IBM's Watson Project Report

Table of Contents

Marginean Ovidiu-Adrian

Moldovan Florin

Key Components

1. **Libraries and API Key:**
   - The script imports several libraries, including **os**, **re**, **time**, **openai**, **whoosh**, and **nltk**.
   - An API key for the OpenAI GPT-3.5-turbo model is provided for chat-based completions.
2. **Schema and Indexing**
   - A Whoosh schema is defined with fields for the title and content of documents.
   - The **create_index** function processes Wikipedia documents, extracts titles and content, tokenizes, stems, and removes stop words from the content, and then creates an index.
3. **Question Processing**
   - The script reads and processes a file containing questions, categories, and answers (**questions.txt**).
   - The **read_and_process_questions** function extracts categories, clues, and answers from the file.
4. **Searching and Evaluation**
   - The script performs searches on the index using both regular querying and GPT-3.5-turbo based querying.
   - The **search_single_query** and **search_single_query_with_GPT** functions evaluate the correctness of the answers.
   - The **search_index_in_question_file** function evaluates the system's performance and prints accuracy metrics.
5. **Answering Questions**
   - The **answer_question** function takes a category and a question, searches the index, and provides answers using GPT-3.5-turbo.
6. **Menu and User Interface**
   - The **main** function provides a simple text-based menu for users to create an index, compare search results, ask a question, or exit the program.
7. **Comparing Search Results**
   - The **compare_search_results** function compares the search results obtained with and without GPT-3.5-turbo.

Indexing Terms Preparation

The terms for indexing in this code undergo a series of preprocessing steps to enhance the efficiency and effectiveness of the information retrieval system. Here are the key steps:

1. **Tokenization:**
    - The content of Wikipedia articles is tokenized using the **nltk.word_tokenize** function, breaking it into individual words.
2. **Stemming:**
    - Porter stemming is applied using the **nltk.stem.PorterStemmer**. This process reduces words to their root or base form, helping to consolidate related terms.
3. **Stop Word Removal:**
    - Stop words, common words that typically do not contribute much to the meaning, are removed. This is achieved using the NLTK library's **stopwords** set for the English language.

Addressing Wikipedia-specific Issues

Wikipedia content poses some unique challenges during the indexing process:

1. **Structured Content**
    - Wikipedia articles often contain structured content, such as links within double square brackets **[[...]]**. The code utilizes regular expressions to identify and remove these links, ensuring that the extracted content is more representative of the actual text.
2. **Title and Content Extraction**
    - The script extracts both the title and content of Wikipedia articles. This dual extraction allows for a more comprehensive representation of the document in the index.
3. **Tokenization Challenges**
    - Tokenizing Wikipedia content might lead to the inclusion of special characters and non-alphanumeric tokens. To address this, the code employs checks to ensure that only valid alphanumeric words are considered during tokenization.

Retrieval Component

The retrieval component is implemented through the **answer_question** function, which takes a category and a Jeopardy clue as input. The retrieval process involves the following steps:

1. **Query Construction**
   - The **build_query_for_text** function preprocesses the input text (Jeopardy clue) by tokenizing, stemming, and removing stop words. The resulting terms are then used to construct queries for the Whoosh index.
2. **Combining Queries**
   - The queries are combined using the **whoosh.query.Or** operator. This allows for flexibility, where a match on any of the terms can contribute to the relevance of a document.
3. **Utilizing Category Information**
   - The **category** information is also incorporated into the queries, enhancing the specificity of the search. The terms from the category are treated similarly to the terms from the Jeopardy clue.

