

# 大语言模型在经济学中的应用

## 从 API 调用出发的学术应用实例

陆知雨

中国财政发展协同创新中心

财政学基地班 22

2025 年 3 月 19 日

引言

大语言模型的  
代码调用

如何强化大语  
言模型在特定  
领域的理解力

在经济学领域  
应用的具体案  
例

基于大语言模  
型的开源项目  
分享

- ① 引言
- ② 大语言模型的代码调用
- ③ 如何强化大语言模型在特定领域的理解力
- ④ 在经济学领域应用的具体案例
- ⑤ 基于大语言模型的开源项目分享

## 1 引言

## 2 大语言模型的代码调用

## 3 如何强化大语言模型在特定领域的理解力

Prompt Engineering（提示词工程）

RAG（检索增强生成）

Fine-Tuning（微调）

## 4 在经济学领域应用的具体案例

## 5 基于大语言模型的开源项目分享

Chatbox & CherryStudio

PDF2zh

Ollama

其他

## 引言

大语言模型的  
代码调用

如何强化大语  
言模型在特定  
领域的理解力

在经济学领域  
应用的具体案  
例

基于大语言模  
型的开源项目  
分享

- 文本是知识的载体
- 合理处理文本，是构造解释、被解释变量的关键
- 当前处理文本数据以文本匹配为核心：
  - ① 使用正则表达式提取 CEO 简历数据
  - ② 使用情感词词典判断媒体情绪
  - ③ 使用数字化转型词典判断企业是否发生数字化转型
- 在面临更加“复杂”的文本环境时，由于难以穷尽各类匹配情况，研究者需要手动操作进行提取。
- 但是，机器可以替代重复简单劳动！

# 示例 1

- 如何判断普通网民的情绪？
- 与媒体情绪的区别：网络留言更为随意，文本样式多变
- 例如（摘自东方财富股吧）：
  - ① 工行没吃饭呢林黛玉似的好柔弱
  - ② 顶王粉丝光荣站岗

# 示例 1

- 如何判断普通网民的情绪？
- 与媒体情绪的区别：网络留言更为随意，文本样式多变
- 例如（摘自东方财富股吧）：
  - ① 工行没吃饭呢林黛玉似的好柔弱
  - ② 顶王粉丝光荣站岗

请你判断这段股民评论属于：积极、消极还是相对中性？#留言：工行没吃饭呢林黛玉似的好柔弱



Thought for 18 seconds

这条股民评论属于消极。

分析依据：

1. **负面比喻**：用“林黛玉似的”（体弱多病、多愁善感的形象）形容工行（工商银行），暗示其股价或市场表现疲软无力。
2. **情绪指向**：“没吃饭”和“好柔弱”带有调侃和批评意味，表达对工行走势弱于预期的不满或失望。
3. **语境联想**：股民通常期待股票或市场表现强劲，用“柔弱”等反向描述直接传递负面情绪。

因此，评论整体通过隐喻和贬义表达了对工行表现的不认可，属于消极情绪。

## 示例 1

- 如何判断普通网民的情绪？
- 与媒体文本的区别：网络留言更为随意，文本样式多变
- 例如（摘自东方财富股吧）：
  - ① 工行没吃饭呢林黛玉似的好柔弱
  - ② 顶王粉丝光荣站岗

请你判断这段股民评论属于：积极、消极还是相对中性？#留言：顶王粉丝光荣站岗



Thought for 19 seconds

这条评论属于消极。

分析：

1. “\*\*站岗\*\*”是股市常见俚语，指投资者在高位买入后被套牢，无法脱身，带有明显的负面含义。
2. “\*\*光荣\*\*”在此处为反讽，表面上看似褒义，实则强调无奈或自嘲，暗示粉丝因盲目追随（“顶王”）而蒙受损失。
3. 整体语气通过矛盾修辞（用积极词汇描述负面情境）强化了消极情绪，反映了对投资决策的懊悔或调侃。

