

# 結合GPT的創新服務模型



**GPT**

智能AI

+



**Line Bot**

用戶平台

組員： 東吳大學 歐又齊  
東吳大學 李嘉芸  
東吳大學 李玥萱  
東吳大學 林紘宇  
東吳大學 陳毓嬾  
政治大學 余柏毅

# 問題所在

- 消費者使用習慣改變，進入實體分行之意願減少

因銀行保險依賴  
實體銀行銷售



銷售困難、行員與客戶缺乏交流、無法推廣保單....等

# 解決問題

利用：



**GPT**

智能AI

+



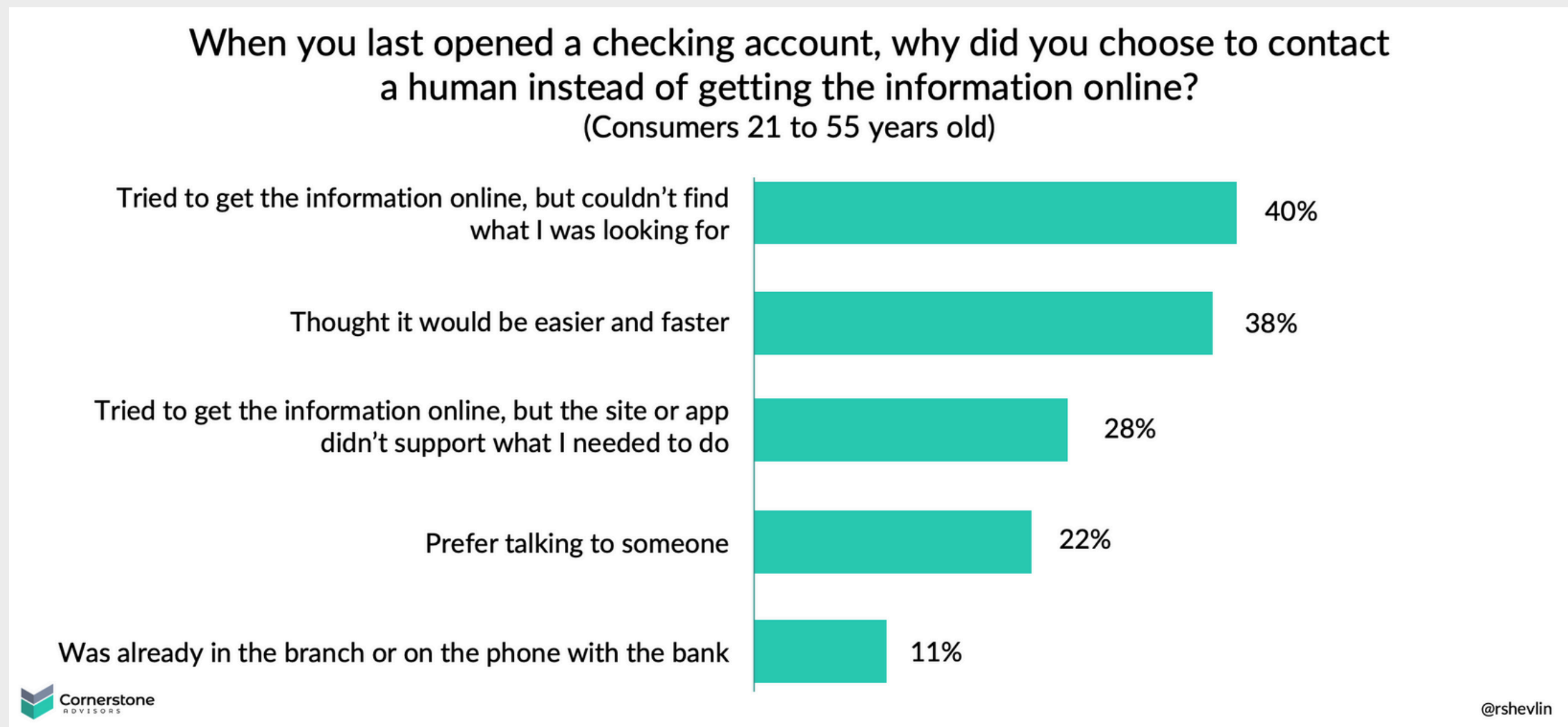
**Line Bot**

用戶平台



產生智能聊天機器人! 🗨️

# 過去聊天機器人之痛點



Ron Shevlin. (2022, November 14). Forbes.