Measuring performance

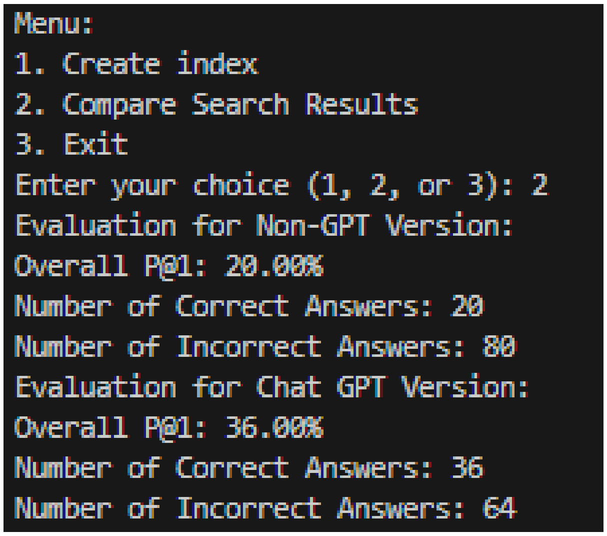
In the provided code, the performance of the Jeopardy system is measured using the Precision at 1 (P@1) metric. P@1 measures the accuracy of the system by evaluating whether the correct answer is ranked first in the search results. This metric is appropriate for evaluating the performance of a question-answering system when there is a single correct answer expected.

Justification for P@1:

1. Relevance to Question-Answering
   - P@1 is well-suited for question-answering scenarios where only one correct answer is expected. In Jeopardy-style questions, there is typically a single correct response.
2. Simplicity and Interpretability
   - P@1 provides a straightforward and interpretable measure of accuracy. It directly indicates the percentage of questions for which the correct answer is ranked first.
3. Focus on Top-ranked Result
   - P@1 places emphasis on the most relevant result, which is essential in scenarios where users are likely to expect the most relevant answer to be presented first.

Performance Reporting:

The performance of the Jeopardy system is reported using the Precision at 1 (P@1) metric. The evaluation is conducted on a set of Jeopardy-style questions, comparing the system's top-ranked answers against the ground truth.



```
Menu:
1. Create index
2. Compare Search Results
3. Exit
Enter your choice (1, 2, or 3): 2
Evaluation for Non-GPT Version:
Overall P@1: 20.00%
Number of Correct Answers: 20
Number of Incorrect Answers: 80
Evaluation for Chat GPT Version:
Overall P@1: 36.00%
Number of Correct Answers: 36
Number of Incorrect Answers: 64
```

The performance evaluation using Precision at 1 (P@1) provides insights into the accuracy of the Jeopardy system. It indicates the proportion of questions for which the correct answer is ranked first in the search results. The reported precision percentage offers a clear measure of how effectively the system retrieves relevant information for Jeopardy-style questions.

Error analysis:

To perform an error analysis of the Jeopardy system implemented in the provided code, we need to analyze the questions that were answered correctly and incorrectly. The analysis aims to identify patterns or classes of errors and understand the system's strengths and weaknesses.

