**Data mining project report:  
“Building (a part of) Watson”**

Bucur Răzvan | Misan Maria-Teodora | Muresan Vlad

Universitatea Babes-Bolyai Cluj-Napoca

Data Mining - Fall 2023

**Chapter 1: Introduction and Overview**

This project aims to develop a simplified Question Answering (QA) system inspired by IBM’s Watson, specifically tailored to answer Jeopardy questions using Wikipedia page titles as potential answers. The core objective is to devise an efficient mechanism to classify Wikipedia pages, thereby identifying the most relevant page that corresponds to a given Jeopardy clue.

#### ***Dataset Overview***

Jeopardy Questions: A set of 100 questions derived from previous Jeopardy episodes, formatted to include a category, clue, and the correct answer, reflecting the show's wide-ranging subject matter.

Wikipedia Pages: A comprehensive collection of approximately 280,000 Wikipedia pages, structured to ensure the presence of correct answers within the dataset. This collection is distributed across 80 files, with each page's title marked distinctly within double square brackets (e.g., “[[BBC]]”).

#### ***Methodology***

The project leverages Whoosh, an open-source IR library, for its indexing and search capabilities. The choice of Whoosh is motivated by its adaptability and the rich feature set it offers for custom indexing and query handling. Prior to indexing, the dataset undergoes a series of NLP preprocessing steps, including:

1. Lemmatization: Applied to normalize words to their base form, enhancing the search's linguistic accuracy.
2. Stop Words Removal: Employed to eliminate common words, focusing the search on significant terms.
3. Wikipedia-Specific Content Handling: Strategies are implemented to address redirects and disambiguation pages, ensuring the integrity and relevance of the indexed content.

The retrieval process is intricately designed to convert Jeopardy clues into effective search queries. This involves a critical analysis of clue content, selective word usage, and the strategic incorporation of the question category to refine search results.

# **Chapter 2: Indexing and Retrieval Process**

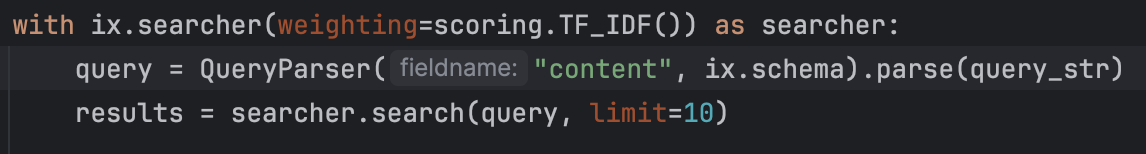
#### ***Preprocessing and Term Preparation***

The foundational step in preparing the Wikipedia dataset for indexing involved an extensive preprocessing phase, utilizing spaCy, a robust NLP library. This decision was motivated by spaCy's comprehensive capabilities in handling complex NLP tasks efficiently. The following preprocessing steps were implemented:

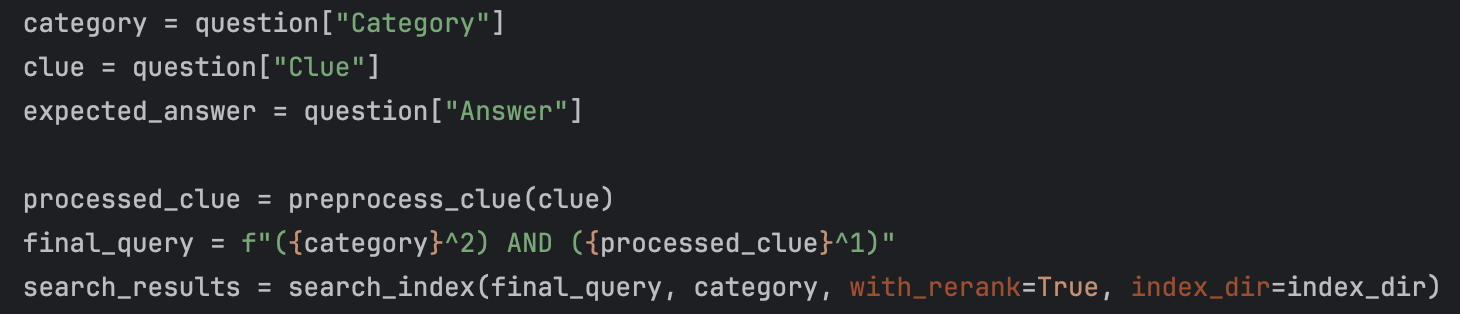
* Lemmatization: Leveraging spaCy's lemmatization allowed for the reduction of words to their base or dictionary forms, ensuring that variations of a word (e.g., "runs," "running") were uniformly indexed under a single lemma (e.g., "run"). This process was crucial for enhancing the accuracy of the search by aligning the query terms with their corresponding indexed forms.
* Stop Words Removal: spaCy's list of stop words was employed to filter out common but low-information words from the text. This step focused the indexing and subsequent searches on the most meaningful components of the content.

***Retrieval Component and Query Construction***

The retrieval process was initially experimented with using Boolean queries. This approach attempted to match documents based on the presence or absence of specific terms extracted directly from the Jeopardy clues. However, this method quickly showed limitations, primarily due to its inability to grasp the context and nuances of the clues effectively, leading to poor retrieval performance.