## 示例 2

- 如何提取文本中的关键信息？
- 当目标文本格式不统一，正则表达式提取就变得极为复杂
- 例如：从政府工作报告提取经济增长目标值
- 使用正则表达式流程：
  - ① 读取多篇文献，找到几类常见用语，并转译成正则表达式
  - ② 使用正则表达式进行匹配
  - ③ 匹配后检查未匹配到的城市的文本，完善正则表达式，返回步骤 2
- 使用大语言模型：





## ① 引言

## ② 大语言模型的代码调用

## ③ 如何强化大语言模型在特定领域的理解力

Prompt Engineering（提示词工程）

RAG（检索增强生成）

Fine-Tuning（微调）

## ④ 在经济学领域应用的具体案例

## ⑤ 基于大语言模型的开源项目分享

Chatbox & CherryStudio

PDF2zh

Ollama

其他

# 代码调用与网页调用对比

- 但是，如果待处理样本量过大，从网页逐个处理是否反而繁琐？
- 机器可以替代重复简单劳动！

# 代码调用与网页调用对比

## 引言

## 大语言模型的代码调用

## 如何强化大语言模型在特定领域的理解力

## 在经济学领域应用的具体案例

## 基于大语言模型的开源项目分享

- 但是，如果待处理样本量过大，从网页逐个处理是否反而繁琐？
- 机器可以替代重复简单劳动！
- 实际上，除了相对易用的网页端口，厂商还提供了 API（编程接口），供开发者在代码中嵌入大语言模型。这为批量处理文本提供条件。

请你判断这段股民评论属于：积极、消极还是相对中性？#前言：顶王粉丝光荣站岗



Thought for 19 seconds

这条评论属于消极。

分析：

1. “站岗”是股市常见俚语，指投资者在高位买入后被套牢，无法脱身，带有明显的负面含义。
2. “光荣”在此处为反讽，表面上看似自豪，实则强调无奈或自嘲，暗示粉丝因盲目追随（“顶王”）而蒙受损失。
3. 整体语气通过矛盾修辞（用积极词汇描述负面情绪）强化了消极情绪，反映了对投资决策的懊悔或调侃。

```
from openai import OpenAI

# 填入你的实际 API Key
client = OpenAI(api_key="DeepSeek API Key", base_url="https://api.deepseek.com")

prompt = "你是一名文本情感分析专家"
input_text = "请判断这段股民评论属于：积极、消极还是相对中性？#前言：顶王粉丝光荣站岗"
completion = client.chat.completions.create(
    model="deepseek-reasoner",
    messages=[
        {"role": "system", "content": prompt},
        {
            "role": "user",
            "content": input_text
        }
    ],
    temperature=1,
    max_tokens=2048, # 修改为 max_tokens
    top_p=1,
    frequency_penalty=0,
    presence_penalty=0
)

# 打印输出内容
print(completion.choices[0].message.content)
```

这段评论中的“顶王粉丝光荣站岗”属于==消极==情绪。

==分析依据==：

1. “站岗”== 在股市语境中常被投资者在高位买入股票后股价下跌，被套牢时持有（类似民兵站岗无法离开），带有套牢的负面含义。
2. “==光荣==” 此处为反讽，表面看似自豪，实则表达无奈或有亏待头寸的无奈。
3. ==整体语境== 结合股民社区常见表达习惯，该评论通过看似积极的方式反映对持仓亏损的不满、情绪化消极。

# 主流大语言模型 API Key 的获取

- **国内厂商:** 一般而言, 国内厂商的 API 平台与大语言模型平台在相邻页面。以 deepseek 为例



# 主流大语言模型 API Key 的获取

- 国内厂商: 一般而言, 国内厂商的 API 平台与大语言模型平台在相邻页面。以 deepseek 为例

deepseek Platform

Usage

API keys 1

Top up

Billing

Docs

FAQ

Pricing

Integration

Contact us

Profile

### API keys

Your API keys are listed below. The API key is only visible and can be copied once at creation. Save it securely. Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, DeepSeek may also automatically disable any API key that we've found has leaked publicly. Usage of API keys created before April 25, 2024, was not tracked.