Incorrectly Answered Questions

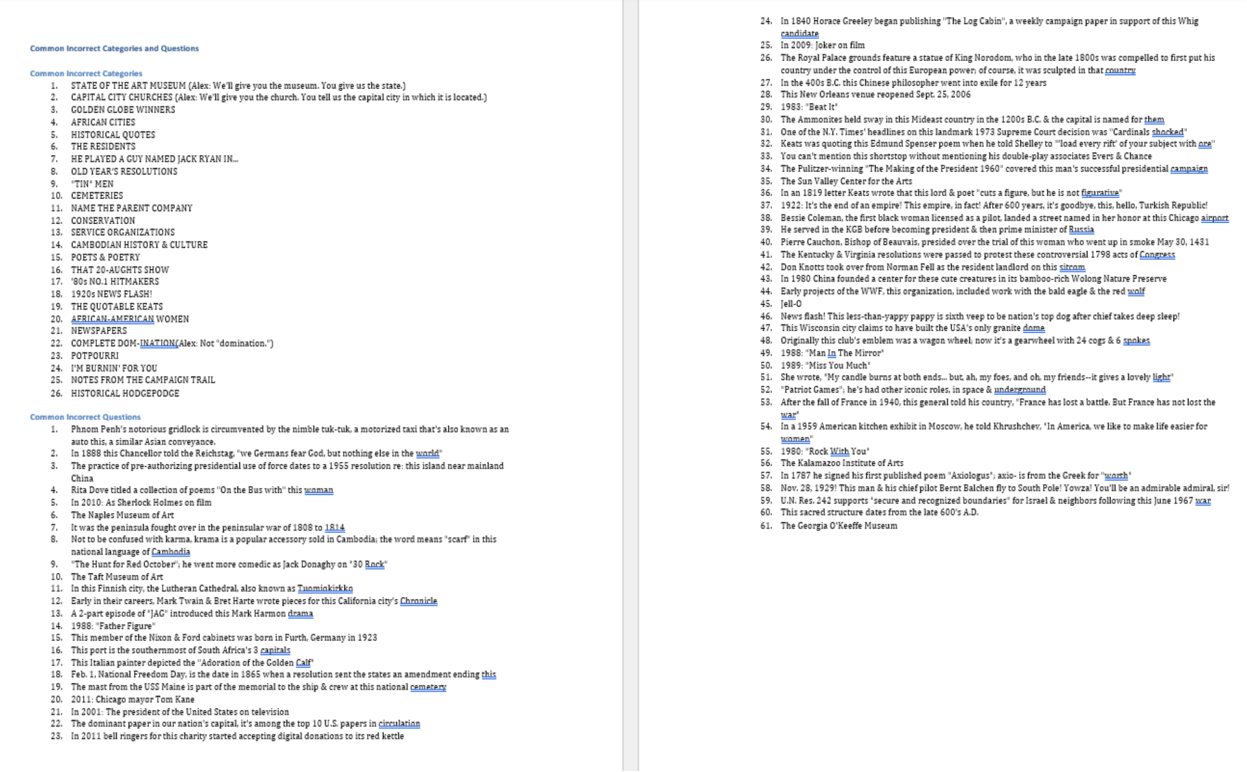
The errors in the system's responses may be grouped into several classes:

1. **Ambiguity in Clues**
    - Some Jeopardy clues may be inherently ambiguous or require additional context for accurate interpretation. The system might struggle with questions where multiple entities or concepts are equally relevant.
2. **Lack of Context**
    - The system relies on the content of Wikipedia articles without considering the context of the question beyond the tokenized terms. This lack of context may lead to incorrect answers when subtle nuances or specific details are essential for accurate responses.
3. **Failure to Capture Synonyms or Variations**

- The system may miss correct answers if the tokenization and stemming process does not capture synonymous or variant expressions present in the Jeopardy clues.

4. **Limited Query Expansion**

   - The system does not implement sophisticated query expansion techniques, which could help broaden the search space for relevant documents. This limitation may result in the system missing documents with related information.



Common Incorrect Categories and Questions

Common Incorrect Categories

1. STATE OF THE ART MUSEUM (Alex: We'll give you the museum. You give us the state.)
2. CAPITAL CITY CHURCHES (Alex: We'll give you the church. You tell us the capital city in which it is located.)
3. GOLDEN GLOBE WINNERS
4. AFRICAN CITIES
5. HISTORICAL QUOTES
6. THE RESIDENTS
7. HE PLAYED A GUY NAMED JACK RYAN IN...
8. OLD YEAR'S RESOLUTIONS
9. "TIN" MEN
10. CEMETERIES
11. NAME THE PARENT COMPANY
12. CONSERVATION
13. SERVICE ORGANIZATIONS
14. CAMBODIAN HISTORY & CULTURE
15. POETS & POETRY
16. THAT 20-AUGHTS SHOW
17. '80s NO.1 HITMAKERS
18. 1920s NEWS FLASH!
19. THE QUOTABLE KEATS
20. AFRICAN-AMERICAN WOMEN
21. NEWSPAPERS
22. COMPLETE DOM-INATION(Alex: Not "domination.")
23. POTPOURRI
24. I'M BURNIN' FOR YOU
25. NOTES FROM THE CAMPAIGN TRAIL
26. HISTORICAL HODGEPODGE

