# LLMs_Hypothetical_Answers_ReRank_Financial_news

July 22, 2024

**How to use Query transformation and Hypothetical answering to re-rank retrieved articles and enhance the performance of your RAG pipeline?**

Hanane DUPOUY LinkedIn: https://www.linkedin.com/in/hanane-d-algo-trader/

Ever wondered how to accurately answer the question, `'What impact did the global outage of CrowdStrike, extensively used by Microsoft, have on Microsoft's stock price?'`

To tackle this, we'll employ the **hypothetical answer re-ranking** technique:

-**GOAL**: I'll evaluate whether the **hypothetical answer** can improve the re-ranking and the retrieved context, and subsequently enhance the LLM's response within our RAG pipeline. Alternatively, we will assess if the original query alone is sufficient to retrieve the appropriate context and deliver accurate results.

- Different techniques will be employed: **Query Transformation, Hypothetical Answers, Embeddings, and Similarity Scoring** to retrieve the relevant context from news articles fetched from the NEWS API.
- In these techniques, we will compare the **capabilities of three LLMs**: **gpt-4o-mini** (the latest small model from OpenAI), **gpt-4o** (the most capable LLM from OpenAI), and **gpt-3.5-turbo**."
- I'll use use **3 evaluation metrics** from **deepEval** for RAG pipelines: **Faithfulness, Context Relevancy and Answer relevancy**. These metrics are explained in the notebook.
- We will compare the three LLMs using two techniques: **Hypothetical Answer Re-ranking vs. Original Query Retrieval**.

**Steps:**

For each of the three LLMs: **GPT-4o-mini, GPT-4o, and GPT-3.5-turbo**:

**1-** We will perform **Search Queries** (or **Query transformation**) using the LLM to generate various formulation with the same keywords from the original user query.

**2-** We will use an LLM to generate a **hypothetical answer**. This creative response will serve as a potential answer, using placeholders instead of actual facts.

**3-** Based on each query from the search queries (1-), we will retrieve news article from NEWS API.

**4-** We will **embedd** user query, hypothetical answer and the collected articles

**5-** We compute the **similarity score** between 2 sets:

5-1- Hypothetical answer (2-) vs retrieved context (3-)

5-1- Original query (2-) vs retrieved context (3-)

**6-** Ask the LLM to give the final answer based on the user query and the retrieved context

**7-** Use 3 evalutaions metrics from DeepVal to evaluate the RAG pipeline: **Faithfulness, Context Relevancy and Answer relevancy.**

**8-** Key Takeways

# 1 Install Lib

[ ]:
```
import json
```

[ ]:
```
!pip install openai
from google.colab import userdata
openai_api_key = userdata.get('OPENAI_API_KEY')
news_api_key = userdata.get('NEWS_API_KEY')

from openai import OpenAI
client = OpenAI(api_key=openai_api_key)
```

Chat method OpenAI

[ ]:
```
def get_completion_gpt(input, gpt_model = "gpt-3.5-turbo"):
  completion = client.chat.completions.create(
        model=gpt_model,
        messages=[
            {"role": "system", "content": "Output only valid JSON"},
            {"role": "user", "content": input},
        ],
        response_format={ "type": "json_object" }
    )

  text = completion.choices[0].message.content
  parsed = json.loads(text)
  return parsed
```

# 2 Search NEWS API

[ ]:
```
import requests

def search_news(query, news_api_key= news_api_key,num_articles=5, from_datetime␣
 ↪= "2024-07-18",to_datetime = "2024-07-21"):
    response = requests.get(
        "https://newsapi.org/v2/everything",
        params={
            "q": query,
            "apiKey": news_api_key,
            "pageSize": num_articles,
            "sortBy": "relevancy",
            "from": from_datetime,
            "to": to_datetime,
        },
    )

    return response.json()
```

## 3 Generate Search Queries

[ ]:

```
user_query = "What impact did the global outage of CrowdStrike, which is used extensively by Microsoft, have on Microsoft's stock price?"

input = f"""
You have access to a NEWS API that returns recent news articles related to the user's question.

1. Make a list of search queries that match the topic described in the user's question.
2. Use different keywords related to the topic to create a variety of queries, making some general and others more specific.
3. Be imaginative and generate as many queries as possible. More queries will help you find better results.
4. Pick 10 of these queries.
For example, you can include queries like ['keyword_1 keyword_2', 'keyword_1', 'keyword_2'].

# User question: {user_query}

# Format: {{"queries": ["query_1", "query_2", "query_3"]}}
"""



llms = ["gpt-3.5-turbo", "gpt-4o-mini", "gpt-4o"]

dict_questions = {}
for llm in llms:
  print(llm)
  parsed = get_completion_gpt(input, gpt_model = llm)
  dict_questions[llm] = parsed
```

```
gpt-3.5-turbo
gpt-4o-mini
gpt-4o
```

```
[ ]: dict_questions
```

```
[ ]: {'gpt-3.5-turbo': {'queries': ['CrowdStrike global outage impact on Microsoft
     stock price',
        'Microsoft stock price reaction to CrowdStrike global issue',
        'CrowdStrike downtime effects on Microsoft shares',
        'Microsoft stock performance after CrowdStrike service disruption',
        'CrowdStrike outage influence on Microsoft stock value',
        'Microsoft stock price correlation with CrowdStrike downtime',
        'CrowdStrike incident impact on Microsoft share value',
        'Microsoft stock response to CrowdStrike global service failure',
        'CrowdStrike outage consequences on Microsoft stock market',
        'Microsoft stock price fluctuation due to CrowdStrike global problem']},
      'gpt-4o-mini': {'queries': ['CrowdStrike outage Microsoft stock impact',
        'global outage CrowdStrike Microsoft',
        'Microsoft stock price reaction CrowdStrike incident',
        'CrowdStrike service disruption effects on Microsoft',
        'impact of cybersecurity outages on stock prices',
        'Microsoft financial performance CrowdStrike downtime',
        'CrowdStrike incident analysis Microsoft equities',
        'how CrowdStrike affects Microsoft stock valuation',
        'effects of CrowdStrike on investor confidence in Microsoft',
        'cybersecurity outages and stock market trends']},
      'gpt-4o': {'queries': ['CrowdStrike global outage impact on Microsoft',
        'Microsoft stock price after CrowdStrike outage',
        'CrowdStrike outage effect on Microsoft',
        'CrowdStrike downtime Microsoft stock',
        'Microsoft shares after CrowdStrike issues',
        'Global outage of CrowdStrike affecting Microsoft',
        'CrowdStrike Microsoft stock market reaction',
        'Impact of CrowdStrike outage on MSFT stock',
        'How CrowdStrike outage influenced Microsoft',
        'CrowdStrike problems Microsoft financial impact']}}
```

