LLM Sourcing Agent

Satyam Agarwal

March 15, 2024

Abstract

This research paper explores the use of AI technologies and novel techniques to improve the efficiency and effectiveness of searching for and analyzing startups in various business sectors. The paper focuses on the utilization of language models and API interactions to gather information from different sources, including search engines and Crunchbase, a popular database of startups. Through the implementation of OpenAI's LLM agents and integration with Bing and Crunchbase APIs, we demonstrate how AI can streamline the process of finding and extracting relevant information about startups, their founders, funding details, and other valuable insights.

1 Introduction

Startups play a pivotal role in driving innovation and economic growth in various industries. However, identifying and analyzing startups can be a challenging and time-consuming task. Traditional methods of manually searching the internet, attending conferences and relying on local connections, often prove to be inefficient, requiring significant human effort and expertise. To address this challenge, AI-powered techniques offer a promising solution by automating and augmenting the search and analysis process. In this paper, we present a comprehensive approach that leverages AI technologies, including language models and API interactions, to enhance the efficiency and accuracy of startup search and analysis.

2 Methodology

The proposed methodology integrates several key components, including OpenAI's language models, Bing API, and Crunchbase API, to enable efficient search and analysis of startups. The process involves the following steps:

2.1 Data Collection and Enrichment

To initiate the search process, we utilize OpenAI's LLM (Language Model) agents to generate an efficient search query based on the desired business sector. The LLM agent interacts with the user to gather specific requirements and generate a precise search query for optimal results. The search query is enriched using contextual information obtained from the LLM agents, ensuring the inclusion of relevant keywords and potential websites related to the business sector. These websites can be news platforms, journalism sites, or other information sources that report on or present information about startups in the desired sector.

2.2 Searching through Bing API

The enriched search query is then passed to the Bing API, which searches through the web to retrieve a list of relevant websites. The Bing API is configured with the provided API key, ensuring secure and authorized access to the search API. The API response provides a list of URLs corresponding to the identified websites. It is important to note that the search excludes Crunchbase.com to avoid duplicating results.

2.3 Web Scraping for Company Names

We employ web scraping techniques to extract company names from the identified websites. The websites' HTML content is parsed using the BeautifulSoup library, and relevant tags such as `<h1>`, `<h2>`, and `<h3>` are searched to extract the company names. The extracted company names provide valuable data for further analysis.

2.4 Crunchbase API Integration

Using the Bing API search results, we proceed to find the Crunchbase URLs corresponding to the extracted company names. The Crunchbase API is invoked using the obtained company names to fetch the corresponding Crunchbase URLs. The integration with the Crunchbase API streamlines the process of retrieving detailed information about the startups.

2.5 Gathering Information from Crunchbase

The obtained Crunchbase URLs are used to interact with the Crunchbase API and retrieve useful information about the startups. The API's endpoint for organizations is accessed, providing the organization ID for each startup. The API response contains a wealth of information, including the company's website, LinkedIn profile, short description, funding details, founder information, and more.

2.6 Data Presentation and Summary Generation

The obtained information from the Crunchbase API is formatted and presented in a human-readable form. This includes displaying the company name, website, description, LinkedIn profile, funding details, and founder information. OpenAI's LLM agents are used to generate concise summaries of the obtained data, ensuring the retention of key information and generating a comprehensive overview of the startup.

3 Results and Discussion

The results obtained from the implemented methodology demonstrate significant improvements in the efficiency of startup search and analysis. By leveraging AI technologies, we were able to streamline the process of finding relevant websites, extracting company names, and retrieving detailed information from Crunchbase. The use of LLM agents facilitated the generation of precise search queries and concise summaries, reducing the time and effort required for manual analysis.

The integration with the Bing API enhanced the search process by leveraging Bing's search algorithms and extensive web index. This ensured reliable and relevant search results, further augmenting the accuracy of the startup identification process. The web scraping technique for extracting company names from websites proved effective, enabling the gathering of rich datasets for analysis.

Integration with the Crunchbase API served as a valuable resource for retrieving detailed company information. The API provided access to crucial data such as website, LinkedIn profile, short description, funding details, and founder information. The availability of comprehensive startup profiles streamlines the analysis process, enabling investors and venture capitalists to make informed decisions with greater confidence.

Additionally, OpenAI's LLM agents offered a versatile and powerful tool for generating concise and informative summaries of the obtained data. By utilizing prompt engineering techniques, the LLM agents were configured to provide maximum information within limited tokens, ensuring the generated summaries were highly informative.

