See The Visualization Here:

👉 Click Here

Documentation for Text Analysis and Visualization Pipeline

This R script here performs various text analysis and visualization tasks on interview responses in a dataset. The primary tasks include sentiment analysis, text preprocessing, word frequency analysis, and emotional sentiment analysis, followed by visualizations of the results. Code for creating the dashboard is here. Below is an overview of the steps in the script used to do performs various text analysis:

Step-by-Step Breakdown

1. Load Required Libraries

The script begins by loading several R libraries necessary for text processing, sentiment analysis, and visualization:

- `tidyverse`: Collection of R packages for data manipulation and visualization.
- `Rcpp`, `textdata`, `tidytext`, `tm`: Libraries for text analysis, including tokenization and text cleaning.
- `plotly`, `highcharter`, `ggwordcloud`, `wordcloud2`: Used for generating interactive and static visualizations of text data.
- `viridis`, `viridisLite`: For creating color palettes for charts.
- `igraph`, `ggraph`, `visNetwork`: For network-based visualizations and word association analysis.
- `syuzhet`, `qdapRegex`, `base64enc`: Libraries for sentiment analysis, regex-based text cleaning, and encoding.

2. Data Import

- The data is imported from an Excel file, `biogasoutcomesmalawi.xlsx`. The data is assumed to contain interview responses in a column, with associated interview dates.

3. Data Transformation

- The interview date is converted to a `Date` type.

- The `interviewee` column is tokenized using the `unnest_tokens` function to split text into individual words. This is done for sentiment analysis and word frequency counting.

4. Sentiment Analysis

- **Bing Sentiment Lexicon**: The script uses the `bing` lexicon to classify words into positive or negative sentiments. The counts of these sentiments are calculated and visualized in a pie chart using `highcharter`.
- **AFINN Sentiment Lexicon**: An additional sentiment lexicon (`afinn`) is accessed, although it's not explicitly used for further visualization.

5. Tokenization and Word Frequency Count

- The script tokenizes the `interviewee` responses using `unnest_tokens`. The resulting tokens (words) are counted to determine the frequency of each word, which is used for word frequency analysis.

6. Text Preprocessing (Custom Function)

- A custom text processing function, `Textprocessing`, is defined to clean the data by:
    - Removing URLs, special characters, digits, punctuation, and extra whitespaces.
    - Removing specific text patterns like ****interviewer**** and square-bracketed content.
- A **corpus** (`myCorpus`) is created using the interview responses, and the text preprocessing function is applied to clean the data further.

7. Term-Document Matrix (TDM) Creation

- A Term-Document Matrix (`TDM`) is created from the cleaned corpus, and it is converted to a matrix format for easy inspection. The matrix provides a count of word occurrences in the document.

8. Most Frequent Words (MFW) Analysis

- The frequency of words is calculated from the TDM, and the top 40 most frequent words are identified.
- A bar chart of the top 40 words is generated using the `highcharter` library.

9. Positive and Negative Word Analysis

- **Top 40 Positive Words**: Words associated with positive sentiments are filtered and counted. A bar chart is then created to visualize the most frequent positive words.
- **Top 40 Negative Words**: Similarly, words associated with negative sentiments are filtered, counted, and visualized in a bar chart.

10. Emotional Sentiment Analysis (NRC Lexicon)

- The `NRC` lexicon is used to classify words into emotions like joy, sadness, fear, anger, etc. The script generates emotional sentiment scores for the entire corpus, focusing on emotions beyond positive and negative.
- A bar chart visualizing the percentage of different emotions is created to show the distribution of emotions in the responses.

11. Word Association

- **Network Visualization**: The script creates a visualization of word associations. The words that appear frequently together in the responses are linked, showing how they are contextually related. This approach can highlight key themes or concepts that are commonly discussed together in the dataset.

Key Visualizations

1. **Sentiment Pie Chart**: Displays the proportion of positive and negative sentiments based on the Bing lexicon.
2. **Top 40 Most Frequent Words**: A bar chart showing the 40 most common words in the corpus.
3. **Top 40 Positive and Negative Words**: Separate bar charts for the most frequent positive and negative words in the dataset.
4. **Emotion Distribution**: A bar chart showing the percentage distribution of different emotions in the text.
5. **Word Association**: A graphical chart showing relationships between frequently co-occurring words.

Conclusion

This script effectively combines several text mining and sentiment analysis techniques to explore a dataset of interview responses. The key steps include preprocessing the data, analyzing sentiment using different lexicons, calculating word frequencies, and visualizing the results through interactive charts. Additionally, word association analysis offers valuable insights into how different terms relate to each other, which can aid in understanding common themes and the overall emotional tone of the responses.