## 4 Create a Hypothetical answer

```
[ ]: hypoth_answer = f"""
     Make up an answer to the user's question. We'll use this fabricated answer to␣
       ↪sort the search results.
     Imagine you have all the details to answer, but don't use real facts. Do not␣
       ↪give any numbers.
     Instead, use placeholders like 'EVENT affected something,' 'NAME mentioned␣
       ↪something on DATE,' or 'EVENT has caused something.'

     User question: {user_query}

```

```
Format: {{"hypotheticalAnswer": "hypothetical answer text"}}
"""

print(hypoth_answer)
```

```
Make up an answer to the user's question. We'll use this fabricated answer to
sort the search results.
Imagine you have all the details to answer, but don't use real facts. Do not
give any numbers.
Instead, use placeholders like 'EVENT affected something,' 'NAME mentioned
something on DATE,' or 'EVENT has caused something.'

User question: What impact did the global outage of CrowdStrike, which is used
extensively by Microsoft, have on Microsoft's stock price?

Format: {"hypotheticalAnswer": "hypothetical answer text"}
```

[ ]:

```
#Trying differenet llms:

hypoth_answer_llms = {}
for llm in llms:
  # print(llm)
  parsed_hypothet_answer = get_completion_gpt(hypoth_answer, gpt_model = llm)
  hypoth_answer_llms[llm] = parsed_hypothet_answer['hypotheticalAnswer']
  print(f"{llm}\n {hypoth_answer_llms[llm]}")
```

```
gpt-3.5-turbo
gpt-3.5-turbo
 The global outage of CrowdStrike has caused a temporary dip in Microsoft's
stock price as investors reacted to the uncertainty surrounding the
cybersecurity risks. However, Microsoft's resilient business model helped to
recover the losses in the following days.
gpt-4o-mini
gpt-4o-mini
 The global outage of CrowdStrike, which is extensively used by Microsoft, led
to increased investor concerns over cybersecurity vulnerabilities, causing a
temporary decline in Microsoft's stock price. Analysts noted that the EVENT
raised questions about the reliability of third-party security services, and
NAME mentioned something about potential risks on DATE. This situation has
caused a ripple effect in the tech market, affecting investor confidence in
companies reliant on CrowdStrike.
gpt-4o
gpt-4o
 The global outage of CrowdStrike, which is used extensively by Microsoft,
caused immediate concerns among investors regarding the cybersecurity resilience
of Microsoft's services. This EVENT brought about uncertainty in the market,
leading to a brief downturn in Microsoft’s stock price. Analysts cited fears
over potential vulnerabilities and service disruptions as primary reasons for
the stock's volatility during this period. However, NAME mentioned on DATE that
the company is taking necessary steps to mitigate any long-term impacts, which
has since helped stabilize the stock price.
```

## 5 Fetch news articles from NEWS API for each query:

[ ]:

```python
def get_articles_from_news_api(queries):
  articles = []
  for query in queries:
    result = search_news(query)
    if result['status'] == 'ok':
      articles = articles + result['articles']
    else:
      raise Exception(result["message"])
  return articles
```

[ ]:

```python
articles={}
for llm in llms:
  queries = dict_questions[llm]['queries']
  queries.append(user_query)
  articles[llm] =  get_articles_from_news_api(queries)
  if articles[llm]!=None:
    articles[llm] = list({article["url"]: article for article in articles[llm]}.
  ↪values())
```

[ ]:

```python
for llm in llms:
  print(len(articles[llm]))
```

```
26
23
40
```

[ ]:

```python
# #To save data locally
# for llm in llms:
#   pd.DataFrame(articles[llm]).to_csv("articles_"+llm+".csv")
```

[ ]:

```python
#Display some articles:
print("Total number of articles:", len(articles)) #3 LLM ==> 3 set of articles
llm = llms[-1]
for article in articles[llm][0:5]:
    print("Title:", article["title"])
    print("Url:", article["url"])
    print("Description:", article["description"])
    print("Content:", article["content"][0:300] + "...")
    print()
```

```
Total number of articles: 3
Title: The Global CrowdStrike Outage Triggered a Surprise Return to Cash
Url: https://www.wired.com/story/microsoft-crowdstrike-outage-cash/
Description: The event caused chaos at airports, grocery stores, and Starbucks
outlets.
Content: On Friday, when a CrowdStrike update caused millions of Microsoft
systems to crash around the world, many businesses were faced with a choice: Go
cash-only, or close until systems came back online.
… [+2941 chars]…

Title: Huge Microsoft Outage, Linked to CrowdStrike, Takes Down Computers Around
the World
Url: https://www.wired.com/story/microsoft-windows-outage-crowdstrike-global-it-
probems/
Description: A software update from cybersecurity company Crowdstrike appears to
have inadvertently disrupted IT systems globally.
Content: Banks, airports, TV stations, hotels, and countless other businesses
are all facing widespread IT outages, leaving flights grounded and causing
widespread disruption, after Windows machines have disp… [+1941 chars]…

Title: Chaos Reigns as Global Outage Breaks Everything from Airlines to
Emergency Services
Url: https://gizmodo.com/chaos-reigns-as-global-outage-breaks-everything-from-
airlines-to-emergency-services-2000476734
Description: Microsoft confirmed there’s an ongoing outage connected to IT
company CloudStrike. There's a patch, though we're still feeling the impacts.
Content: The whole world was thrown for a loop Friday after Microsoft confirmed
theres a humongous, ongoing outage connected to the IT security company
CrowdStrike. While the makers of Windows said the underl… [+2929 chars]…

Title: How One Bad CrowdStrike Update Crashed the World’s Computers
Url: https://www.wired.com/story/crowdstrike-outage-update-windows/
Description: A defective CrowdStrike kernel driver sent computers around the
globe into a reboot death spiral, taking down air travel, hospitals, banks, and
more with it. Here’s how that’s possible.
Content: That deeper access also introduces a far higher possibility that
security softwareand updates to that softwarewill crash the whole system, says
Matthieu Suiche, head of detection engineering at the s… [+3010 chars]…

Title: Global services slowly recovering after bug causes IT chaos
Url: https://www.bbc.com/news/articles/cg3m4jgdprxo
Description: The incident has sparked concern over the vulnerability of the
world's interconnected technologies.
Content: By Robert Greenall, BBC News
The outage has caused major delays at airports around the world
Businesses and services around the world are slowly recovering after a massive
IT outage affected comput… [+3598 chars]…
```

# 6 Embeddings and cosine similarity

## 6.1 Methods

[ ]:

```
def get_embeddings(input):
    response = client.embeddings.create(model="text-embedding-ada-002",␣
 ↪input=input)
    return [data.embedding for data in response.data]
```