Common Incorrect Questions

1. Phnom Penh's notorious gridlock is circumvented by the nimble tuk-tuk, a motorized taxi that's also known as an auto this, a similar Asian conveyance.
2. In 1888 this Chancellor told the Reichstag, "we Germans fear God, but nothing else in the world"
3. The practice of pre-authorizing presidential use of force dates to a 1955 resolution re: this island near mainland China
4. Rita Dove titled a collection of poems "On the Bus with" this woman
5. In 2010: As Sherlock Holmes on film
6. The Naples Museum of Art
7. It was the peninsula fought over in the peninsular war of 1808 to 1814
8. Not to be confused with karma, krama is a popular accessory sold in Cambodia; the word means "scarf" in this national language of Cambodia
9. "The Hunt for Red October"; he went more comedic as Jack Donaghy on "30 Rock"
10. The Taft Museum of Art
11. In this Finnish city, the Lutheran Cathedral, also known as Tuomiokirkko
12. Early in their careers, Mark Twain & Bret Harte wrote pieces for this California city's Chronicle
13. A 2-part episode of "JAG" introduced this Mark Harmon drama
14. 1988: "Father Figure"
15. This member of the Nixon & Ford cabinets was born in Furth, Germany in 1923
16. This port is the southernmost of South Africa's 3 capitals
17. This Italian painter depicted the "Adoration of the Golden Calf"
18. Feb. 1, National Freedom Day, is the date in 1865 when a resolution sent the states an amendment ending this
19. The mast from the USS Maine is part of the memorial to the ship & crew at this national cemetery
20. 2011: Chicago mayor Tom Kane
21. In 2001: The president of the United States on television
22. The dominant paper in our nation's capital, it's among the top 10 U.S. papers in circulation
23. In 2011 bell ringers for this charity started accepting digital donations to its red kettle
24. In 1840 Horace Greeley began publishing "The Log Cabin", a weekly campaign paper in support of this Whig candidate
25. In 2009: Joker on film
26. The Royal Palace grounds feature a statue of King Norodom, who in the late 1800s was compelled to first put his country under the control of this European power; of course, it was sculpted in that country
27. In the 400s B.C. this Chinese philosopher went into exile for 12 years
28. This New Orleans venue reopened Sept. 25, 2006
29. 1983: "Beat It"
30. The Ammonites held sway in this Mideast country in the 1200s B.C. & the capital is named for them
31. One of the N.Y. Times' headlines on this landmark 1973 Supreme Court decision was "Cardinals shocked"
32. Keats was quoting this Edmund Spenser poem when he told Shelley to "'load every rift' of your subject with ore"
33. You can't mention this shortstop without mentioning his double-play associates Evers & Chance
34. The Pulitzer-winning "The Making of the President 1960" covered this man's successful presidential campaign
35. The Sun Valley Center for the Arts
36. In an 1819 letter Keats wrote that this lord & poet "cuts a figure, but he is not figurative"
37. 1922: It's the end of an empire! This empire, in fact! After 600 years, it's goodbye, this, hello, Turkish Republic!
38. Bessie Coleman, the first black woman licensed as a pilot, landed a street named in her honor at this Chicago airport
39. He served in the KGB before becoming president & then prime minister of Russia
40. Pierre Cauchon, Bishop of Beauvais, presided over the trial of this woman who went up in smoke May 30, 1431
41. The Kentucky & Virginia resolutions were passed to protest these controversial 1798 acts of Congress
42. Don Knotts took over from Norman Fell as the resident landlord on this sitcom
43. In 1980 China founded a center for these cute creatures in its bamboo-rich Wolong Nature Preserve
44. Early projects of the WWF, this organization, included work with the bald eagle & the red wolf
45. Jell-O
46. News flash! This less-than-yappy pappy is sixth veep to be nation's top dog after chief takes deep sleep!
47. This Wisconsin city claims to have built the USA's only granite dome
48. Originally this club's emblem was a wagon wheel; now it's a gearwheel with 24 cogs & 6 spokes
49. 1988: "Man In The Mirror"
50. 1989: "Miss You Much"
51. She wrote, "My candle burns at both ends... but, ah, my foes, and oh, my friends--it gives a lovely light"
52. "Patriot Games"; he's had other iconic roles, in space & underground
53. After the fall of France in 1940, this general told his country, "France has lost a battle. But France has not lost the war"
54. In a 1959 American kitchen exhibit in Moscow, he told Khrushchev, "In America, we like to make life easier for women"
55. 1980: "Rock With You"
56. The Kalamazoo Institute of Arts
57. In 1787 he signed his first published poem "Axiologus"; axio- is from the Greek for "worth"
58. Nov. 28, 1929! This man & his chief pilot Bernt Balchen fly to South Pole! Yowza! You'll be an admirable admiral, sir!
59. U.N. Res. 242 supports "secure and recognized boundaries" for Israel & neighbors following this June 1967 war
60. This sacred structure dates from the late 600's A.D.
61. The Georgia O'Keeffe Museum

