

Jeopardy! Topic Analyzer Documentation

Executive Summary

The "Jeopardy! Topic Analyzer" is a Shiny web application designed to predict and classify the topics of a user-provided Jeopardy question. Leveraging a machine learning model trained on a diverse dataset of Jeopardy questions, the application assists users in categorizing questions into predefined topics. This documentation provides an overview of the application's functionality, details on its implementation, and comprehensive instructions for installation and usage.

Note: To access the current version of this app, simply visit this link-
https://bflag.shinyapps.io/jeopardywebapp_bflagfinalproject/

A video tutorial can be found here: <https://youtu.be/DyDrIWcJyl0>

Installation and configuration of this software is not necessary to access the app, but detailed instructions have been provided for any additions/extensions one may want to develop.

1. Overview

1.1 Purpose

The primary purpose of the Jeopardy! Topic Analyzer is to automate the categorization of Jeopardy questions based on their content. Users input a question into the web application, and the underlying machine learning model predicts and displays the likely topic of the question. The application is intended for enthusiasts, researchers, and educators interested in exploring patterns and topics within the Jeopardy question dataset.

1.2 Key Features

- Predicts Jeopardy question topics using a machine learning model.
- Provides model accuracy and F1 score as additional insights.
- Incorporates a user-friendly interface.

2. Implementation Details

2.1 Preprocessing Functions

`preprocess_text(text)`

- Removes unnecessary characters and formats the text.
- Converts the text to lowercase.
- Performs lemmatization, contraction fixing, and stopword removal.

`vectorize_text(text)`

- Tokenizes the input text into unigrams and bigrams.
- Combines unigrams and bigrams into a vectorized representation.

2.2 LDA Topic Modeling

- Used the scikit-learn implementation of LDA
- Questions were vectorized using top 2000 tokens in entire corpus of questions
 - Each token was either a unigram or a bigram. R
 - Result was a sparse document term matrix
 - Matrix was fed into LDA algorithm for topic discovery
- Each question was labeled with topic of highest likelihood
- Used ChatGPT to summarize topics into human-comprehensible topics
 - ChatGPT summarized top 100 tokens from each topic into concise phrases
- Phrases were used to label each question in the dataset
- Labeled dataset was exported and used to train Naive Bayes Classifier

2.3 Shiny App UI

- Developed using the Shiny framework.
- Sidebar with a text input for entering Jeopardy questions.
- "Analyze" button triggers the analysis.
- Main panel displays predicted topic, model accuracy, and F1 score.
- Loading spinners for improved user experience.

2.4 Server Functions

`predict_topic(vectorized_text)`

- Placeholder function representing the actual call to the predictive model.
- Returns a numeric prediction.

server Function

- Defines reactive outputs for displaying predicted topic, model accuracy, and F1 score.

2.5 Naive Bayes Model

- Used reticulate package to import necessary Python modules (pandas and scikit-learn)
- Jeopardy dataset split into training and test datasets
- Questions in both training and test datasets vectorized using CountVectorizer
- Multinomial Naive Bayes used for classification model (alpha = 10)
- Accuracy and F1 were both around 77%

3. Software Usage

3.1 Installation

- Ensure R is installed on your machine.
- Install required packages using provided code.
- Load necessary libraries.

3.2 Running the App Locally

- Execute the code in an R environment (ensure any files referenced in the code are available in your working directory).
- The Shiny app will launch in a web browser.
- Enter a Jeopardy question, click "Analyze," and view results.

3.3 Deploying on Shinyapps.io

- Create a Shinyapps.io account and obtain token and secret.
- On the R console, use `rsconnect::setAccountInfo()` to register the account.
- Update deployment code with your app's directory.
- Run deployment code on R console to deploy the app on Shinyapps.io.
- Share the provided public URL for access.

Contributions

Ella Lingafelter:

- Web app development
- Administrative/leader duties

Aashish Anand:

- Naive Bayes Model development
- Web app development + deployment
- Troubleshooting

John Biggan:

- Data cleaning and preprocessing
- Model development assistance
- Presentation assistance
- Troubleshooting

John Fox:

- LDA topic modeling
- Data cleaning and preprocessing
- Topic labeling with ChatGPT

Kush Gupta:

- Video tutorial
- Topic modeling