Name	Key	Created	Last used		
translation	sk-51cd2*****04be	2025-03-02	2025-03-06	✎	🗑
cherry studio	sk-59714*****b52a	2025-03-04	2025-03-18	✎	🗑
pdf	sk-f669b*****e83e	2025-03-04	2025-03-05	✎	🗑
trae	sk-755eb*****148c	2025-03-15	2025-03-15	✎	🗑

Create new API key 2

# 主流大语言模型 API Key 的获取

## 引言

## 大语言模型的 代码调用

## 如何强化大语 言模型在特定 领域的理解力

## 在经济学领域 应用的具体案 例

## 基于大语言模 型的开源项目 分享

- **国外厂商:** 通常需要通过境外信用卡支付, 对网络条件有一定要求。

解决方案 (仅作参考):

- ① 使用虚拟信用卡<https://yeka.ai>, 优化网络环境。
- ② 使用代理商<https://openrouter.ai>

## API 参考定价 (2025-03-09)

## 引言

大语言模型的  
代码调用如何强化大语  
言模型在特定  
领域的理解力在经济学领域  
应用的具体案  
例基于大语言模  
型的开源项目  
分享

机构	模型	输入/输出价格 (美元/百万 tokens)
阿里	Qwen-max	1.6 / 6.4
	Qwen-plus	0.4 / 1.2
OpenAI	GPT-4o	2.5 / 10
	GPT-4o-mini	0.15 / 0.6
	o1	15 / 60
谷歌	Gemini 2.0 Flash	0.1 / 0.4
	Gemini 2.0 Flash-Lite	0.075 / 0.3
	Gemini 1.5 Flash	0.075 / 0.3
Anthropic	Claude 3.7 Sonnet	3 / 15
	Claude 3.5 Haiku	0.8 / 4
	Claude 3 Sonnet	15 / 75
深度求索	deepseek-chat	0.27 / 1.1
	deepseek-reasoner	0.55 / 2.19
月之暗面	kimi-latest (8k-128k)	1.62-8.1 / 1.62-8.1
字节跳动	Doubao-pro (4k-256k)	0.1-0.675 / 0.27-1.215

注意：实际使用中，不同厂商有不同优惠政策，成本可能更低。

输入价格：上传至模型的数据收费；输出价格：模型返回结果的数据收费。

**tokens** 为模型处理文本的基本计量单位，通常一个汉字约等于 0.6 个 tokens，一个英文单词约等于 0.3 个 tokens。

# 代码详细说明 (Python)

使用 **openai** 库: 目前主流厂商的 API 调用都可以通过 **openai** 库实现, 因此仅展示 **openai** 库的使用方法。

```
from openai import OpenAI
client = OpenAI() # 初始化OpenAI客户端, 可填入API Key与url网址
prompt = "You are a helpful assistant" # 系统角色提示
input_text = "What is the capital of France?" # 用户输入问题
completion = client.chat.completions.create(
    model="gpt-4o", # 指定模型
    messages=[
        {"role": "system", "content": prompt}, # 系统角色消息
        {"role": "user", "content": inquire1}, # 第一轮对话用户角色消息
        {"role": "assistant", "content": output1}, # AI返回消息
        {"role": "user", "content": input_text} # 第二轮对话输入信息
    ],
    temperature=1, # 温度1为最高
    max_tokens=2048, # 限制生成最大长度
    top_p=1, # 采样策略1为全概率
    frequency_penalty=0, # 禁用重复惩罚
    presence_penalty=0 # 禁用主题惩罚
)
print(completion.choices[0].message.content) # 输出首个响应内容
#输出示例: The capital of France is Paris.
```



# 代码详细说明: 重要参数解析

## 引言

### 大语言模型的 代码调用

### 如何强化大语 言模型在特定 领域的理解力

### 在经济学领域 应用的具体案 例

### 基于大语言模 型的开源项目 分享

- **model**: 指定模型版本, 如"gpt-4o"、"deepseek-chat"。
- **messages**: 构建对话历史 (系统、用户、助手)。
- **temperature**: 控制随机性, 数值越高生成越多样。
- **top\_p**: 随机采样策略, 作用与 temperature 相同。
- **max\_tokens**: 限制返回文本的最大长度。
- **frequency\_penalty** & **presence\_penalty**: 调整重复率和新话题引入。