***TF-IDF Scoring:*** Implementing TF-IDF scoring through Whoosh's searcher provided a mechanism to evaluate the significance of terms within the clue in relation to their distribution across the indexed Wikipedia pages. This approach enabled the retrieval of documents that were not only a textual match but contextually relevant to the clue and also the category given in the jeopardy question.  


***Query Expansion:*** Although not initially pursued, the exploration of Boolean queries highlighted the potential benefits of expanding queries to include synonyms or related terms. When provided, the category of the question was used to refine the search. The category acted as an additional filter, prioritizing Wikipedia pages related to the category's domain.



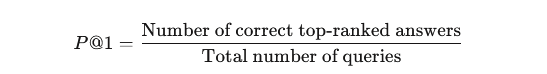
**Chapter 3: Measuring Performance:**

• Measure the performance of your Jeopardy system, using at least one of the metrics discussed in class - justify your choice

We have selected P@1 and MRR as the metrics to evaluate the performance of our system due to their suitability for our specific use case.

### ***P@1***

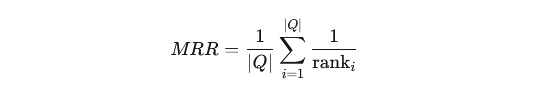
P@1(or precision at one) is a binary evaluation metric that measures whether the top answer provided by the system is correct. For a set of queries, P@1 is calculated as:



This measurement is particularly useful in the context of a Jeopardy game,since only the first response can score points in a Jeopardy game, P@1 effectively captures the system's accuracy where it matters most, making it an essential metric for tuning and evaluating the system's performance.

#### ***MRR***

MRR is a statistic measure for evaluating any system that produces a list of possible answers to a query, ordered by confidence. MRR focuses on the rank of the first correct answer. MRR is calculated as the average of the reciprocal ranks of the first correct answer across all queries, where ∣Q∣ is the number of queries, and rank(i)is the rank of the first correct answer for the ith query:



While P@1 is strict, MRR provides insight into how well the system performs on average, including when the top answer isn't correct but the right answer is still highly ranked. This is useful for understanding and improving the overall effectiveness of the answer ranking algorithm.

MRR helps in tuning the system to not only aim for the top spot but also to ensure that correct answers are not buried down the list. This has been particularly useful during the development phase, when we experimented with different weights for the query terms and tried different retrieval strategies.

***Initial Results:***

The initial version of the Jeopardy question answering system, without the integration of LLM features like ChatGPT, achieved a Precision at 1 of 0.12. The estimated Mean Reciprocal Rank (MRR) for this setup is 0.20, indicating moderate effectiveness in ranking relevant Wikipedia titles within the top positions of search results.

#### ***Chapter 4: Error Analysis***

**Indexing and Preprocessing Limitations**

A significant hurdle was encountered in the initial attempt to utilize built-in stemming and lemmatization during the indexing phase. The rationale behind this approach was to streamline query processing by ensuring a uniform representation of terms within the index. However, this strategy inadvertently compromised the system's ability to engineer more effective queries.

Specifically:

* Over-Generalization: The aggressive normalization of terms led to an oversimplification of content, diminishing the unique context of certain Wikipedia pages and Jeopardy clues. This over-generalization hindered the system's ability to distinguish between nuanced differences in meaning and context.
* Index Size and Efficiency Trade-off: Minimal lemmatization was necessary to preserve the contextual integrity of the indexed content. However, this approach resulted in a larger index size, impacting the efficiency of the retrieval process. Striking a balance between the depth of preprocessing and the manageability of the index size proved challenging.

***Ranking System Challenges***

The initial implementation of the ranking system, devoid of TF-IDF scoring, yielded results that were inconsistent and difficult to rationalize. The primary issues identified were:

* Lack of Contextual Relevance: Without the nuanced weighting provided by TF-IDF scoring, the retrieval component struggled to prioritize results based on the actual relevance of the content to the Jeopardy clues. This often led to seemingly random results being returned.
* Difficulty in Error Diagnosis: The absence of a clear ranking rationale made it challenging to diagnose why certain pages were retrieved over more contextually appropriate ones, complicating efforts to iteratively improve the system.

***Special Case Challenges***

Certain Jeopardy clues presented unique challenges due to their specific requirements or linguistic structures, such as:

* Category Notes: Some questions included additional notes within the category line, providing essential context for correctly interpreting the clue. For example, identifying the capital city based on a given church required not only textual matching but also geographical and historical knowledge. The system's initial design did not account for extracting and utilizing these contextual cues effectively.
* Wordplay: Clues based on wordplay or requiring an understanding of cultural references (e.g., the "fifth Beatle") demanded a level of semantic and cultural comprehension beyond simple keyword matching. These cases highlighted the system's limitations in processing and responding to linguistically creative clues.
* Contextual Understanding: The necessity for external knowledge and context for accurately answering certain clues underscored the system's dependency on the depth and breadth of the indexed content. Without access to broader contextual information, the system struggled to address clues that extended beyond the surface text of the Wikipedia pages.