Common Incorrect Categories

1. **STATE OF THE ART MUSEUM**

   - **Potential Issue:** The system may struggle with questions related to specific museums and their locations within states. It might not effectively link museum names to their respective states.

2. **CAPITAL CITY CHURCHES**

   - **Potential Issue:** The system might face difficulty associating churches with their respective capital cities. This could be due to insufficient context or difficulty in identifying the locations of specific churches.

3. **GOLDEN GLOBE WINNERS**

- **Potential Issue:** Identifying Golden Globe winners might require up-to-date information, and the system may not be adequately equipped to handle real-time data or recent award winners.

4. **AFRICAN CITIES**

   - **Potential Issue:** Questions about African cities may involve a wide range of geographical and cultural knowledge. The system may struggle to accurately link city names to their corresponding countries or regions.

5. **HISTORICAL QUOTES**

   - **Potential Issue:** Historical quotes may require a nuanced understanding of the context in which they were made. The system may not effectively capture the historical significance or context of specific quotes.

Common Incorrect Questions

The incorrect questions provide additional insights into potential challenges:

1. **Phnom Penh's notorious gridlock**

   - **Potential Issue:** The system may not effectively associate Phnom Penh with Cambodia or recognize the term "tuk-tuk" as an auto-rickshaw commonly used in Asian countries.

2. **In 1888 this Chancellor told the Reichstag**

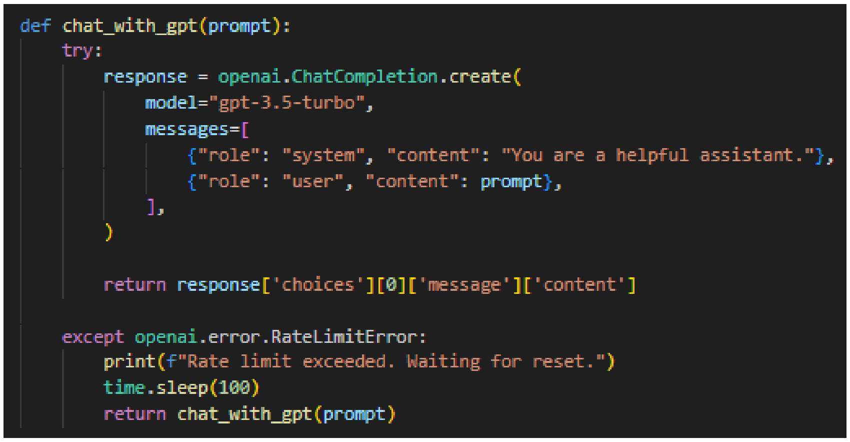
   - **Potential Issue:** The question requires knowledge of historical figures, and the system may not accurately link the quote to Otto von Bismarck, the Chancellor mentioned.

3. **The practice of pre-authorizing presidential use of force**

   - **Potential Issue:** This question involves understanding historical resolutions and their context, and the system may not capture the nuances of the 1955 resolution related to the Taiwan Strait.