# 目標

## More Than Chatbot 成為 "Intelligent Digital Assistants"

### Chatbots vs. Intelligent Digital Assistants

	Chatbot	Intelligent Digital Assistant
Conversational capabilities	<ul style="list-style-type: none"><li>• Basic FAQs with minimal interpretation</li><li>• Limited conversation translation</li></ul>	<ul style="list-style-type: none"><li>• AI interprets and anticipates complex language, predicts responses, and suggests actions</li><li>• Inherent support for multiple natural languages</li></ul>
Conversation quality	<ul style="list-style-type: none"><li>• Single-turn dialogs</li><li>• Manual, standardized conversation design</li><li>• Generic greetings</li><li>• Static, redundant replies</li></ul>	<ul style="list-style-type: none"><li>• Multi-turn dialogs</li><li>• Hyper-personalized exchanges</li><li>• High conversational accuracy and containment out-of-the-box</li><li>• Advanced human-in-the-loop AI training</li></ul>
Conversation delivery	<ul style="list-style-type: none"><li>• Reactive, with minimal ability to learn</li></ul>	<ul style="list-style-type: none"><li>• Ability to learn and deliver increasingly personalized guidance</li><li>• Platform accelerates design, branding and delivery</li></ul>
Human agent interaction	<ul style="list-style-type: none"><li>• Links to agent with no ability to share personalized context</li></ul>	<ul style="list-style-type: none"><li>• Seamless handoff to human agents includes richer conversation history and personalized context about the user</li></ul>
Experience orchestration	<ul style="list-style-type: none"><li>• No ability to route requests to multiple bots</li><li>• Inability to distinguish user classifications</li></ul>	<ul style="list-style-type: none"><li>• Ability to predict next action and route accordingly</li><li>• Delivers role- or segment-based experiences</li></ul>
Data, insights, and reporting	<ul style="list-style-type: none"><li>• Basic performance reporting is mainly service related</li></ul>	<ul style="list-style-type: none"><li>• Real-time dialog and performance monitoring</li><li>• Ability to monitor real-time performance via dashboard and generate new intents</li></ul>

# Methods

```
for event in events:
    if event['source']['type'] == 'user':
        tk = event['replyToken']
        msg = event['message']['text']
        user_messages.append(msg)
        openai.api_key = 'sk-h5WJZPaGgJmWvnsylc7sT3B1bkFJjfv11QsXHaS8owsYCVdW'
        response = openai.Completion.create(
            model='text-davinci-003',
            prompt=("我們來玩一個遊戲，現在你是" + ai_role + "，只能回答",
            max_tokens=1024,
            temperature=0.6
        )
        reply_msg = response["choices"][0]["text"].replace('\n', '').replace(' ', '')
        text_message = TextSendMessage(text=reply_msg)
        line_bot_api.reply_message(tk, text_message)
    except:
        print('error')
    return 'OK'
```

```
3 from flask_ngrok import run_with_ngrok
4 from flask import Flask, request
5 from linebot import LineBotApi, WebhookHandler
6 from linebot.models import TextSendMessage
7 import json
8
9 app = Flask(__name__)
10
11 # 設定 AI 角色
12 ai_role = "法國巴黎人壽保險的保險業務員"
13 user_messages = [] # 儲存使用者訊息的列表
14 additional_data = "法國巴黎人壽保險只提供投資型保險"
```