[ ]:

```
def get_embeddings_articles(articles):
  articles_prepare_embedd =  [
        f"{article['title']} {article['description']} {article['content'][0:
 ↪700]}"
        for article in articles
    ]

  print(f"Length of articles to embed: {len(articles_prepare_embedd)}")
  article_embeddings = get_embeddings(articles_prepare_embedd)
  return article_embeddings
```

[ ]:

```
similarity_score_func=lambda x, y: 1 - spatial.distance.cosine(x, y)

def calculate_cosine_distance(embedding_hypoth, article_embeddings):

    cosine_similarities = []
    for article_embedding in article_embeddings:
        cosine_similarities.append(similarity_score_func(embedding_hypoth,␣
 ↪article_embedding))
    return cosine_similarities
```

[ ]:

```
def sort_articles_by_cosine_similarity(articles, cosine_similarities):
    scored_articles = zip(articles, cosine_similarities)
    sorted_articles = sorted(scored_articles, key=lambda x: x[1], reverse=True)
    print(f"Top 5 articles scores: {[score for _,score in sorted_articles[0:
 ↪5]]}\n")
    # for article, score in sorted_articles[0:5]:
    #     print("Title:", article["title"])
    #     # print("Url:", article["url"])
    #     # print("Date of publication:", article["publishedAt"])
    #     # print("Description:", article["description"])
    #     # print("Content:", article["content"][0:50] + "...")
    #     print("Score:", score)
    #     print()
    return sorted_articles
```

[ ]:

```
def context_retrieval(sorted_articles):
    """Get top 5 articles based on their similarity scores."""
    formatted_top_results =␣
  ↪[article["title"]+"\n"+article["description"]+"\n"+article["content"] for␣
  ↪article, _score in sorted_articles[0:5]]

    return formatted_top_results


def get_final_answer(user_query, formatted_top_results, llm):
  """Answer the user's question based on the retrieved context using a GPT␣
  ↪model: gpt-4o, gpt-4o-mini, gpt-3.5-turbo."""
  final_input = f"""
  Generate an answer to the user's question based on the given search results.
  TOP_RESULTS: {formatted_top_results}
  USER_QUESTION: {user_query}

  Include as much information as possible in the answer. Reference the relevant␣
  ↪search result urls as markdown links.
  """

  completion = client.chat.completions.create(
          model=llm,
          messages=[
              {"role": "user", "content": final_input},
          ],
      )

  return completion.choices[0].message.content
```

# 7 Example 1 LLM:

## 7.1 Similarities against the Hypothetical Answer

[ ]:

```
llm = llms[0]
embedding_hypoth = get_embeddings(hypoth_answer_llms[llm])[0]
article_embeddings = get_embeddings_articles(articles[llm]) #{list of embedded␣
  ↪articles , there are 26 articles}
cosine_similarities_hypoth = calculate_cosine_distance(embedding_hypoth,␣
  ↪article_embeddings)

print(f" len embedding vector={len(embedding_hypoth)}, len artciles␣
  ↪embedded={len(article_embeddings)}, len cosine_distance␣
  ↪vector={len(cosine_similarities_hypoth)}")
print(cosine_similarities_hypoth[:5])
print("\n")
```

```
sorted_articles_hypoth = sort_articles_by_cosine_similarity(articles[llm],␣
 ↪cosine_similarities_hypoth)
```

```
 len embedding vector=1536, len artciles embedded=26, len cosine_distance
vector=26
[0.8830353301233373, 0.880280016893984, 0.9041821516135816, 0.9036387715042302,
0.8692776540391578]


Top 5 articles scores: [0.9057614716323923, 0.905359099913487,
0.9041821516135816, 0.9036387715042302, 0.9022047694353023]
```

### 7.2 Similarities against the original query

[ ]:

```
llm = llms[0]
embedding_original_query = get_embeddings(user_query)[0]
# article_embeddings = get_embeddings_articles(articles[llm]) #already embedded␣
 ↪in the cell before
cosine_similarities_original=␣
 ↪calculate_cosine_distance(embedding_original_query, article_embeddings)

print(f" len embedding vector={len(embedding_original_query)}, len artciles␣
 ↪embedded={len(article_embeddings)}, len cosine_distance␣
 ↪vector={len(cosine_similarities_original)}")
print(cosine_similarities_original[:5])
print("\n")

sorted_articles_original = sort_articles_by_cosine_similarity(articles[llm],␣
 ↪cosine_similarities_original)
```

```
 len embedding vector=1536, len artciles embedded=26, len cosine_distance
vector=26
[0.8586576952397326, 0.8529322372028058, 0.8813125051656486, 0.8697606885503189,
0.8401190639278262]


Top 5 articles scores: [0.8844741226876582, 0.8813125051656486,
0.8789248038009736, 0.8744655908578396, 0.8744095819084349]
```

## 7.3 Final Answer: Calling LLM to answer the user query

### 7.3.1 Against the hypothetical answer and the original user query

[ ]:

```
#Using the retrieved context coming from the hypothetical answer
formatted_top_results_hypoth = context_retrieval(sorted_articles_hypoth)
final_answer_hypoth = get_final_answer(user_query,␣
 ↪formatted_top_results_hypoth, llm)
print("Final answer against the Hypothetical query")
display.display(display.Markdown(final_answer_hypoth))

#Using the retrieved context coming from the original answer
formatted_top_results_original = context_retrieval(sorted_articles_original)
final_answer_original = get_final_answer(user_query,␣
 ↪formatted_top_results_original, llm)
print("Final answer against the original query")
display.display(display.Markdown(final_answer_original))
```

```
Final answer against the Hypothetical query
```

The global outage caused by CrowdStrike's botched software update had a significant impact on various sectors globally, leading to chaos and disruptions. However, in terms of Microsoft's stock price, the direct impact of CrowdStrike's outage on Microsoft's stock price is not explicitly mentioned in the provided search results.

CrowdStrike's stock fell by 11% due to the software update issue, but there is no specific information on how this influenced Microsoft's stock price. The outage affected users of Microsoft's Windows operating system, which contributed to the widespread disruptions.

For more details on the impact of CrowdStrike's outage on Microsoft's stock price, you may refer to the search results: 1. CrowdStrike CEO Updates Solutions to Global Microsoft Outage 2. Microsoft's CrowdStrike leaves business black and blue in India

These articles may provide further insights into how the global outage caused by CrowdStrike's software update impacted Microsoft and potentially its stock price.

```
Final answer against the original query
```

The global outage of CrowdStrike, a cybersecurity company used extensively by Microsoft, had a significant impact on both CrowdStrike's stock price and global systems. CrowdStrike experienced a botched software update that caused chaos around the world, leading to global outages and a drop in its stock price by 11%. This incident affected services for their 30,000 subscribers and prompted a rethink among investors and customers.

