

Prefer using raw HTTP requests instead of wrappers

```
import requests
import json
import time

def translate_text(text, target_language, special=False):
    if not text or not text.strip():
        return ""

    if target_language == 'en':
        print(f"  Skipping translation for English: {text[:50]}...")
        return text

    print(f"  Translating text: {text[:50]}...")

    retries = 3
    for attempt in range(retries):
        try:
            response = client.chat.completions.create(
                model=MODEL_NAME,
                messages=[
                    {"role": "system", "content": create_translation_prompt(target_language, special)},
                    {"role": "user", "content": text}
                ],
                stream=False
            )
            if not response or not response.choices or not response.choices[0].message.content:
                print(f"  Error: Translation response is empty or invalid: {response}")
            if response and response.choices:
                translated_text = response.choices[0].message.content
                return translated_text
            else:
                print(f"  Translation failed on attempt {attempt + 1}.")
                if attempt == retries - 1:
                    return None
        except Exception as e:
            print(f"  Translation failed with error on attempt {attempt + 1}: {e}")
            if attempt == retries - 1:
                return None
            time.sleep(1)  # Wait before retrying
    return None
```

Error:

```
Translation failed with error on attempt 1: Expecting value: line 5 column 1 (char 4)
```

This error indicates that the DeepSeek API is returning a response that is not valid JSON, possibly HTML or some other format. This is unexpected, as the API is expected to return JSON. The issue might be due to a temporary API problem, rate limiting, or an issue with the prompt. It's important to handle this gracefully by logging the error and potentially retrying.