***Additional Observations***

Some clues were inherently ambiguous, lacking specific details that could lead to a single, definitive answer. This ambiguity posed a challenge for both query formulation and result ranking, as multiple Wikipedia pages might equally satisfy the clue's criteria.

# 

# 

# 

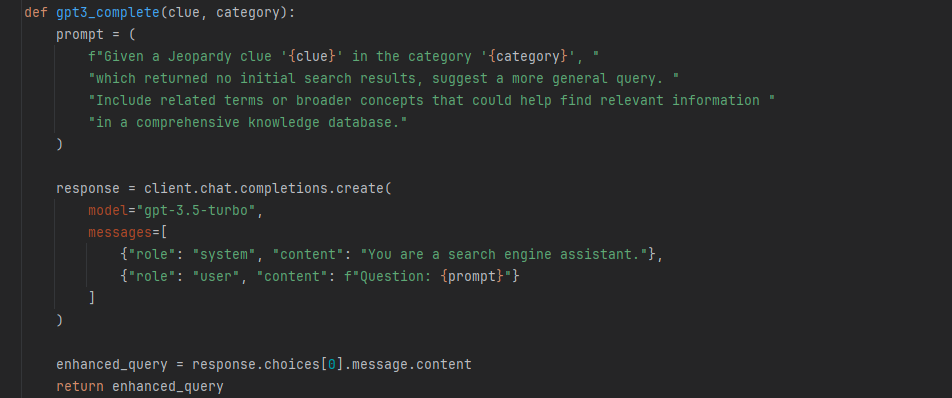
# 

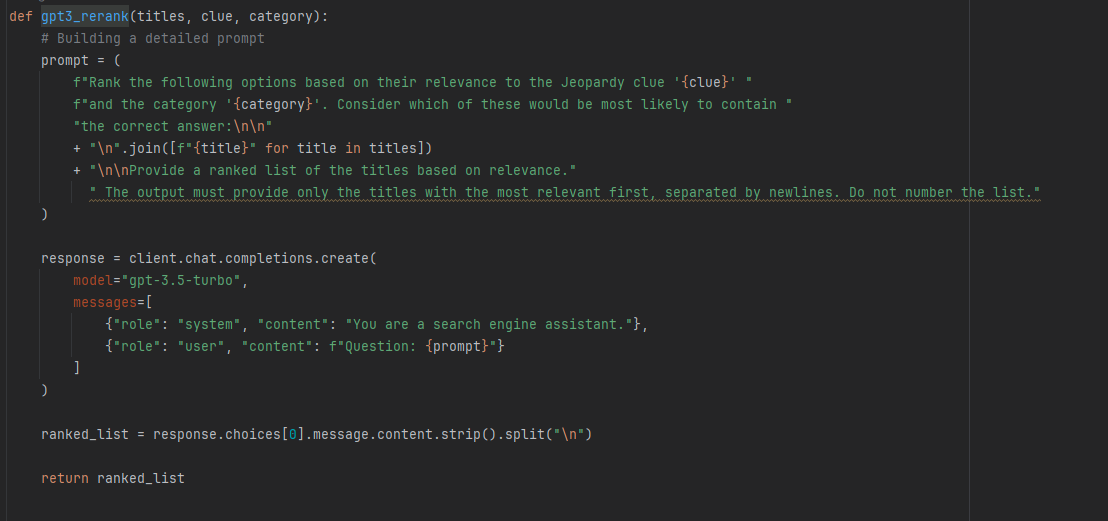
# 

# **Chapter 5: Improving retrieval:**

To improve the accuracy of our system, we implemented two solutions leveraging OpenAi’s GPT3.5 model.

The first improvement we did is more of a failsafe mechanism. If our query does not return any results in the first iteration, we will prompt the model to generate a broader query based on the clue and context of the question:



The second improvement uses the model to rerank the top 10 results from our query:

This measure greatly improves the P@1 and MRR scores, but only for the cases in which the correct document is found within the 10 results our initial query returned. We provide only the titles and not the content of the documents due to the token limitations of the OpenAi api. However, this also ensures that the results of our systems are weighted heavily in the final answers and we didn’t simply build a ChatGPT wrapper.

***Results and Limitations***

Due to ChatGPTs non deterministic nature, there were slight differences across multiple runs. Most common issues we faced were formatting issues where the model would include an introductory sentence, or numbering/formatting the list with bullet points.

After integrating ChatGPT into our Jeopardy question answering system, we observed a significant boost in contextual understanding and query handling, leading to a notable increase in the Mean Reciprocal Rank (MRR). However, this integration introduced a challenge where ChatGPT occasionally added extra words or expanded titles, affecting the Precision at 1 (P@1), which adjusted to around 60%. This phenomenon highlights the need for a delicate balance between leveraging ChatGPT's expansive knowledge for query enrichment and maintaining the precision of search results.

The enhanced version managed to address and correct many of the initial limitations, particularly in dealing with ambiguous clues, wordplay, and questions requiring specific contextual knowledge. These additions although rare heavily skew the P@1 and MRR measurements in a negative way.