

Wikipedia Pages Classification for Answering Jeopardy-like Questions

(Credits: Tudor Lechințan, Mădălin Raiu, Andrei Lupșa)

1. Code Description

The application is written in Java 17 and uses Apache Lucene to index and search through the documents. The program is structured into 3 main parts: the document parser which takes the input files and splits them into documents that can be fed into Lucene, the index builder & searcher which is responsible for creating and querying the document index and, finally, the command line interface for accessing all the functionalities of the application.

Document Parser

The document parser transforms the input files into a format that can be more easily processed by Lucene. The parser iterates over all files in the input directory and tries to split each file into its underlying documents. Each document has a title, a content and a tree of chapters. To achieve good performance and low memory usage, the parser is designed in such a way that not more than one document is loaded in memory at any given time. As such, on our machines, the complete indexing of the input files takes ~2 minutes.

Index Builder & Searcher

The first responsibility of the index builder & searcher is to index the documents retrieved by the parser and save the results on disk. To accomplish this task, we configured Lucene's index writer to save the index to a directory on the file system. We chose to use Lucene's English analyzer for processing the text as it includes a series of filters and stop words tailored specifically for English which is the language used in the input data.

For answering the Jeopardy questions, we used the index searcher provided by Lucene. To build a query string we process the question and its category with an English analyzer and look for the generated terms in the body of the document. The title of the document is not taken into account when searching as most of the time the text of the question does not include any words that can be found in the answer.

Command Line Interface

The application offers a simple command line interface for accessing its functionalities. It has options for creating & deleting a Lucene index, running an arbitrary search query, or answering a list of questions from a questions file.

2. Performance Measurement: Precision at 1 (P@1)

The Precision at 1 (P@1) metric evaluates the accuracy of our search program by measuring the relevance of the top-ranked result. In our analysis, we processed a set of 100 questions and observed that the program returned 20 correct answers, resulting in a P@1 score of 20%.

The 20% precision at the top rank indicates that, on average, the initial search result was accurate for a small portion of the queries. While this is a positive outcome, it also highlights opportunities for improvement in enhancing the precision of the top-ranked results.

The P@1 analysis serves as a baseline for understanding the strengths and areas for improvement in our search program. Continuous monitoring, feedback collection, and strategic enhancements are integral to providing users with a more accurate and effective search experience.

Correctly Answered Questions

The system's capability to correctly answer 20 questions suggests effective direct matching for queries with explicit references to entities or facts that are clearly represented on Wikipedia pages. This indicates that when the clues are straightforward and well-aligned with the titles or primary content of Wikipedia articles, the system performs reliably.

3. Error Analysis

Our Watson-like system was evaluated on a dataset of 100 Jeopardy questions, achieving a success rate of correctly answering 20 questions. This analysis aims to facilitate the understanding of why the system managed to answer these questions correctly and to categorize the errors made on the remaining questions into three general categories: "Incorrect recognition of relevant terms", "Overgeneralization," and "Not Found".

Error Categories

The analysis of the incorrectly answered questions allows us to categorize the errors into several classes.

1. Misinterpretation of Clue

Some errors occurred because the system misinterpreted the main keywords of the clue, leading it to retrieve related but incorrect information. For example, for the question about the Naples Museum of Art, the system found "Michael Bierut" instead of recognizing the query was about the location and answering "Florida".

2. Overgeneralization

In some cases, the system provided an answer that was too general or related to the topic but did not specifically address the question. This indicates a lack of depth in understanding the specificity required by the question. For example, on multiple occasions, when asked about a specific song from a year, the program answers with the record label associated with that song, instead of the actual artist. Example: for "1983: "Beat It"", the program answers "Fontana Records" instead of "Michael Jackson".

3. Misinterpretation of Context

Some errors were due to the program's inability to perceive the bigger context of the clue, and instead focused on singular words from the clue. For example, for the clue "The dominant paper in our nation's capital, it's among the top 10 U.S. papers in circulation", the program answered "Media of the United States", instead of "The Washington Post", most likely because of words like "paper" and "U.S." in the clue.

4. Lack of sufficient information

Sometimes the program answered in the correct manner to the clue, but the answer was just wrong. For example, for the clue "Song that says, "you make me smile with my heart; your looks are laughable, unphotographable"" the program answered "Bess, You Is My Woman Now" instead of "My Funny Valentine".

Areas of Improvement

The analysis of errors, separated into these four categories, highlights significant areas for improvement in the system's design and functionality. Enhancing the system's ability to accurately interpret clues, avoid overgeneralization, and fill the gaps in its knowledge base or retrieval mechanisms could substantially increase its performance. Addressing these issues would not only improve the accuracy of the system but also its reliability and depth in processing and responding to complex and nuanced queries, moving closer to the goal of competing at a human champion level in real-time environments like Jeopardy.

Using ChatGPT to improve results

We took an example question such as "Don Knotts took over from Norman Fell as the resident landlord on this sitcom", for which the program initially answered wrongly with "Don Knotts". We took the top 5 documents, which were: <ANSWER0: Don Knotts, ANSWER1: San Pedro, Laguna, ANSWER2: Three's Company, ANSWER3: William de Beauchamp (1185), ANSWER4: Walter Knott> and we fed the content of each document to ChatGPT, asking it which document, in its opinion, best fits the clue. The AI was able to determine the correct answer, "Three's Company", which indicates that the program can be improved by using AI and neural nets in the final stages of it.

Knott's Berry Farm until the death of Cordelia in 1974, at which point he turned his attention toward political causes, Kooiman, Helen, Walter Knott: Keeper of the Flame, pp. 171–84, Plycon Press, Fullerton, CA, 1973. Christiane Victoria, Cordelia Knott: Pioneering Business Woman, pp. 75–78, The Literature Connection E Buena Park, CA, 2009. leaving day-to-day park operations to his children. He died in Buena Park, CA, and was buried at Loma Vista Cemetery in Fullerton, California. The Knott family today no longer owns the theme park, which has since been sold to the Cedar Fair Entertainment Company. Additionally, The J.M. Smucker Co. now carries the "Knott's Berry Farm" brand of jam and jelly (purchased from ConAgra Foods in 2008). Merritt, Christopher Lynxwiler, J. Eric, Knott's Preserved: From Boysenberry to Theme Park, the History of Knott's Berry Farm, pp. 154–60, Angel City Press, Santa Monica, CA, 2010.

>>>

You have here 5 documents. Choose the document which best fits this clue: <<<Don Knotts took over from Norman Fell as the resident landlord on this sitcom>>>



ChatGPT

The document that best fits the clue "Don Knotts took over from Norman Fell as the resident landlord on this sitcom" is document 3, which discusses "Three's Company." This document describes Don Knotts joining the cast as the roommates' new landlord Ralph Furley, taking over after the original landlords, Helen Roper and her husband Stanley Roper, left the show for their own spin-off series.



4. Conclusions

Our experiment shows that even a simple bag-of-words approach can lead to good results. Taking the error analysis into account, if we were to further refine our indexing and searching methods we would consider implementing a better document parser that also keeps track of the links between related documents and finding a better way of selecting the most relevant words in a question.