As for Microsoft, the outage caused disruptions in various sectors, including emergency services, medical practices, airlines, banks, and more. The outage was attributed to a dodgy channel file related to CrowdStrike's software update. This had a cascading effect on systems that rely on Microsoft's Windows operating system, leading to further disruptions.

Overall, the outage had a negative impact on CrowdStrike's stock price and operations, as highlighted in the following articles: 1. CrowdStrike chaos could prompt rethink among investors, customers 2. CrowdStrike shares sink as global IT outage savages systems worldwide 3. Microsoft's

CrowdStrike leaves business black and blue in India

It’s important to note that the impact on Microsoft’s stock price specifically was not explicitly mentioned in the search results provided.

## 8 All together: With the 3 LLMs

```
[ ]: llms
```

```
[ ]: ['gpt-3.5-turbo', 'gpt-4o-mini', 'gpt-4o']
```

```
[ ]: embedding_original_query = get_embeddings(user_query)[0]

     for llm in llms:
       print(llm)
       #Hypothetical answer
       embedding_hypoth = get_embeddings(hypoth_answer_llms[llm])[0]
       article_embeddings = get_embeddings_articles(articles[llm])
       cosine_similarities_hypoth = calculate_cosine_distance(embedding_hypoth,␣
      ↪article_embeddings)
       print("Hypothetical Answer: Most relevant News\n")
       sorted_articles_hypoth = sort_articles_by_cosine_similarity(articles[llm],␣
      ↪cosine_similarities_hypoth)
       print("-"*50)

       #Original Query
       cosine_similarities_original=␣
      ↪calculate_cosine_distance(embedding_original_query, article_embeddings)
       print("Original Answer: Most relevant News\n")
       sorted_articles_original = sort_articles_by_cosine_similarity(articles[llm],␣
      ↪cosine_similarities_original)
       print("-"*50)

       formatted_top_results_hypoth = context_retrieval(sorted_articles_hypoth)
       final_answer_hypoth = get_final_answer(user_query,␣
      ↪formatted_top_results_hypoth, llm)
       print("Final answer against the Hypothetical query")
       display.display(display.Markdown(final_answer_hypoth))
       print("-"*50)

       formatted_top_results_original = context_retrieval(sorted_articles_original)
       final_answer_original = get_final_answer(user_query,␣
      ↪formatted_top_results_original, llm)
       print("Final answer against the original query")
       display.display(display.Markdown(final_answer_original))
       print("-"*50)
       print("-"*50)
```

```
gpt-3.5-turbo
Length of articles to embed: 26
Hypothetical Answer: Most relevant News

Top 5 articles scores: [0.9056245013678469, 0.9054580631940904,
0.9040203721717224, 0.9037450078676516, 0.9022586274192076]

--------------------------------------------------
Original Answer: Most relevant News

Top 5 articles scores: [0.8843513564825746, 0.8812493480390219,
0.8790028253085169, 0.8745070837943586, 0.8743864449395214]

--------------------------------------------------
Final answer against the Hypothetical query
```

The global outage caused by CrowdStrike's botched software update had a significant impact on various sectors and businesses, including Microsoft. CrowdStrike's software update led to disruptions in services, affecting emergency services, medical practices, airlines, banks, and more worldwide. This resulted in CrowdStrike's stock price plummeting by 19%, prompting concerns among investors and customers.

As for Microsoft, the outage caused chaos and system snarls, impacting users of Microsoft's Windows operating system. This event had repercussions worldwide, causing disruptions in India with airlines canceling flights, hospitals, banks, and various businesses facing operational challenges. However, the direct impact on Microsoft's stock price is not explicitly mentioned in the search results provided.

For more detailed information on the impact of the global outage of CrowdStrike on Microsoft's stock price, you can refer to the following search result URLs: 1. CrowdStrike chaos could prompt rethink among investors, customers 2. CrowdStrike CEO Updates Solutions to Global Microsoft Outage 3. CrowdStrike shares sink as global IT outage savages systems worldwide 4. Microsoft's CrowdStrike leaves business black and blue in India

```
--------------------------------------------------
Final answer against the original query
```

The global outage caused by a botched software update from CrowdStrike had significant repercussions, impacting various sectors worldwide. This incident led to global outages affecting emergency services, medical practices, airlines, banks, and more. CrowdStrike's stock price plummeted by 11% and later by more than 19% amidst the chaos, prompting concerns among investors and customers.

As for Microsoft, the outage that started by affecting users of Microsoft's Windows operating system was attributed to CrowdStrike's cybersecurity firm. This outage snarled systems from airports to stock exchanges, causing disruptions globally. However, there is no direct mention of the impact on Microsoft's stock price in the provided search results.

For more information, you can refer to the following links: 1. CrowdStrike chaos could prompt rethink among investors, customers 2. CrowdStrike CEO Updates Solutions to Global Microsoft Outage 3. Microsoft's CrowdStrike leaves business black and blue in India

```
gpt-4o-mini
```

```
Length of articles to embed: 23
Hypothetical Answer: Most relevant News

Top 5 articles scores: [0.8985451259633306, 0.89684699586748,
0.8845112396187657, 0.8836237320217197, 0.8829846407367234]

--------------------------------------------------
Original Answer: Most relevant News

Top 5 articles scores: [0.8822393199106383, 0.8679302189191838,
0.867538207728775, 0.8593652185006599, 0.8586576952397326]

--------------------------------------------------
Final answer against the Hypothetical query
```

The recent global outage resulting from a software update by CrowdStrike, which significantly affected Microsoft systems, led to widespread chaos and operational disruptions across various sectors. However, the specific impact on Microsoft's stock price was not detailed in the search results provided.

The incident caused millions of Microsoft systems to crash globally, impacting banks, airports, TV stations, grocery stores, and more, highlighting the vulnerabilities in our interconnected digital infrastructure. Reports indicated that many businesses had to switch to cash-only operations or completely shut down until systems were restored. This extensive disruption underscored the risks posed by the reliance on technology and the potential financial ramifications for companies like Microsoft that are intertwined with such systems source 1, source 2, source 3, source 4, source 5.

While the search results reflect the broader operational impacts of the outage, including significant consequences for various industries, they did not provide current stock performance metrics or analyses reflecting investor reactions in the wake of the outage. Therefore, for specific information regarding Microsoft's stock price movements or investor sentiment post-outage, one would need to refer to financial news or stock market analysis platforms for real-time updates and insights.

```
--------------------------------------------------
Final answer against the original query
```

The global outage linked to a faulty update from CrowdStrike, which significantly affected millions of Microsoft systems, caused wide-ranging disruptions across various sectors including banks, airports, and healthcare. However, the search results provided do not specifically address the impact of this incident on Microsoft's stock price directly.

