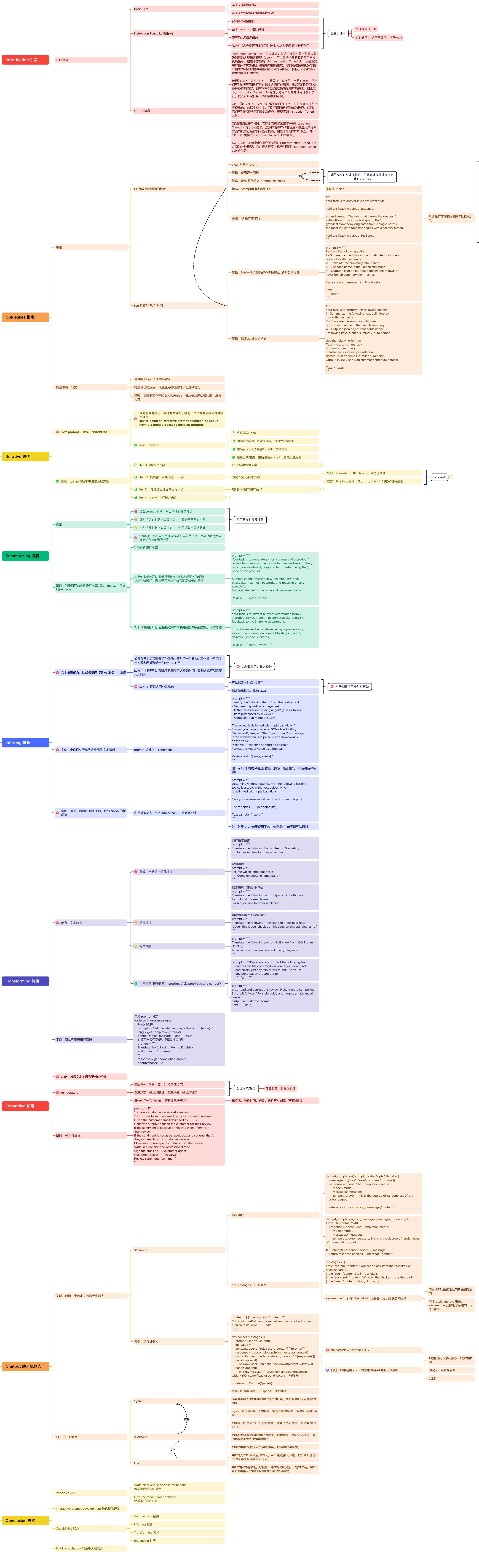


Prompt Engineering 提示工程师 by Kevin



```
def get_completion(prompt, model="gpt-3.5-turbo"):
    messages = [{"role": "user", "content": prompt}]
    response = openai.ChatCompletion.create(
        model=model,
        messages=messages,
        temperature=1, # this is the degree of randomness of the model's output
    )
    return response.choices[0].message["content"]

def get_completion_from_messages(messages, model="gpt-3.5-turbo", temperature=0):
    response = openai.ChatCompletion.create(
        model=model,
        messages=messages,
        temperature=temperature, # this is the degree of randomness of the model's output
    )
    # print(response.choices[0].message["content"])
    return response.choices[0].message["content"]

messages = [
    {"role": "system", "content": "You are an assistant that speaks like Shakespeare."},
    {"role": "user", "content": "tell me a joke"},
    {"role": "assistant", "content": "Why did the chicken cross the road?"},
    {"role": "user", "content": "I don't know!" }
]
```

来自 twitter 用户的建议:  
当引用一个很长的上下文，最好是先放上下文再放指示，如  
Context: `` `xxxxx` ``  
为一段关于 Context 的总结，最多50字。  
否则对指示的执行就不会很好。

ChatGPT 是对用户来说是隐藏的  
对于 assistant role 来说，  
system role 像是它旁边的一个“低语者”

问题: 如果给出了 api 的文本数据该如何让它继续?  
切割任务，避免超过api的文本限制  
限定gpt 回答的字数  
其他?