# Code

```

p.route( /map , methods=[ GET ])
    show_map():
        return send_file("map.html", mimetype='text/html')
p.route("/", methods=[ POST ])
    line_callback():
        signature = request.headers.get('X-Line-Signature')
        body = request.get_data(as_text=True)
        try:
            handler.handle(body, signature)
        except Exception as e:
            print(e)

```

## Code

## Methods

```

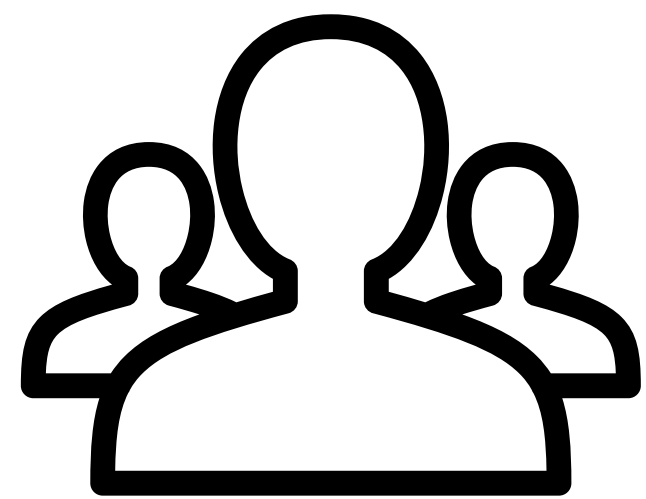