The reports detail how the outage led to chaos at airports, grocery stores, and other businesses, and resulted in a choice between going cash-only or closing until systems were restored. Flights were grounded, medical procedures were canceled, and numerous other critical services were impacted by the disruption in IT systems globally. This outage is described as causing significant economic losses and operational paralysis across various industries, highlighting the severity of the situation.

While we can infer that such widespread issues could potentially affect investor confidence and, consequently, stock prices, the search results do not provide any specific data or analysis regarding the direct impact on Microsoft's stock. For a thorough understanding of stock market reactions, one would typically look at financial news sources or stock market analytics which may not have been included in the search results.

For more details on the outage itself, you can refer to these articles: - The Global CrowdStrike Outage Triggered a Surprise Return to Cash - Huge Microsoft Outage, Linked to CrowdStrike, Takes Down Computers Around the World - CrowdStrike IT Outage Cripples the World - Banks and payments hit as faulty CrowdStrike update causes global Microsoft outage - Microsoft Outage: CrowdStrike Update Causes Chaos for Flights, Hospitals and Businesses Globally

If you're interested in how stock prices reacted, you might want to check financial news outlets or stock tracking services for the latest data and market analysis.

```
gpt-4o
Length of articles to embed: 40
Hypothetical Answer: Most relevant News

Top 5 articles scores: [0.9242878502864379, 0.9098273026927012,
0.9022636887337233, 0.9003755977571286, 0.8961472978984708]

--------------------------------------------------
Original Answer: Most relevant News

Top 5 articles scores: [0.9069811516950537, 0.8862757478801461,
0.885469855492687, 0.8822393199106383, 0.8758189694686986]

--------------------------------------------------
Final answer against the Hypothetical query
```

Based on the search results, there is no direct mention of the specific impact of the CrowdStrike outage on Microsoft's stock price. However, it's clear that the outage caused by CrowdStrike's software update caused widespread disruptions, affecting airlines, banks, supermarkets, and emergency services (source).

CrowdStrike's shares fell significantly, plunging by as much as 20% in premarket trading and experiencing a 13% drop as various businesses and services around the globe were disrupted due to the outage linked to their software (source).

Although the specific effect on Microsoft's stock isn't detailed in the search results, the extent of the disruption and the reliance of Microsoft services on CrowdStrike's cybersecurity solutions imply potential repercussions for Microsoft's performance and operational continuity during the outage period. Other sources may need to be consulted to provide a precise impact assessment on Microsoft's stock.

To explore further details about the involvement of Microsoft and related impact, you may consider examining the following articles: Crowdstrike shares plunge in premarket after massive global IT outage, CrowdStrike shares fall more than 13% as global IT outage grounds flights, cuts off 911 access.

```
--------------------------------------------------
Final answer against the original query
```

The global IT outage linked to CrowdStrike, which extensively impacts Microsoft's services, had several significant impacts, but the specific effect on Microsoft's stock price isn't directly mentioned in the provided search results. The outage caused substantial disruptions across various sectors, including airlines, banks, and supermarkets. It led to global chaos, grounding flights, and cutting off 911 access (source).

The chaos also had a ripple effect on the stock market, where the Dow Jones Industrial Average dropped 200 points shortly after the market opened (source). This indicates broader market turbulence, possibly affecting many stocks, including Microsoft's.

However, what is reported is the significant plunge in CrowdStrike's shares, falling as much as 20% in premarket trading (source). Additionally, CrowdStrike's rivals saw a boost in their stock prices as they benefited from the issues CrowdStrike faced (source).

Given these references, while it's clear that the outage affected multiple sectors and caused significant market movements, the exact impact on Microsoft's stock price specifically isn't documented in the provided sources.

```
[ ]: len(sorted_articles_hypoth)
```

```
[ ]: 26
```

# 9 Evaluation

We will be using DeepEval, to compute 3 metrics:

- Faithfulness
- Context Relevancy
- Anwser Relevancy

```
[ ]: !pip install deepeval -q
```

You need to speficy your OpenAI API key to use DeepEval, in our case.

To compute metrics, this library makes several calls to a given LLM, per default they are using GPT-4o. You can use a custom LLM if you want.

However note that the under-hood pormpt templates, in the metrics, the LLM is asked to outpout a json format, if you are using a small LLM, this part may not work.

```
[ ]: from google.colab import userdata
     OPENAI_API_KEY = userdata.get('OPENAI_API_KEY')
     import os
     os.environ[ "OPENAI_API_KEY" ] = OPENAI_API_KEY
```

## 9.1 Faithfullness: Retrieved Context vs LLM's final answer

This evaluates the factual consistency of the **generated answer** relative to the **provided context**.

it outputs a **reason** for its **metric score**.

### 9.1.1 Methodology

1- Use an LLM to break it into statements

2- Using an LLM, assert if the statement can or not be inferred from the context  Verdict: yes or no or idk.

3- Compute Faithfulness Score:

Faithfulness= Number of Truthful Claims/Total Number of Claims

https://docs.confident-ai.com/docs/metrics-faithfulness

[ ]:

```python
from deepeval.metrics import FaithfulnessMetric
from deepeval.test_case import LLMTestCase
```

[ ]:

```python
def get_faithfulness_metric(user_query,final_answer,formatted_top_results):
  metric = FaithfulnessMetric(
    threshold=0.7,
    model="gpt-4o",
    include_reason=True
  )
  test_case = LLMTestCase(
      input= user_query,
      actual_output=final_answer,
      retrieval_context=formatted_top_results
  )

  metric.measure(test_case)
  score = metric.score
  reason = metric.reason
  return score, reason
```

### 9.1.2 Original query

[ ]:

```python
metric = FaithfulnessMetric(
    threshold=0.7,
    model="gpt-4o",
    include_reason=True
)
test_case = LLMTestCase(
    input=user_query,
    actual_output=final_answer_original,
    retrieval_context=formatted_top_results_original
)

metric.measure(test_case)
print(metric.score)
print(metric.reason)
```

```
Output()
```

```
Event loop is already running. Applying nest_asyncio patch to allow async execution…
```

```
0.8
The score is 0.80 because the actual output incorrectly states that
CrowdStrike's stock price dropped by 11% when the retrieval context mentions it
is down more than 19%, and it also incorrectly claims an outage had a cascading
effect on systems relying on Microsoft's Windows operating system, which the
retrieval context does not mention.
```

### 9.1.3 Hypothetical answer

```
[ ]: import nest_asyncio
     nest_asyncio.apply()
```

```
[ ]: score, reason = get_faithfulness_metric(user_query,final_answer_hypoth,formatted_top_results_hypoth)
     print(f"Score: {score}, Reason: {reason}")
```

```
[ ]: score, reason = get_faithfulness_metric(user_query,final_answer_original,formatted_top_results_original)
```

```
Output()
```

```
Event loop is already running. Applying nest_asyncio patch to allow async execution…
```

## 9.2 Context Relevancy

This evaluates how relevant the **retrieved context** is to the **input query**.

