

# OptiPrompt

version 602

A conversational AI chatbot for enhancing user queries and assisting in generating improved questions.

[Explore the docs »](#)

[View Demo](#) · [Report Bug](#) · [Request Feature](#)

► Table of Contents

## About The Project

×

Choose local model

Baichuan-13B-Chat

Model name: Baichuan-13B-Chat

Select input type

☐ multi-step

☐ judge

☒ default


Customize instructions 🗨️

Press Ctrl+Enter to submit form • 0/500

Submit

Deploy

⋮

 **OptiPrompt**

Baichuan-13B-Chat loaded, enjoy your journey!

What is up? ➤

Welcome to the **OptiPrompt** project, your gateway to a world of intelligent conversation and assistance! 🤖 ✨

The **OptiPrompt** is an innovative chatbot powered by state-of-the-art natural language processing models. Whether you're seeking answers, code optimization suggestions, or even help with refining

your questions, our chatbot is here to assist you effectively. Designed for a variety of use cases, from code optimization to language model guidance, it's your go-to AI assistant.

Our project leverages the latest advancements in language modeling technology and provides you with a seamless conversational experience. Engage in productive discussions, refine your queries, and harness the power of AI to enhance your workflow.

Get ready to embark on an exciting journey with the **OptiPrompt**. Explore its capabilities, improve your communication, and boost your productivity. It's more than just a chatbot; it's your AI companion in the world of natural language understanding and generation.

Start your conversation with the **OptiPrompt** today and elevate your interactions to the next level! 🚀



[\(back to top\)](#)

## Built With

- Streamlit
- Langchain
- Baichuan-13b

[\(back to top\)](#)

## Getting Started

Before getting started with the project, ensure that you have the following prerequisites installed:

## Prerequisites

- **Python 3.9:** You'll need Python 3.10 to run this project. We recommend using Anaconda to manage your Python environments. You can create a Python 3.10 environment with the following command:

```
conda create -n infinite-fdu python=3.9
```

Activate the environment using:

```
conda activate infinite-fdu
```

# Installation

1. Clone the repository to your local machine using the following command:

```
git clone https://github.com/Infinite-FDU/OptiPrompt
```

2. Navigate to the project directory
3. Install the project dependencies from the `requirements.txt` file:

```
pip install --pre --upgrade bigdl-llm[all]  
pip install -r requirements.txt
```

## Check for CUDA Compatibility

Before running the project, it's essential to check whether your computer satisfies the requirements for CUDA support. CUDA is a parallel computing platform and application programming interface (API) model created by NVIDIA.

Some of the deep learning libraries used in this project, such as PyTorch, can leverage CUDA for GPU acceleration. If you have an NVIDIA GPU and wish to enable GPU support, you should ensure that your GPU is compatible with CUDA and that you have the appropriate NVIDIA drivers installed.

If your GPU is not CUDA-compatible or you encounter issues with CUDA, **remove the torch based packages in the `requirements.txt` and install on your own.**

[\(back to top\)](#)

# Usage

## Prompt Engineering Overview

Welcome to the OptiPrompt, your intelligent assistant for prompt engineering. As a prompt engineer, your role is to craft and refine prompts to elicit precise and meaningful responses from language models. This chatbot offers a range of optimization options to assist you in your task.

# 1. Default Optimization

- **User Input:** *Enter your original prompt here.*
- **Optimized Output:** *The chatbot will refine your prompt to make it more explicit and effective for language models.*

# 2. Code Optimization

- **User Input:** *Share your code snippet or programming-related query.*
- **Optimized Output:** *The chatbot will review and enhance your code for better readability and functionality.*

# 3. Judge Prompt Word

- **User Input:** *Submit a prompt word or phrase for evaluation.*
- **Evaluation Output:** *The chatbot will assess the quality of your prompt and provide a score along with feedback.*

# 4. Multi-Step Question

- **User Input:** *Pose a complex question that can be divided into smaller steps.*
- **Optimized Output:** *The chatbot will break down your question into manageable parts, perfect for step-by-step exploration.*

**Note:** This project has been tested using artificial intelligence, Claude, for prompt optimization. It compares user inputs to optimized inputs and achieves a perfect score in **39 out of 39 tests**. For detailed information, please refer to the "test\_default.ipynb" and "output.json" files.

## Custom Instructions for Persistent Memory

The OptiPrompt empowers you with the ability to provide custom instructions, allowing you to whisper specific guidance to the language model. These custom instructions serve as a form of persistent memory, influencing the behavior of the chatbot across different sessions.

## How Custom Instructions Work

1. **Customize Your Guidance:** Craft personalized instructions in plain text to guide the language model in a particular direction. You can provide context, preferences, or specific expectations.

2. **Save to Local File:** Your custom instructions are saved to a local file, ensuring they persist even after closing the chatbot session. This file acts as a repository of your guidance.
3. **Influence Model Behavior:** During subsequent interactions, the chatbot will refer to your saved custom instructions, effectively remembering your preferences and following your guidance.
4. **Adapt and Refine:** You can update and refine your custom instructions over time, enabling the language model to adapt to your evolving needs and preferences.

## Enhance Your Conversations

Custom instructions provide a powerful way to tailor your interactions with the OptiPrompt. Whether you want to fine-tune responses, shape the conversation, or achieve specific outcomes, your instructions serve as a valuable resource for persistent memory and guidance.

Harness the potential of custom instructions to create more meaningful and personalized conversations with the chatbot, making it a truly adaptable assistant that understands your unique requirements.

*For more examples, please refer to the [Documentation](#)*

[\(back to top\)](#)

# Roadmap

## Version 1.0.0 (Initial Release)

- ✓ Streamlit app setup with customizable settings and styling.
- ✓ Integration of Transformers LLM with support for model selection.
- ✓ Four distinct input types: multi-step, judge, code, and default.
- ✓ User-friendly interface for prompt engineering with real-time previews.
- ✓ System message functionality for context-aware interactions.
- ✓ Chat history display for tracking conversations.
- ✓ Save customized instructions to a local file for future sessions.
- ✓ Documentation and GitHub repository setup.

## Future Versions

- ☐ Continued language support expansion.

- ☐ Integration with other local models.
- ☐ Collaboration with domain-specific experts for specialized knowledge.
- ☐ Exploration of educational and research applications.
- ☐ User feedback sessions for continuous improvement.