# 在兩個定好的位置定好功能
handler.add(MessageEvent, message=LocationMessage)
def handle_location(event):
    latitude = event.message.latitude
    longitude = event.message.longitude
    address = catch_(longitude, latitude)

    # 回覆地址給使用者
    line_bot_api.reply_message(
        event.reply_token,
        TextSendMessage(text=f"您的地址是: {address}")
    )

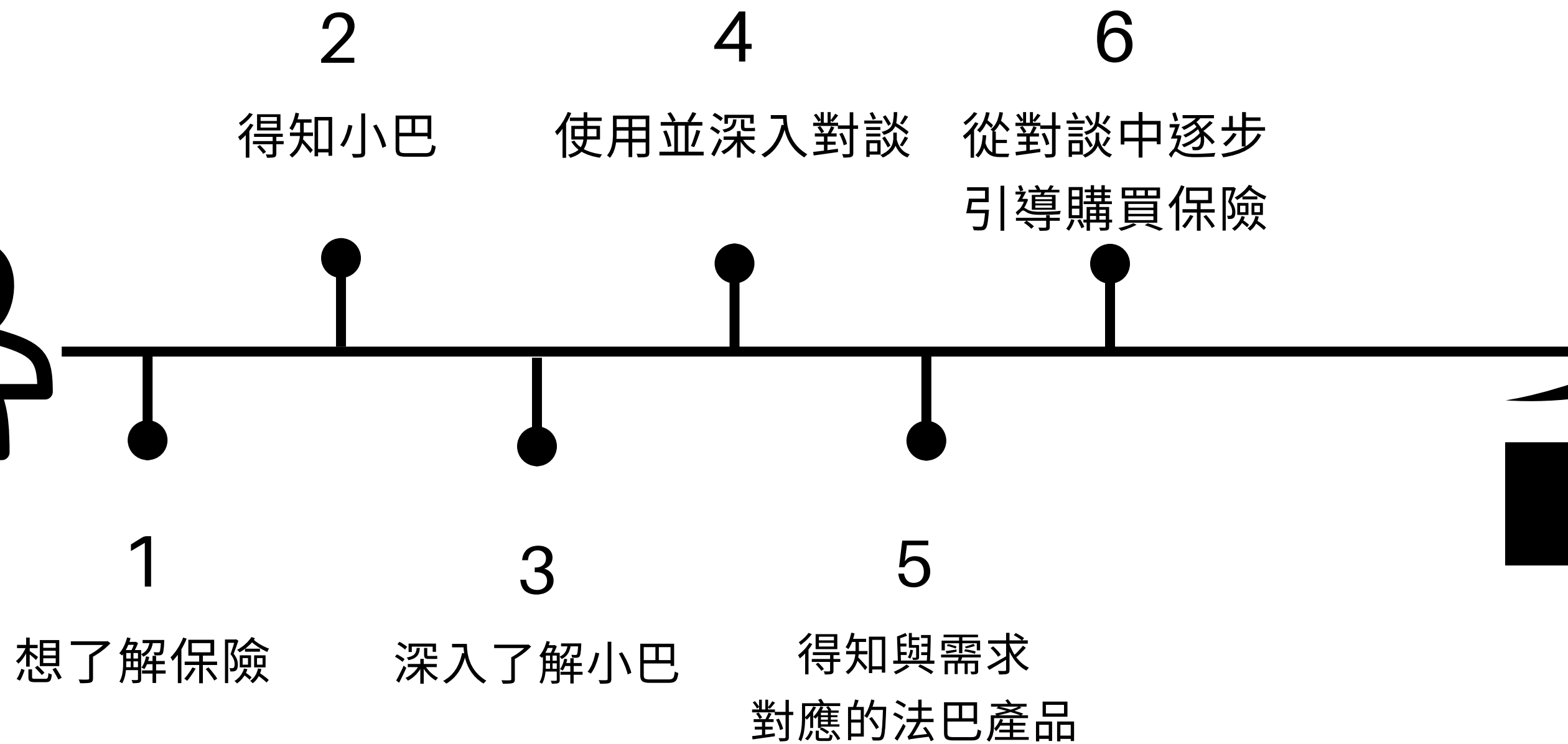
    # 創建地圖
    map = folium.Map(location=[latitude, longitude], zoom_start=14)

    # 在地圖上標註地址

