge retrieval, complementing language model capabilities with rich, up-to-date information.


 Dive into my [GenAI Agents Repository](#) for a wide range of AI agent implementations and tutorials, from simple conversational bots to complex, multi-agent systems for various applications.

A Community-Driven Knowledge Hub

This repository grows stronger with your contributions! Join our vibrant Discord community — the central hub for shaping and advancing this project together 🤝

[DiamantAI Discord Community](#)

Whether you're a novice eager to learn or an expert ready to share your knowledge, your insights can shape the future of prompt engineering. Join us to propose ideas, get feedback, and collaborate on innovative implementations. For contribution guidelines, please refer to our [CONTRIBUTING.md](#) file. Let's advance prompt engineering technology together!

 For discussions on GenAI, or to explore knowledge-sharing opportunities, feel free to [connect on LinkedIn](#).

Key Features

- 🎓 Learn prompt engineering techniques from beginner to advanced levels
- 🧠 Explore a wide range of prompt structures and applications
- 📖 Step-by-step tutorials and comprehensive documentation
- 🛠️ Practical, ready-to-use prompt implementations
- ⭐ Regular updates with the latest advancements in prompt engineering
- 🤝 Share your own prompt engineering creations with the community

Prompt Engineering Techniques

Explore our extensive list of prompt engineering techniques, ranging from basic to advanced:

🌱 Fundamental Concepts

1. [Introduction to Prompt Engineering](#)

Overview 🔍

A comprehensive introduction to the fundamental concepts of prompt engineering in the context of AI and language models.

Implementation 🛠️

Combines theoretical explanations with practical demonstrations, covering basic concepts, structured prompts, comparative analysis, and problem-solving applications.

2. [Basic Prompt Structures](#)

Overview 🔍

Explores two fundamental types of prompt structures: single-turn prompts and multi-turn prompts (conversations).

Implementation 🛠️

Uses OpenAI's GPT model and LangChain to demonstrate single-turn and multi-turn prompts, prompt templates, and conversation chains.

3. [Prompt Templates and Variables](#)

Overview 🔍

Introduces creating and using prompt templates with variables, focusing on Python and the Jinja2 templating engine.

Implementation 🛠️

Covers template creation, variable insertion, conditional content, list processing, and integration with the OpenAI API.

Core Techniques

4. [Zero-Shot Prompting](#)

Overview

Explores zero-shot prompting, allowing language models to perform tasks without specific examples or prior training.

Implementation

Demonstrates direct task specification, role-based prompting, format specification, and multi-step reasoning using OpenAI and LangChain.

5. [Few-Shot Learning and In-Context Learning](#)

Overview

Covers Few-Shot Learning and In-Context Learning techniques using OpenAI's GPT models and the LangChain library.

Implementation

Implements basic and advanced few-shot learning, in-context learning, and best practices for example selection and evaluation.

6. [Chain of Thought \(CoT\) Prompting](#)

Overview

Introduces Chain of Thought (CoT) prompting, encouraging AI models to break down complex problems into step-by-step reasoning processes.

Implementation

Covers basic and advanced CoT techniques, applying them to various problem-solving scenarios and comparing results with standard prompts.

Advanced Strategies

7. [Self-Consistency and Multiple Paths of Reasoning](#)

Overview

Explores techniques for generating diverse reasoning paths and aggregating results to improve AI-generated answers.

Implementation

Demonstrates designing diverse reasoning prompts, generating multiple responses, implementing aggregation methods, and applying self-consistency checks.

8. [Constrained and Guided Generation](#)

Overview 🔍

Focuses on techniques to set up constraints for model outputs and implement rule-based generation.

Implementation 🛠️

Uses LangChain's PromptTemplate for structured prompts, implements constraints, and explores rule-based generation techniques.

9. [Role Prompting](#)

Overview 🔍

Explores assigning specific roles to AI models and crafting effective role descriptions.

Implementation 🛠️

Demonstrates creating role-based prompts, assigning roles to AI models, and refining role descriptions for various scenarios.

🚀 Advanced Implementations

10. [Task Decomposition in Prompts](#)

Overview 🔍

Explores techniques for breaking down complex tasks and chaining subtasks in prompts.

Implementation 🛠️

Covers problem analysis, subtask definition, targeted prompt engineering, sequential execution, and result synthesis.

11. [Prompt Chaining and Sequencing](#)

Overview 🔍

Demonstrates how to connect multiple prompts and build logical flows for complex AI-driven tasks.