**建议:** 根据任务需求调整参数, 如创意写作可提高 temperature, 事实查询则建议降低 temperature, 但是 temperature 与 top\_p 不能同时修改。

## API 调用进阶 2: 多线程调用

### 引言

### 大语言模型的 代码调用

### 如何强化大语 言模型在特定 领域的理解力

### 在经济学领域 应用的具体案 例

### 基于大语言模 型的开源项目 分享

在实际应用中，为了提高 API 调用效率，可以采用多线程并发处理多个请求。多线程调用的优势包括：

- 提高响应速度
- 更高的处理量

具体代码示例，已保存在项目库中，可进一步查看。

**注意：**由于大部分平台都有访问流量上限，使用多线程时需要特别注意线程数量不能超过限制。

# 实际应用：使用大语言模型提取 上海市历年经济增长目标

- ① 遍历文件夹，将政府工作报告文本、年份、城市名，整理成规范表格
- ② 选择合适模型，获取所需要模型的 API Key
- ③ 使用多线程方法，批量调用大语言模型处理文本
- ④ 等待结果返回

- 最终结果：由于工作报告文本量较大，处理 20 个样本最终耗时 2 分钟，计费 0.1 元
- 处理结果展示

城市		年份		文本内容		economic_growth_goal	
缺少:	0 (0%)	缺少:	0 (0%)	缺少:	0 (0%)	缺少:	0 (0%)
非重复:	1 (5%)	非重复:	20 (100%)	非重复:	20 (100%)	非重复:	13 (65%)
上海市		100%		20 非重复值		6.5% 7.5% 8% 其他 20% 10% 10% 60%	
0	上海市	2016		上海市市长杨雄 各位代表:	现在	6.5%-7%	
1	上海市	2017		上海市市长杨雄 各位代表:	现在	6.5%	
2	上海市	2015		——2015年1月25日在上海市第十四届人民代表大会第一次会议上			
3	上海市	2014		各位代表:	现在, 我代表上海市	7.5%	
4	上海市	2010		政府工作报告 ——2010年1月26日在上海市人民代表大会上		8%以上	
5	上海市	2004		韩正市长在上海市第十二届人民代表大会第一次会议上		10%以上	
6	上海市	2005		政府工作报告 ——2005年1月22日在上海市人民代表大会上		11%	
7	上海市	2011		政府工作报告 ——2011年1月16日在上海市人民代表大会上		8%	
8	上海市	2007		政府工作报告 ——2007年1月18日在上海市人民代表大会上		9%	
9	上海市	2013		各位代表:	现在, 我代表上海市	7.5%	
10	上海市	2012		政府工作报告 ——2012年1月11日在上海市人民代表大会上		8%	
11	上海市	2006		关于上海市国民经济和社会发展第十一个五年规划纲要		10%	
12	上海市	2023		韩正市长在上海市第十六届人民代表大会第一次会议上		5.5%	
13	上海市	2022		——2022年1月11日在区第六届人民代表大会第一次会议上		高于全市平均水平	
14	上海市	2008		政府工作报告 ——2008年1月24日在上海市人民代表大会上		10%	
15	上海市	2020		上海市市长 应勇 各位代表:		6.5%	
16	上海市	2021		2021政府工作报告 各位代表:		6%	
17	上海市	2009		政府工作报告 ——2009年1月14日在上海市人民代表大会上		9%	
18	上海市	2019		上海市市长 应勇 各位代表:		6.5%	
19	上海市	2018		上海市市长 应勇 各位代表:		6.5%	

- 2022 年工作报告原文：“2022 年全区经济社会发展的主要预期目标是：地区生产总值增速高于全市平均水平，区级一般公共预算收入增长 8%”

## ① 引言

## ② 大语言模型的代码调用

## ③ 如何强化大语言模型在特定领域的理解力

Prompt Engineering (提示词工程)

RAG (检索增强生成)

Fine-Tuning (微调)

## ④ 在经济学领域应用的具体案例

## ⑤ 基于大语言模型的开源项目分享

Chatbox & CherryStudio

PDF2zh

Ollama

其他

## 引言

### 大语言模型的 代码调用

### 如何强化大语言模型在特定领域的理解力

Prompt Engineering  
(提示词工程)