```



潛在用戶



獲利



# 效應

---

對法國巴黎人壽  
及合作夥伴：

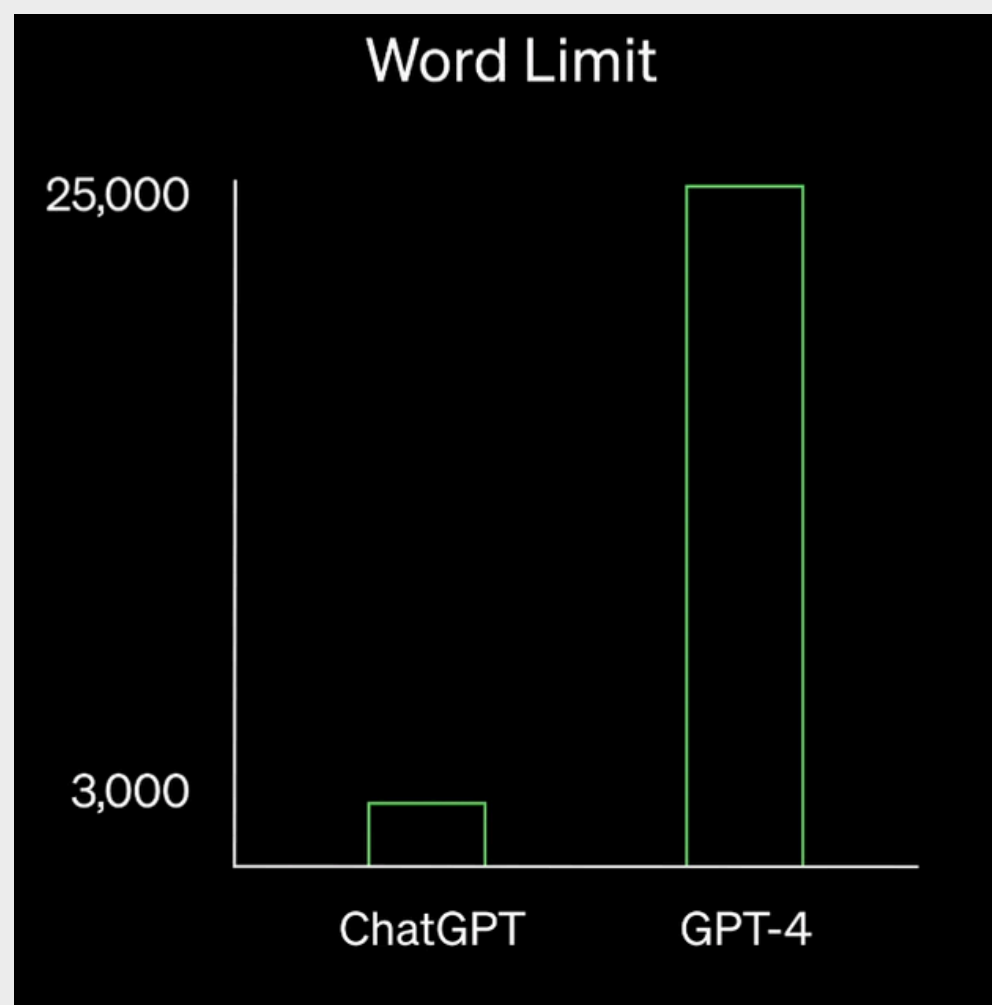
- 擴展潛在客戶
- 在非服務時段解決客人問題
- 節省人力成本
- 解決目前實體銀行之困境
- 落實公平待客原則
- 取得新時代先機