Implementation 🛠️

Explores basic prompt chaining, sequential prompting, dynamic prompt generation, and error handling within prompt chains.

12. [Instruction Engineering](#)

Overview 🔍

Focuses on crafting clear and effective instructions for language models, balancing specificity and generality.

Implementation 🛠️

Covers creating and refining instructions, experimenting with different structures, and implementing iterative improvement based on model responses.

Optimization and Refinement

13. [Prompt Optimization Techniques](#)

Overview 🔍

Explores advanced techniques for optimizing prompts, focusing on A/B testing and iterative refinement.

Implementation 🛠️

Demonstrates A/B testing of prompts, iterative refinement processes, and performance evaluation using relevant metrics.

14. [Handling Ambiguity and Improving Clarity](#)

Overview 🔍

Focuses on identifying and resolving ambiguous prompts and techniques for writing clearer prompts.

Implementation 🛠️

Covers analyzing ambiguous prompts, implementing strategies to resolve ambiguity, and exploring techniques for writing clearer prompts.

15. [Prompt Length and Complexity Management](#)

Overview 🔍

Explores techniques for managing prompt length and complexity when working with large language models.

Implementation 🛠️

Demonstrates techniques for balancing detail and conciseness, and strategies for handling long contexts including chunking, summarization, and iterative processing.

Specialized Applications

16. [Negative Prompting and Avoiding Undesired Outputs](#)

Overview 🔍

Explores negative prompting and techniques for avoiding undesired outputs from large language models.

Implementation 🛠️

Covers basic negative examples, explicit exclusions, constraint implementation using LangChain, and methods for evaluating and refining negative prompts.

17. [Prompt Formatting and Structure](#)

Overview 🔍

Explores various prompt formats and structural elements, demonstrating their impact on AI model responses.

Implementation 🛠️

Demonstrates creating various prompt formats, incorporating structural elements, and comparing responses from different prompt structures.

18. [Prompts for Specific Tasks](#)

Overview 🔍

Explores the creation and use of prompts for specific tasks: text summarization, question-answering, code generation, and creative writing.

Implementation 🛠️

Covers designing task-specific prompt templates, implementing them using LangChain, executing with sample inputs, and analyzing outputs for each task type.

🌐 Advanced Applications

19. [Multilingual and Cross-lingual Prompting](#)

Overview 🔍

Explores techniques for designing prompts that work effectively across multiple languages and for language translation tasks.

Implementation 🛠️

Covers creating multilingual prompts, implementing language detection and adaptation, designing cross-lingual translation prompts, and handling various writing systems and scripts.

20. [Ethical Considerations in Prompt Engineering](#)

Overview 🔍

Explores the ethical dimensions of prompt engineering, focusing on avoiding biases and creating inclusive and fair prompts.

Implementation 🛠️

Covers identifying biases in prompts, implementing strategies to create inclusive prompts, and methods to evaluate and improve the ethical quality of AI outputs.

21. [Prompt Security and Safety](#)

Overview 🔍

Focuses on preventing prompt injections and implementing content filters in prompts for safe and secure AI applications.

Implementation 🔧

Covers techniques for prompt injection prevention, content filtering implementation, and testing the effectiveness of security and safety measures.

22. [Evaluating Prompt Effectiveness](#)

Overview 🔍

Explores methods and techniques for evaluating the effectiveness of prompts in AI language models.

Implementation 🔧

Covers setting up evaluation metrics, implementing manual and automated evaluation techniques, and providing practical examples using OpenAI and LangChain.

Getting Started

To begin exploring and implementing prompt engineering techniques:

1. Clone this repository:

```
git clone https://github.com/NirDiamant/Prompt_Engineering.git
```



2. Navigate to the technique you're interested in:

```
cd all_prompt_engineering_techniques
```



3. Follow the detailed implementation guide in each technique's notebook.

Contributing

We welcome contributions from the community! If you have a new technique or improvement to suggest:

1. Fork the repository
2. Create your feature branch: `git checkout -b feature/AmazingFeature`
3. Commit your changes: `git commit -m 'Add some AmazingFeature'`
4. Push to the branch: `git push origin feature/AmazingFeature`
5. Open a pull request

License

This project is licensed under a custom non-commercial license - see the [LICENSE](#) file for details.

🌟 If you find this repository helpful, please consider giving it a star!

Releases



No releases published

Sponsor this project



NirDiamant

 Sponsor

[Learn more about GitHub Sponsors](#)

Packages

No packages published

Languages