It outputs a **reason** for its **metric score**.

### 9.2.1 Methodology

1- Use an LLM to extract statements from the retrieved context

2- Using an LLM, assert if each statement is relevant to the input query ==> yes or no.

3- Compute Contextual Relevancy Score:

Contextual Relevancy= Number of Relevant Statements/Total Number of Statements

https://docs.confident-ai.com/docs/metrics-contextual-relevancy

```
[ ]: from deepeval.metrics import ContextualRelevancyMetric
     # from deepeval.test_case import LLMTestCase
```

```
[ ]: def␣
      ↪get_context_relevancy_metric(user_query,final_answer_hypoth,formatted_top_results_hypoth):
      ↪
       metric = ContextualRelevancyMetric(
         threshold=0.7,
         model="gpt-4o",
         include_reason=True
       )
       test_case = LLMTestCase(
           input= user_query,
           actual_output=final_answer_hypoth,
           retrieval_context=formatted_top_results_hypoth
       )

       metric.measure(test_case)
       score = metric.score
       reason = metric.reason
       return score, reason
```

### 9.2.2 Original query

```
[ ]: score, reason =␣
      ↪get_context_relevancy_metric(user_query,final_answer_original,formatted_top_results_origina
     print(f"Score: {score}, Reason: {reason}")
```

```
Output()
```

```
Event loop is already running. Applying nest_asyncio patch to allow async␣
 ↪execution…
```

```
Score: 0.0, Reason: The score is 0.00 because the context focuses on
CrowdStrike's stock performance and issues without mentioning Microsoft's stock
price.
```

### 9.2.3 Hypothetical answer

[ ]:

```python
score, reason = get_context_relevancy_metric(user_query,final_answer_hypoth,formatted_top_results_hypoth)
print(f"Score: {score}, Reason: {reason}")
```

```
Output()
```

```
Event loop is already running. Applying nest_asyncio patch to allow async execution…
```

```
Score: 0.0, Reason: The score is 0.00 because the context discusses
CrowdStrike's stock and disruptions but does not mention Microsoft's stock price
or its impact.
```

## 9.3 Answer Relevancy

The answer relevancy metric measures the quality of your RAG pipeline's generator by evaluating how relevant the **actual_output** (final answer) of your LLM application is compared to the provided **input**.

deepeval's answer relevancy metric is a self-explaining LLM-Eval, meaning it outputs a **reason** for its **metric score**.

https://docs.confident-ai.com/docs/metrics-answer-relevancy

[ ]:

```python
from deepeval.metrics import AnswerRelevancyMetric
```

[ ]:

```python
def get_answer_relevancy_metric(user_query,final_answer_hypoth):
  metric = AnswerRelevancyMetric(
    threshold=0.7,
    model="gpt-4o",
    include_reason=True
  )
  test_case = LLMTestCase(
      input= user_query,
      actual_output=final_answer_hypoth,
      # retrieval_context=formatted_top_results_hypoth
  )

  metric.measure(test_case)
  score = metric.score
  reason = metric.reason
  return score, reason
```

### 9.3.1 Original query

[ ]:

```
score, reason = get_answer_relevancy_metric(user_query,final_answer_original)
print(f"Score: {score}, Reason: {reason}")
```

```
Output()
```

```
Event loop is already running. Applying nest_asyncio patch to allow async␣
 ↪execution…
```

```
Score: 0.09090909090909091, Reason: The score is 0.09 because the statements
focus on CrowdStrike's impact, global systems, and broader disruptions, but fail
to specifically address Microsoft's stock price.
```

### 9.3.2 Hypothetical answer

[ ]:

```
score, reason = get_answer_relevancy_metric(user_query,final_answer_hypoth)
print(f"Score: {score}, Reason: {reason}")
```

```
Output()
```

```
Event loop is already running. Applying nest_asyncio patch to allow async␣
 ↪execution…
```

```
Score: 0.8333333333333334, Reason: The score is 0.83 because the output
correctly mentions the CrowdStrike outage but includes irrelevant details about
CrowdStrike's share price and the disruptions caused by the outage, rather than
directly addressing the impact on Microsoft's stock price.
```

# 10 All together: LLMs + Evaluation

## 10.1 Run All

[ ]:

```
def get_all_eval_metrics(user_query,final_answer,formatted_top_results):
    score_faithfulness, reason_faithfulness =␣
 ↪get_faithfulness_metric(user_query,final_answer,formatted_top_results)
    score_cxt_relev, reason_cxt_relev =␣
 ↪get_context_relevancy_metric(user_query,final_answer,formatted_top_results)
```

```
  score_answ_relev, reason_answ_relev =␣
 ↪get_answer_relevancy_metric(user_query,final_answer)
  print("\nFaithfulness\n")
  print(score_faithfulness, reason_faithfulness)
  print("\nContext Relevancy\n")
  print(score_cxt_relev, reason_cxt_relev)
  print("\nAnswer Relevancy\n")
  print(score_answ_relev, reason_answ_relev)

  return score_faithfulness, reason_faithfulness, score_cxt_relev,␣
 ↪reason_cxt_relev, score_answ_relev, reason_answ_relev
    # return (score_faithfulness, reason_faithfulness), (score_cxt_relev,␣
 ↪reason_cxt_relev), (score_answ_relev, reason_answ_relev)
```

[ ]:

```
# llm = llms[0]

embedding_original_query = get_embeddings(user_query)[0]
scores_hypoth = {}
scores_original = {}

for llm in llms:
  print(llm)
  #Hypothetical answer
  embedding_hypoth = get_embeddings(hypoth_answer_llms[llm])[0]
  article_embeddings = get_embeddings_articles(articles[llm]) #{list of␣
 ↪embedded articles , there are 26 articles}
  cosine_similarities_hypoth = calculate_cosine_distance(embedding_hypoth,␣
 ↪article_embeddings)
  print("Hypothetical Answer: Most relevant News\n")
  sorted_articles_hypoth = sort_articles_by_cosine_similarity(articles[llm],␣
 ↪cosine_similarities_hypoth)
  print("-"*50)

  #Original Query
  cosine_similarities_original=␣
 ↪calculate_cosine_distance(embedding_original_query, article_embeddings)
  print("Original Answer: Most relevant News\n")
  sorted_articles_original = sort_articles_by_cosine_similarity(articles[llm],␣
 ↪cosine_similarities_original)
  print("-"*50)

  formatted_top_results_hypoth = context_retrieval(sorted_articles_hypoth)
  final_answer_hypoth = get_final_answer(user_query,␣
 ↪formatted_top_results_hypoth, llm)
  print("#Final answer against the Hypothetical query")
  display.display(display.Markdown(final_answer_hypoth))
```

```
    print("-"*50)

    #Get Evaluations Metrics
    scores_hypoth[llm] =␣
   ↪get_all_eval_metrics(user_query,final_answer_hypoth,formatted_top_results_hypoth)

    formatted_top_results_original = context_retrieval(sorted_articles_original)
    final_answer_original = get_final_answer(user_query,␣
   ↪formatted_top_results_original, llm)
    print("#Final answer against the original query")
    display.display(display.Markdown(final_answer_original))

    #Get Evaluations Metrics
    scores_original[llm] =␣
   ↪get_all_eval_metrics(user_query,final_answer_original,formatted_top_results_original)
```