RAG (检索增强生成)

Fine-Tuning (微调)

### 在经济学领域 应用的具体案例

### 基于大语言模型 的开源项目 分享

- 可靠性是学术研究的关键，在实际论文中使用大语言模型构造 X、Y，仍有两个问题还需解决：
- 如何使大语言模型掌握处理特定文本数据的能力？
- 如何证明大语言模型确实掌握了这个能力？
- 解决方案：
  - ① 设计合理提示词 (Prompt)
  - ② 构造知识库 (RAG)
  - ③ 对大语言模型进行微调 (Fine-Tuning)

# Prompt 工程

- 什么是 **Prompt** 如果把大语言模型比作工人，Prompt 就相当于指导工人的号令。
- 为什么还需要 **prompt**? 虽然目前最先进大语言模型已内置思维链，使用时无需复杂提示，但经济学领域的任务多为文本分类和语义识别。此时，采用思维链模型成本较高且优势不明显，因此应选择效果好且经济的方案。
- 什么是好的 **prompt**? 好的 prompt 应具备以下特点：
  - ① 明确任务目标。
  - ② 提供清晰的输入。
  - ③ 明确期望的输出格式。
  - ④ 提供思维链。
  - ⑤ 指令保证格式清晰

## 小贴士：

- 众多 API 平台都提供了 prompt 生成功能，可以先在对应平台生成 prompt 再进行修改。

# Prompt 示例

```
# 时间间隔判断任务
## 任务描述
请编写一个Prompt，用于判断两个时间点之间的小时数。需要确保格式清晰、用词准确，使模型能够准确理解并执行任务。
## 输出要求
1. **角色设定**:
    - 你是一位专门处理时间计算的智能助手，擅长时间管理和日历计算。
    - 你的首要目标是准确计算时间间隔，以小时为单位。
2. **具体要求**:
    - **应该做**:
        - 计算两个时间点之间的总小时数。
        - 提供详细的计算过程。
        - 输出结果需以用户友好的方式呈现。
    - **不应该做**:
        - 不要忽略任何时间单位的转换。
        - 不要提供与时间间隔无关的信息。
3. **参考信息依赖**:
    - 严格根据用户输入的两个时间点进行计算。
## 示例
**示例输入**:
- 开始时间: 2023年10月1日 08:00:00
- 结束时间: 2023年10月3日 10:30:45
**示例输出**:
### 小时数计算
- 总小时数: 50小时
### 计算过程
1. 从2023年10月1日 08:00到2023年10月2日 08:00，共计24小时。
2. 从2023年10月2日 08:00到2023年10月3日 08:00，再加24小时。
3. 从2023年10月3日 08:00到2023年10月3日 10:30，共计2小时。
4. 综合计算: 24小时 + 24小时 + 2小时 = 50小时。
## 思维链引导
请按照以下步骤进行思考和计算:
1. **第一步**: 根据输入的开始和结束时间，识别时间间隔。
2. **第二步**: 将时间间隔转换为小时数，确保计算准确。
3. **第三步**: 综合计算总小时数。
4. **第四步**: 检查计算过程，确保无误后输出结果。
通过以上步骤，确保准确得出总小时数，并以清晰的格式呈现给用户。
### 输入
```



# 构建知识库

- 什么是 **RAG** 如果把大语言模型比作工人，知识库就相当于指导工人如何工作的手册，可供其翻阅。
- 目前主流 API 平台，如：阿里、coze，大多支持上传 pdf、word、txt 等格式的文档来构建知识库，在调用过程中，模型可从知识库中读取知识来提高回复质量。
- 与直接使用大语言模型相比，RAG 具有以下特点：
  - ① 成本更低：只收取上传一次时的费用，后续不再收费
  - ② 更高效：利用文档中的信息作为上下文，能提高生成回答的准确性和完整性
  - ③ 文本读取相对随机：“工人”是否会读取知识库，会受到知识库多大影响无法准确衡量。因此，知识库并不适用于严肃的应用场景。