4 Conclusion

This research paper presented a comprehensive approach for leveraging AI technologies to enhance the search and analysis of startups. By integrating OpenAI's LLM agents, Bing API, and Crunchbase API, we demonstrated how the process of identifying and analyzing startups can be streamlined, saving valuable time and effort. The use of LLM agents facilitated the generation of efficient search queries and concise summaries, allowing researchers, investors, and venture capitalists to quickly gather insights about startups in specific business sectors.

The results obtained from the implemented methodology showcased the effectiveness of AI technologies in improving the efficiency and accuracy of startup search and analysis. The integration with the Bing API and web scraping techniques enabled the identification of relevant websites and extraction of company names, while the Crunchbase API provided detailed information about the startups. By presenting the gathered information in a human-readable form and generating summaries, the methodology offered valuable insights for decision-making in venture capital and investment scenarios.

As AI technologies continue to advance, further refinements and enhancements can be made to improve the overall efficiency and usability of startup search and analysis. Future research can explore the integration of additional data sources, such as social media platforms and industry-specific databases, to gather a more comprehensive view of startups. Furthermore, the utilization of advanced AI techniques, including natural language understanding and sentiment analysis, can provide deeper insights into the potential viability and growth prospects of startups.

In conclusion, the utilization of AI technologies such as LLM agents, API interactions, and novel techniques offers immense potential for the venture capital industry and startups' analysis. The presented methodology showcases the power of AI in efficiently identifying and analyzing startups, empowering investors and venture capitalists with timely and accurate information for making informed decisions.

5 Code Snippets:

```python
from openai import OpenAI
import requests
from bs4 import BeautifulSoup

# Function to get the response from openai api with configurable models and tokens
def get_chat_response(prompt, pre_prompt="", model="gpt-3.5-turbo", tokens=500):
    response = client.chat.completions.create(
        model=model,
        messages=[{"role": "system",
                   "content": "You are a research assistant application whose purpose is to provide results for queries to the user. You only provide the answers without any explanations and introductions. For all the questions asked, return only the required values without other text."},
                  {"role": "user", "content": f"{pre_prompt} {prompt}"}],
        max_tokens=tokens
    )
    return response.choices[0].message.content
```

Listing 1: OpenAI Api integration

```python
def get_company_info(url):
    # Extract organization ID from the URL
    # Extracting permalink from crunchbase company url
    # The number '40' is the length of the string 'www.crunchbase.com/organization'
    organization_id = url[40:].split('/')[0]
    API_KEY = crunchbase_api_key
    # Base URL for the Entity Lookup API
    base_url = f"https://api.crunchbase.com/api/v4/entities/organizations/{organization_id}?card_ids=founders,fields&field_ids=website,linkedin,short_description&user_key={API_KEY}"

    try:
        # Send GET request with parameters
        response = requests.get(base_url)
        company_name = founder_name = founder_description = linkedin_url = website = description = funding_amt = funding_type = None
        # Check for successful response
        if response.status_code == 200:
            # Extracting information from crunchbase api response. Fully customisable to extract relevant information
            data = response.json()
            try:
                company_name = data["properties"]["identifier"]["value"]
                website = data["properties"]["website"]["value"]
                linkedin_url = data["properties"]["linkedin"]["value"]
                description = data["properties"]["short_description"]
                funding_amt = data['cards']['fields']['funding_total']['value_usd']
                funding_type = data['cards']['fields']['last_equity_funding_type']
                founder_name = data['cards']['founders'][0]['identifier']['value']
                founder_description = data['cards']['founders'][0]['description']
            except:
                pass

            return company_name, website, description, linkedin_url, founder_name, founder_description, funding_amt, funding_type
    except requests.exceptions.RequestException as e:
        print("Error:", e)

    return None
```

Listing 2: Extracting Information using crunchbase api

```python
print("Summary of results: ")
summary = get_chat_response(responses, "Given is a python dictionary containing information about various startup companies. You are an experienced venture capitalist who has invested in many successful startups and know very well what distinguishes a successful, unicorn startup from its mediocre counterpart at an early stage. Examine the company details one by one and evaluate how successful you think the company can be. As a venture capitalist, you prioritise smaller companies with innovative ideas with well-educated founders that are visionaries in their field. Assign a numerical ranking to the companies according to your evaluation and return this ranked list with an up to 10 word description of why you think the company deserves this rank. Format:
1.) <Company 1> - <reason why it is better than others>
2.) <Company 2> - <reason>
... and so on.
Up to 10 companies max but use your discretion to remove companies that you feel do not deserve to be on the list. The dictionary is below: ", model="gpt-4-turbo-preview")
print(summary)
```

Listing 3: Generating Summary o fdata