## 10.2 Final Results

```
[ ]: scores_original
```

```
[ ]: {'gpt-3.5-turbo': (0.8888888888888888,
       'The score is 0.89 because the actual output claims the botched software
     update affected services for its 30,000 subscribers, which is not mentioned in
     the retrieval context.',
       0.0,
       "The score is 0.00 because the context does not provide any information about
     Microsoft's stock price, focusing only on the impact of CrowdStrike's outage on
     other services and its own stock.",
       0.16666666666666666,
       "The score is 0.17 because the output extensively discusses various aspects of
     CrowdStrike's outage but fails to address the specific impact on Microsoft's
     stock price."),
      'gpt-4o-mini': (1.0,
       'Fantastic job! The faithfulness score is 1.00 because there are no
     contradictions between the actual output and the retrieval context. Keep up the
     great work!',
       0.0,
       "The score is 0.00 because the context discusses operational impacts of the
     CrowdStrike outage but does not provide any information on Microsoft's stock
     price.",
       0.75,
       "The score is 0.75 because the answer provides relevant information but
     includes specific examples of industry impact and references for further reading
     that do not directly address the question about Microsoft's stock price."),
      'gpt-4o': (1.0,
       'The score is 1.00 because there are no contradictions. Great job on
     maintaining perfect alignment with the retrieval context!',
       0.0,
       "The score is 0.00 because the context does not provide any information about
     the impact of the global outage on Microsoft's stock price.",
       0.6,
       "The score is 0.60 because while the actual output does mention related
     information about CrowdStrike, it includes several irrelevant details about
     CrowdStrike and other entities, which do not directly address the impact on
     Microsoft's stock price.")}
```

```
[ ]: scores_hypoth
```

```
[ ]: {'gpt-3.5-turbo': (0.8,
       "The score is 0.80 because the actual output incorrectly states that
     CrowdStrike's stock price fell by more than 11%, while the correct figure from
     the retrieval context is more than 19%.",
       0.0,
       "The score is 0.00 because the context focuses on CrowdStrike's issues and
     stock performance but does not provide any information about the impact on
     Microsoft's stock price.",
       0.3,
       "The score is 0.30 because the output contains multiple statements about
     CrowdStrike and other unrelated information instead of specifically discussing
     the impact on Microsoft's stock price."),
      'gpt-4o-mini': (1.0,
       'The score is 1.00 because there are no contradictions, indicating that the
     actual output is perfectly aligned with the retrieval context. Great job!',
       0.0,
       "The score is 0.00 because the context does not provide any information about
     the impact on Microsoft's stock price.",
       0.8571428571428571,
       "The score is 0.86 because while the response addresses the impact on
     Microsoft's stock price, it includes irrelevant information by directing the
     reader to external articles rather than focusing solely on the stock price
     impact."),
      'gpt-4o': (0.9,
       "The score is 0.90 because the claim states that the global disruption
     involved Microsoft’s Windows platform, but the retrieval context does not
     mention Microsoft's Windows platform being involved.",
       0.0,
       "The score is 0.00 because the context only discusses CrowdStrike's stock and
     the general disruption from the outage, without mentioning Microsoft's stock
     price or its impact.",
       0.5,
       "The score is 0.50 because while there is some mention of disruptions and
     impacts across various industries, most of the statements do not directly
     address the impact on Microsoft's stock price, thus reducing the relevancy.")}
```

```
[ ]: import pandas as pd
     import numpy as np
```

### 10.2.1 Against Original Query

```
[ ]: scores_original_values = {}
     for llm in llms:
       scores = [score for score in scores_original[llm] if type(score)!= str]
       scores_original_values[llm] = scores

     pd.DataFrame(scores_original_values,␣
      ↪index=['faithfulness','context_relevancy','answer_relevancy'])
```

```
[ ]:                    gpt-3.5-turbo  gpt-4o-mini  gpt-4o
     faithfulness            0.888889         1.00     1.0
     context_relevancy       0.000000         0.00     0.0
     answer_relevancy        0.166667         0.75     0.6
```

```
[ ]: scores_original_raisons = {}
     for llm in llms:
       raisons = [score for score in scores_original[llm] if type(score)== str]
       scores_original_raisons[llm] = raisons

     pd.DataFrame(scores_original_raisons,␣
      ↪index=['faithfulness','context_relevancy','answer_relevancy'])
```

```
[ ]:                                                         gpt-3.5-turbo  \
     faithfulness       The score is 0.89 because the actual output cl…
     context_relevancy  The score is 0.00 because the context does not…
     answer_relevancy   The score is 0.17 because the output extensive…


                                                               gpt-4o-mini  \
     faithfulness       Fantastic job! The faithfulness score is 1.00 …
     context_relevancy  The score is 0.00 because the context discusse…
     answer_relevancy   The score is 0.75 because the answer provides …


                                                                    gpt-4o
     faithfulness       The score is 1.00 because there are no contrad…
     context_relevancy  The score is 0.00 because the context does not…
     answer_relevancy   The score is 0.60 because while the actual out…
```

```
[ ]: index_metrics=['faithfulness','context_relevancy','answer_relevancy']
     for llm in llms:
       print(f"#{llm}:")
       for i in range(len(index_metrics)):
         print(f"{index_metrics[i]}")
         print(scores_original_raisons[llm][i])
```

```
    print("-"*50)
```

```
#gpt-3.5-turbo:
faithfulness
The score is 0.89 because the actual output claims the botched software update
affected services for its 30,000 subscribers, which is not mentioned in the
retrieval context.
context_relevancy
The score is 0.00 because the context does not provide any information about
Microsoft's stock price, focusing only on the impact of CrowdStrike's outage on
other services and its own stock.
answer_relevancy
The score is 0.17 because the output extensively discusses various aspects of
CrowdStrike's outage but fails to address the specific impact on Microsoft's
stock price.
--------------------------------------------------
#gpt-4o-mini:
faithfulness
Fantastic job! The faithfulness score is 1.00 because there are no
contradictions between the actual output and the retrieval context. Keep up the
great work!
context_relevancy
The score is 0.00 because the context discusses operational impacts of the
CrowdStrike outage but does not provide any information on Microsoft's stock
price.
answer_relevancy
The score is 0.75 because the answer provides relevant information but includes
specific examples of industry impact and references for further reading that do
not directly address the question about Microsoft's stock price.
--------------------------------------------------
#gpt-4o:
faithfulness
The score is 1.00 because there are no contradictions. Great job on maintaining
perfect alignment with the retrieval context!
context_relevancy
The score is 0.00 because the context does not provide any information about the
impact of the global outage on Microsoft's stock price.
answer_relevancy
The score is 0.60 because while the actual output does mention related
information about CrowdStrike, it includes several irrelevant details about
CrowdStrike and other entities, which do not directly address the impact on
Microsoft's stock price.
--------------------------------------------------
```