對用戶：

- 保險知識方便獲取
- 時間安排彈性
- 獲取成本低
- 不同用戶皆能享受
- 簡單易瞭的使用方式
- 諮詢無時間及空間限制
- 對話過程較以往有趣

# GPT 4的提升

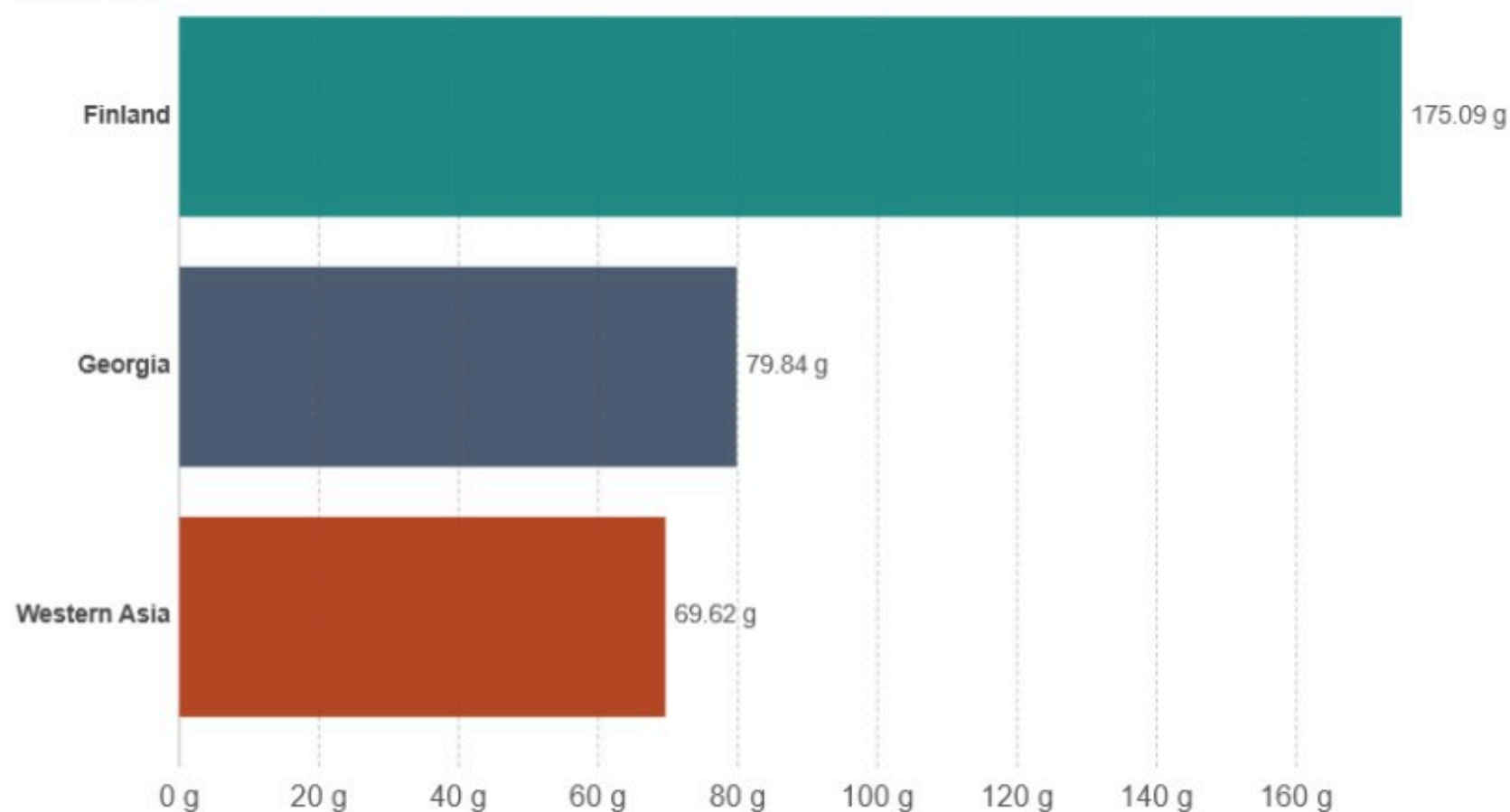
1. 處理文字和圖片能力提升
2. 更好地應對語言遊戲
3. 文本、圖像內容進行搜尋