## 如何强化大语言模型在特定领域的理解力

在经济学领域  
应用的具体案  
例

## 基于大语言模型的开源项目分享

知识库：由 10 篇关于企业绿色发展的论文组成。

**分析：**同一个模型，引入知识库后，回答更倾向于目前学术研究的关注方向，与不使用知识库的回答有较大区别。

# 通过微调强化特定领域能力

## 引言

### 大语言模型的 代码调用

### 如何强化大语 言模型在特定 领域的理解力

Prompt Engineering  
(提示词工程)

RAG (检索增强生成)

Fine-Tuning (微调)

### 在经济学领域 应用的具体案 例

### 基于大语言模 型的开源项目 分享

- **什么是微调?**: 如果把大语言模型比作工人, 微调就相当于使用实现准备好的针对性教材 (数据), 以上课的方式, 对大语言模型的思维模式进行改进, 以适应新工作。
- **为什么要微调?**: 通过微调一方面可以提升模型处理数据的能力, 另一方面也可以验证模型处理数据的可靠度。

## 微调步骤:

- ① 准备并上传训练数据
- ② 训练新的微调模型
- ③ 评估结果并根据需要返回步骤 1
- ④ 使用微调模型

## 如何准备微调数据:

- ① **数据标注**: 根据需要处理的目标任务, 针对性的提供一系列回答的结果示例。
- ② **数据格式化**: 将预处理后的数据转换为模型所需的格式。常见格式有 JSON、CSV 等。例如, 在基于文本分类的微调中, 由于要求回答相对单一, 可以采用如下格式:

```
{"prompt": "请你判断“今天股票又亏了三个点”  
属于乐观还是悲观?", "completion": "悲观"}
```

- ③ **数据扩增 (可选)**: 当可用数据量不足时, 可考虑使用数据扩增技术, 如同义句生成、数据混合或噪声注入等方法, 增加数据多样性, 提高模型鲁棒性 (稳定性)。

关于微调的具体介绍, 已保存在项目库中, 可进一步查看。

# 微调测试案例

## 测试案例一：

采购公告：怒江州林业局森林防火地理信息指挥系统采购项目中标公示

是否微调	是否需要 prompt	模型名称	分类结果
是	否	ERNIE 4.0	两者
否	是	ERNIE 4.0	(未能识别任务)
否	是	GPT-4o	两者
否	是	GPT-4o-mini	两者

## 结果发现：

微调后的 ERNIE 4.0 在识别准确性上大致接近于未经过微调的 GPT-4o-mini。

后续查询相关论坛帖子，发现如果只是对文本进行分类处理，只要选择足够优秀的大语言模型，微调并不必要。

## ① 引言

## ② 大语言模型的代码调用

## ③ 如何强化大语言模型在特定领域的理解力

Prompt Engineering（提示词工程）

RAG（检索增强生成）

Fine-Tuning（微调）

## ④ 在经济学领域应用的具体案例

## ⑤ 基于大语言模型的开源项目分享

Chatbox & CherryStudio

PDF2zh

Ollama

其他

# 案例一：大语言模型、文本情绪与金融市场（姜富伟等，2024）

## 主要研究内容：

- 构建中文金融大语言模型，用于预测金融市场情绪、资产价格回报等。
- 将结构化的市场数据（如股价、回报率）与非结构化文本数据（如财经新闻、公司公告）相结合，通过大模型更精准地刻画金融市场非理性情绪。

## 核心技术方案：

- 以 BERT/ERNIE 类深度学习模型为基础做预训练与微调。
- 引入市场回报标签，进行有监督的情绪分类或回归预测。
- 对比传统字典法，验证“大模型情绪”在极端市场状况下的预测优越性。

# 案例一：实现过程

- 思路：
  - 在通用预训练模型基础上，利用市场回报数据作为情绪标签进行有监督微调。
- 具体流程：
  - ① 选定预训练模型：如 BERT、ERNIE、RoBERTa 等。
  - ② 标注数据集：以股票收益正负为情绪标签（0/1 分类），或多档位标签。
  - ③ 模型评估：分类准确率、F1-score、在样本外的收益预测表现等。
  - ④ 模型部署：调用 API，将微调好的模型用于未来新闻文本情绪批量处理。
- 效果：
  - 明显优于传统字典法，捕捉到非线性、上下文关联和极端事件下情绪冲击。



