

# Istruzioni per l'installazione e l'uso di Prompt Perfezionatore

## Requisiti di sistema

- Python 3.7 o superiore
- Connessione internet (per l'uso delle API DeepSeek)
- Chiave API di DeepSeek

## Installazione

### 1. Clona o scarica il repository

```
bash
git clone https://github.com/tuonome/prompt-perfezionatore.git
cd prompt-perfezionatore
```

### 2. Crea un ambiente virtuale (consigliato)

```
bash
python -m venv venv
```

Attiva l'ambiente virtuale:

- Su Windows:

```
bash
venv\Scripts\activate
```

- Su macOS/Linux:

```
bash
source venv/bin/activate
```

### 3. Installa le dipendenze

```
bash
pip install -r requirements.txt
```

Altrimenti, installa manualmente i pacchetti necessari:

```
bash
pip install gradio openai python-dotenv spacy
python -m spacy download it_core_news_sm
```

### 4. Configura la chiave API

Crea un file `.env` nella directory principale del progetto con la tua chiave API di DeepSeek:

```
DEEPSEEK_API_KEY="la-tua-chiave-api-qui"
```

## Esecuzione

```
bash
python prompt_perfezionatore.py
```

Dopo l'avvio, l'interfaccia web sarà accessibile all'indirizzo: <http://127.0.0.1:7860>

## Utilizzo

1. Inserisci il tuo prompt originale nel campo di testo a sinistra
2. Clicca sul pulsante "🧙 Migliora il Prompt"
3. Visualizza il prompt migliorato nel campo di testo a destra
4. Esplora i suggerimenti di miglioramento e l'analisi dettagliata nelle sezioni espandibili

## Personalizzazione

### Cambiare il modello di linguaggio

Per utilizzare un modello diverso di DeepSeek, modifica la riga nel metodo `migliora_prompt`:

```
python

data = {
    "model": "deepseek-chat", # Cambia qui il modello
    # ...
}
```

## Utilizzare un modello locale

Per utilizzare un modello locale invece delle API OpenAI, puoi modificare il metodo `migliora_prompt` per utilizzare librerie come transformers (Hugging Face) o LlamaCpp. Ad esempio:

```
python

from transformers import pipeline

# Inizializzazione del modello (da aggiungere al costruttore)
self.modello_locale = pipeline("text2text-generation", model="mistralai/mistral-7b-instruct-v0.1")

# Modifica al metodo migliora_prompt
def migliora_prompt(self, prompt):
    # ... codice esistente ...

    try:
        # Sostituisci la chiamata API con il modello locale
        risultato = self.modello_locale(
            f"Sei un esperto di prompt engineering. Migliora il seguente prompt:\n{prompt}",
            max_length=2048
        )

        testo_generato = risultato[0]['generated_text']

        # Elaborazione del risultato...
        # ... resto del codice ...
```

## Modificare l'interfaccia

Per personalizzare l'interfaccia, modifica la funzione `crea_interfaccia()`. Puoi cambiare colori, layout e aggiungere nuove funzionalità utilizzando la documentazione di Gradio come riferimento.

## Aggiungere nuove analisi

Per aggiungere nuove metriche di analisi, estendi il metodo `analizza_prompt()` con le tue funzioni personalizzate.

## Risoluzione dei problemi

### L'applicazione si avvia ma non riesce a migliorare i prompt

- Verifica che la chiave API di DeepSeek sia corretta nel file `.env`
- Controlla la connessione internet
- Verifica i log per eventuali errori di autorizzazione

### Errori relativi al modello spaCy

- Assicurati di aver installato il modello italiano con `python -m spacy download it_core_news_sm`
- Se continui ad avere problemi, puoi disabilitare temporaneamente l'analisi grammaticale modificando `nlp = None`

### Problemi di memoria con modelli locali

- Se utilizzi modelli locali e riscontri problemi di memoria, considera l'uso di modelli più leggeri o di tecniche di quantizzazione

## Estensioni future

- Integrazione con altri provider di modelli (es. Mistral AI, Anthropic)
- Supporto multilingua per l'analisi grammaticale
- Salvataggio di prompt nella cronologia
- Funzionalità di esportazione/importazione di prompt
- Visualizzazione grafica dei miglioramenti