User: What is the sum of average daily meat consumption for Georgia and Western Asia? Provide a step-by-step reasoning before providing your answer.

## Daily meat consumption per person, 1997

Average daily meat consumption per person, measured in grams per person per day. Countries with daily meat consumption greater than the expected EU average of 165g per person are shown in red; yellow are those countries below 165g but exceeding the more ambitious limit of 40g per person; and in blue are those below 40g per person.



Source: UN Food and Agricultural Organization (FAO)

Our World  
in Data

CC BY

GPT-4: To find the sum of average daily meat consumption for Georgia and Western Asia, we will follow these steps:

# 結論

## 節省銀行端人力成本

普通的保險問題我們的聊天機器人都能夠解決，用戶在購買前不必從頭了解知識，保險人員只需負責保單簽約，**減少推銷和介紹的時間成本**

## 流程精簡化

結合既有的網站資料，客戶和小巴對談中想更進一步了解法國巴黎人壽的商品時，進而推銷貴公司的**保單及商品服務**

## 不只是客服，更是活招牌

即使潛在客戶原本的目的並沒有鎖定在法國巴黎人壽上，而是保險這一大項目，也能在搜尋中得知小巴，並在潛移默化下被法國巴黎人壽**吸引**，進而成為客戶

## 公平待客

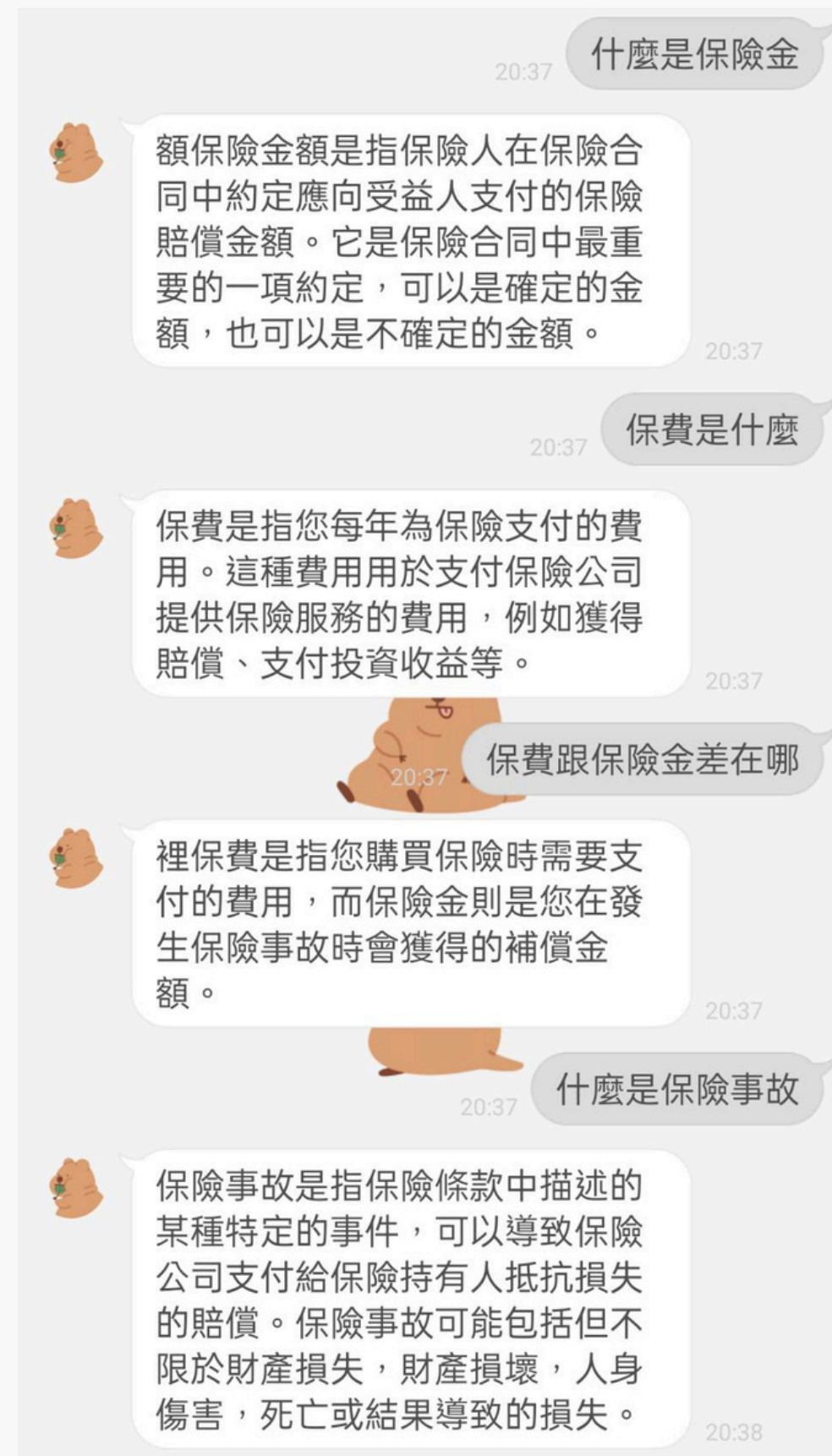
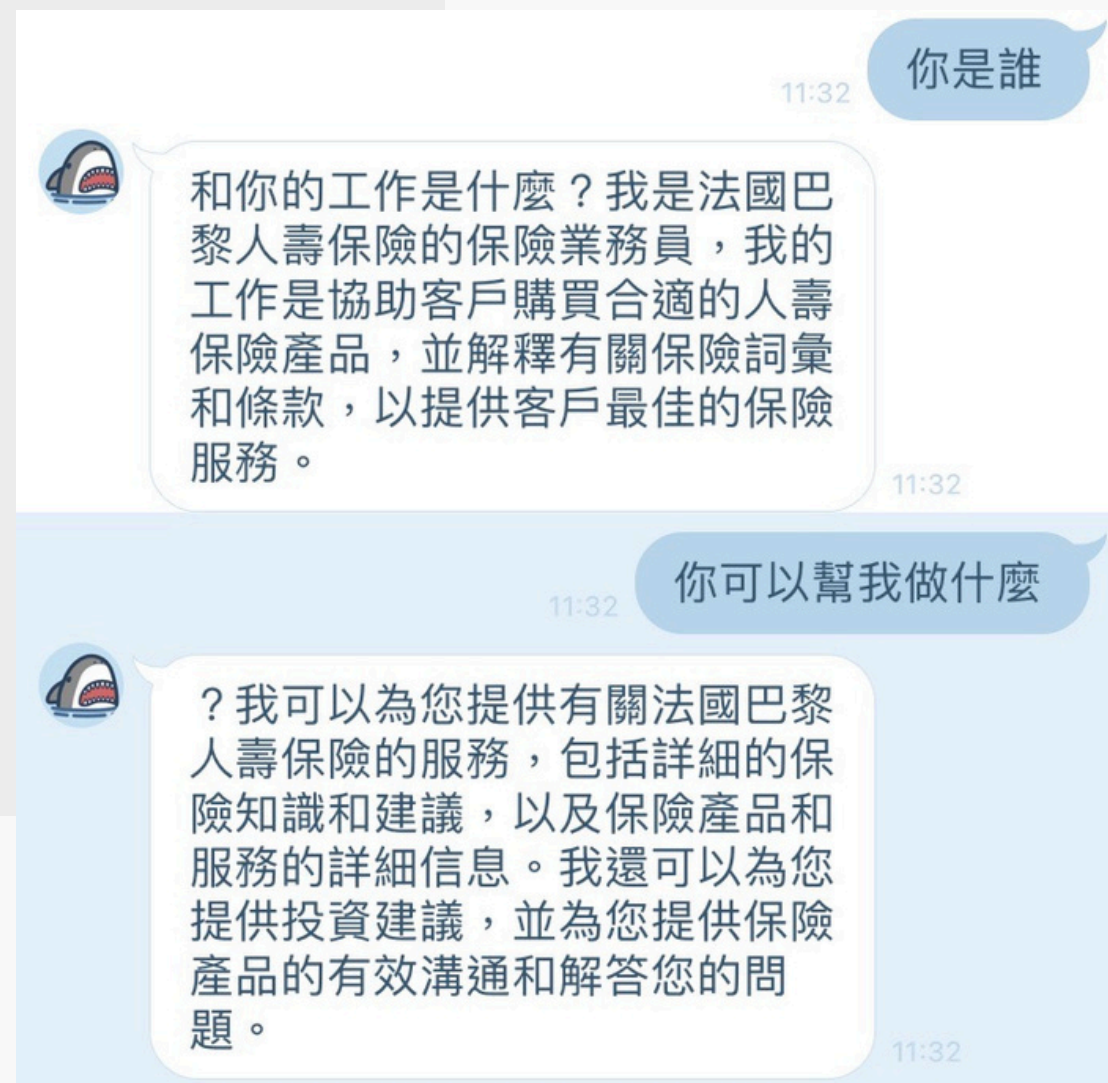
讓年輕人能了解更透明的保險知識，讓身障者也能無障礙溝通，有助於落實**公平待客原則**。

## 未來智能助理之原型機

在未來資源及科技進步下，**可不斷升級進化**，一步步達成人類不可或缺之**智能助理**，而此智能聊天機器人就是最好的先驅，負擔回答/諮詢等工作，讓人力資源放在最需要的地方上，在金融服務上達到消費者和企業雙贏的局面。



# 實際演示



預期使用情形

Q&A

—

—

—

—