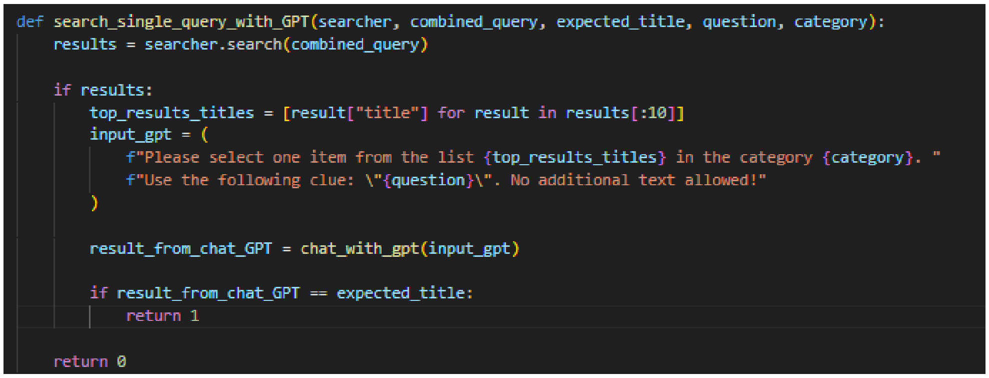
Improving retrieval

For this step we integrated ChatGPT. Utilize the standard Information Retrieval (IR) system to obtain the top K pages based on the Jeopardy clue and category



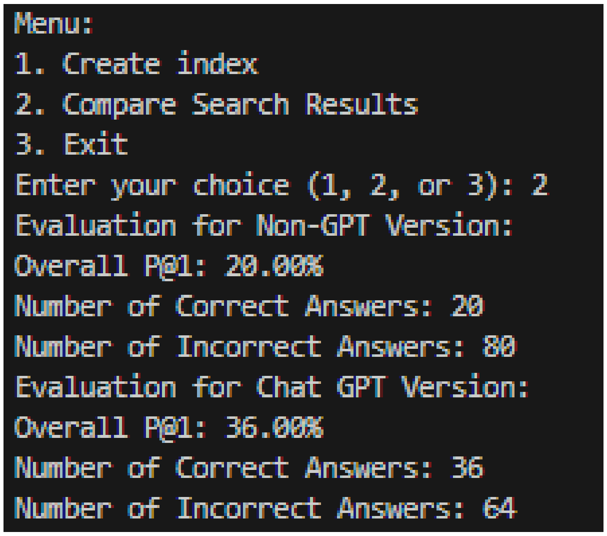
```
def chat_with_gpt(prompt):
    try:
        response = openai.ChatCompletion.create(
            model="gpt-3.5-turbo",
            messages=[
                {"role": "system", "content": "You are a helpful assistant."},
                {"role": "user", "content": prompt},
            ],
        )

        return response['choices'][0]['message']['content']

    except openai.error.RateLimitError:
        print(f"Rate limit exceeded. Waiting for reset.")
        time.sleep(100)
        return chat_with_gpt(prompt)
```

And this is how we used the function.



```
def search_single_query_with_GPT(searcher, combined_query, expected_title, question, category):
    results = searcher.search(combined_query)

    if results:
        top_results_titles = [result["title"] for result in results[:10]]
        input_gpt = (
            f"Please select one item from the list {top_results_titles} in the category {category}. "
            f"Use the following clue: \"{question}\". No additional text allowed!"
        )

        result_from_chat_GPT = chat_with_gpt(input_gpt)

        if result_from_chat_GPT == expected_title:
            return 1

    return 0
```

What is the performance of your system after this improvement?



```
Menu:
1. Create index
2. Compare Search Results
3. Exit
Enter your choice (1, 2, or 3): 2
Evaluation for Non-GPT Version:
Overall P@1: 20.00%
Number of Correct Answers: 20
Number of Incorrect Answers: 80
Evaluation for Chat GPT Version:
Overall P@1: 36.00%
Number of Correct Answers: 36
Number of Incorrect Answers: 64
```