[ ]:

```
for llm in llms:
    mean_score = np.mean([score for score in scores_original[llm] if type(score)!
 ↪= str])
```

```
print(f"{round(mean_score,3)} = Mean score for {llm}")
```

```
0.352 = Mean score for gpt-3.5-turbo
0.583 = Mean score for gpt-4o-mini
0.533 = Mean score for gpt-4o
```

**Key Takeaway 1**: gpt-4o-mini shows the best score among the other LLMs. Its score in answer relevancy was better than the one from gpt-4o.

### 10.2.2 Against Hypothetical Answer

[ ]:

```
scores_hypoth_values = {}
for llm in llms:
  scores = [score for score in scores_hypoth[llm] if type(score)!= str]
  scores_hypoth_values[llm] = scores

pd.DataFrame(scores_hypoth_values,␣
 ↪index=['faithfulness','context_relevancy','answer_relevancy'])
```

[ ]:

```
                   gpt-3.5-turbo  gpt-4o-mini  gpt-4o
faithfulness                 0.8     1.000000     0.9
context_relevancy            0.0     0.000000     0.0
answer_relevancy             0.3     0.857143     0.5
```

[ ]:

```
scores_hypoth_raisons = {}
for llm in llms:
  raisons = [score for score in scores_hypoth[llm] if type(score)== str]
  scores_hypoth_raisons[llm] = raisons

pd.DataFrame(scores_hypoth_raisons,␣
 ↪index=['faithfulness','context_relevancy','answer_relevancy'])
```

[ ]:

```
                                                        gpt-3.5-turbo  \
faithfulness       The score is 0.80 because the actual output in…
context_relevancy  The score is 0.00 because the context focuses …
answer_relevancy   The score is 0.30 because the output contains …


                                                          gpt-4o-mini  \
faithfulness       The score is 1.00 because there are no contrad…
context_relevancy  The score is 0.00 because the context does not…
answer_relevancy   The score is 0.86 because while the response a…


                                                               gpt-4o
faithfulness       The score is 0.90 because the claim states tha…
context_relevancy  The score is 0.00 because the context only dis…
answer_relevancy   The score is 0.50 because while there is some …
```

[ ]:

```python
index_metrics=['faithfulness','context_relevancy','answer_relevancy']
for llm in llms:
  print(f"#{llm}:")
  for i in range(len(index_metrics)):
    print(f"{index_metrics[i]}")
    print(scores_hypoth_raisons[llm][i])
  print("-"*50)
```

```
#gpt-3.5-turbo:
faithfulness
The score is 0.80 because the actual output incorrectly states that
CrowdStrike's stock price fell by more than 11%, while the correct figure from
the retrieval context is more than 19%.
context_relevancy
The score is 0.00 because the context focuses on CrowdStrike's issues and stock
performance but does not provide any information about the impact on Microsoft's
stock price.
answer_relevancy
The score is 0.30 because the output contains multiple statements about
CrowdStrike and other unrelated information instead of specifically discussing
the impact on Microsoft's stock price.
--------------------------------------------------
#gpt-4o-mini:
faithfulness
The score is 1.00 because there are no contradictions, indicating that the
actual output is perfectly aligned with the retrieval context. Great job!
context_relevancy
The score is 0.00 because the context does not provide any information about the
impact on Microsoft's stock price.
answer_relevancy
The score is 0.86 because while the response addresses the impact on Microsoft's
stock price, it includes irrelevant information by directing the reader to
external articles rather than focusing solely on the stock price impact.
--------------------------------------------------
#gpt-4o:
faithfulness
The score is 0.90 because the claim states that the global disruption involved
Microsoft’s Windows platform, but the retrieval context does not mention
Microsoft's Windows platform being involved.
context_relevancy
The score is 0.00 because the context only discusses CrowdStrike's stock and the
general disruption from the outage, without mentioning Microsoft's stock price
or its impact.
answer_relevancy
The score is 0.50 because while there is some mention of disruptions and impacts
across various industries, most of the statements do not directly address the
impact on Microsoft's stock price, thus reducing the relevancy.
```

```
--------------------------------------------------
```

[ ]:

```
for llm in llms:
  mean_score = np.mean([score for score in scores_hypoth[llm] if type(score)!=␣
 ↪str])
  print(f"{round(mean_score,3)} = Mean score for {llm}")
```

```
0.367 = Mean score for gpt-3.5-turbo
0.619 = Mean score for gpt-4o-mini
0.467 = Mean score for gpt-4o
```

**Key Takeaway 2**: Again, in the hypothetical answer, GPT-4o-mini shows the best score among the other LLMs. Its score in answer relevancy was significantly better than GPT-4o (0.85 vs. 0.5) and even better than GPT-3.5-turbo (0.3). Furthermore, its score in faithfulness was better than the others.

**Key Takeaway 3** : Another important takeway:

The score of the results coming from retrieval based on the hypotethical answer (0.619) is better than the one where retrieval is based on the original query (0.583), when using gpt-4o-mini. This higlights the fact that the re-ranking process leads to better results.

**Next:**

**Retrieval Context:**

Even if gpt-4o-mini is showing a good performance, however, the context relevance metric is 0 for all LLMs. This part needs to be reworked again. In the retrieval part, I took title + description and the beginning of the content. That was not enough. A good way needs to be : Parsing the whole html for each article, and gathering all this information together, chunking it in a given size

**DeepEval and gpt-4o-mini:**

It could be interesting to run evaluation metrics with gpt-4o-mini instead of gpt-40. Because the underhood calculation of the scores in the evaluation metrics are based on templated prompts and the capability of the LLM to well compare a given claim/statement in the retrieved context (for example) vs the final answer.

This leads me to this conclusion, because I was not expecting gpt-4o-mini to outperfom gpt-4o!!