Sure, I can create a documentation for the News Article Analysis and Entity Extraction Streamlit application. Documentation is essential for any software project, as it helps users understand how to use the application and developers understand how the code works.

Here's a suggested structure for the documentation:

# News Article Analysis and Entity Extraction

This is a Streamlit application that performs text preprocessing, named entity recognition, and topic modeling on news articles. It provides a user-friendly interface for analyzing news article text or uploading text files containing news articles.

Installation

To run this application locally, follow these steps:

1. Clone the repository or download the source code.

2. Install the required Python packages by running `pip install -r requirements.txt` in your terminal or command prompt.

3. Download the `en\_core\_web\_sm` spaCy model by running `python -m spacy download en\_core\_web\_sm`.

Usage

1. Navigate to the project directory in your terminal or command prompt.

2. Run the Streamlit app using the command `streamlit run app.py`.

3. The application will open in your default web browser.

4. Choose one of the two options: "Text Input" or "File Upload".

5. For "Text Input", enter the news article text in the provided text area and click the "Analyze" button.

6. For "File Upload", click the "Upload a news article file" button and select a text file containing the news article.

7. The application will preprocess the text, extract named entities, and perform topic modeling, displaying the results on the page.

Features

- Text Input: Users can enter news article text directly into a text area for analysis.

- File Upload: Users can upload text files containing news articles for analysis.

- Text Preprocessing: The application removes HTML tags, tokenizes the text, removes stopwords, and stems the tokens.

- Named Entity Recognition (NER): The application uses spaCy's pre-trained English language model to identify and extract named entities (e.g., people, organizations, locations) from the text.

- Topic Modeling: The application employs Gensim's Latent Dirichlet Allocation (LDA) algorithm to identify the main topics covered in the news article.

- Results Display: The extracted entities and identified topics are displayed on the page for easy analysis.

Code Structure

- `app.py`: This file contains the main Streamlit application code, including the user interface, text preprocessing, named entity recognition, and topic modeling functions.

- `style.css`: This file contains custom CSS styles to improve the appearance of the Streamlit application.