## 案例二：企业数字化转型的测度 难题 (金星烨等，2024)

引言

大语言模型的  
代码调用

如何强化大语  
言模型在特定  
领域的理解力

在经济学领域  
应用的具体案  
例

基于大语言模  
型的开源项目  
分享

### 主要研究内容：

- 提出用大语言模型识别上市公司年报中是否真实应用数字技术（如 AI、大数据、云计算等），从而构建更准确的数字化转型指标。
- 解决传统词典法关键词遗漏和误判等问题，进而研究数字化转型对企业绩效的影响差异。

### 核心技术方案：

- 采用大语言模型（如 ERNIE）对年报句子逐句判定是否涉及数字技术以及数字技术类型。
- 与人工标注结合，“进一步预训练” + 微调，在构造新指标时更加完备且准确度更高。

## 案例二：实现过程

- **数据标注：**
  - 提前选定一个人工标注数据集（部分年报句子），区分是否真实采用数字技术。
- **微调过程：**
  - ① 在已有的中文模型（如 ERNIE）上进行“进一步预训练”（further pretraining），让模型适应企业年报用语。
  - ② 基于标注数据进行分类微调：句子 → 是否包含“真实使用数字技术”。
  - ③ 最终模型在测试集上评估指标，如 Accuracy、Precision、Recall、F1-score 等。
- **效果：**
  - 相较于关键词检索法，显著降低误判（谈技术却未实际应用）与漏判（实际应用技术却关键词缺失）的概率。

## ① 引言

## ② 大语言模型的代码调用

## ③ 如何强化大语言模型在特定领域的理解力

Prompt Engineering（提示词工程）

RAG（检索增强生成）

Fine-Tuning（微调）

## ④ 在经济学领域应用的具体案例

## ⑤ 基于大语言模型的开源项目分享

Chatbox & CherryStudio

PDF2zh

Ollama

其他

# Chatbox & CherryStudio：通过 API 实现各平台大语言模型集成

- **概述：**基于 API 调用机制，可以通过上传 API 的形式，实现各平台大语言模型的集成。
- **优势：**将 API 调用可视化实现类似于网页的功能；热门模型如 deepseek 网页端经常卡顿，但是使用 API 调用则不会。

# PDF2zh：将大语言模型引入 pdf 文献翻译

- 已有翻译引擎仅限于机翻，少部分可以使用 AI 的翻译软件并不能提供全文翻译
- 此项目可以将 PDF 全文翻译为中文，并保留排版格式，方便阅读。
- 由于项目自身基于命令行，且其可视化窗口配置较为复杂。基于此项目开发了可视化项目，也可供参考：

<https://github.com/luzhiyu-econ/PDF-Translator>

# Ollama: 本地开源大模型部署

## 引言

## 大语言模型的 代码调用

## 如何强化大语 言模型在特定 领域的理解力

## 在经济学领域 应用的具体案 例

## 基于大语言模 型的开源项目 分享

Chatbox &  
CherryStudio

PDF2zh

Ollama

其他

- 依据本地计算机算力，主要受电脑显卡显存于电脑内存限制，可以在本地部署不同的大语言模型模型。并开放 API 供用户调用。
- 通过本地调用模型一方面节省了 API 费用；另一方面，也规避了后续官方模型调整，原实验结果不可复现的局面。

## 引言

## 大语言模型的 代码调用

## 如何强化大语 言模型在特定 领域的理解力

## 在经济学领域 应用的具体案 例

## 基于大语言模 型的开源项目 分享

Chatbox &  
CherryStudio

PDF2zh

Ollama

其他

- Cursor, Trae: AI powered IDE
- 将 AI 功能集成到 IDE 中，可以实现代码生成、代码解释、代码调试等功能。

# References



Dell, M. (2023).  
Deep Learning for Economists.  
*Journal of Economic Literature.*



Korinek, A. (2023).  
Generative AI for Economic Research: Use Cases and Implications for  
Economists.  
*Journal of Economic Literature.*



Hong, Y. & Wang, S. (2024).  
ChatGPT 与大模型将对经济学研究范式产生什么影响?  
计量经济学报.