README - DETAILED EXPLANATION

The current project is designed to work starting from a text provided as output by a speech-to-text model. The goal is to divide this text into multiple parts of uniform length and then extract keywords and topics (from a predefined set) for each of them. Typically, speech-to-text models provide output without punctuation: for this reason, the first necessary operation is to add it. To accomplish this, an ad hoc model based on transformers has been chosen.

The text is then divided into chunks taking into account the introduced punctuation. Keywords and topics are then sought within each chunk. Keywords are searched for by comparing each word in a specially created list with those present in the text. To account for the different forms in which a word may appear, the comparison actually occurs between the stemmed versions of the two words. A dictionary mechanism then associates each keyword with its respective topic (group).

Inside the zip folder there are the following files:

- **app.py:** contains the implementation of the Streamlit interface;
- **keywords elaborated.csv**: presents keywords in a table format, along with their stemmed version and associated topic (group);
- **keywordsExtractor.py:** implementation of the keyword extraction mechanism;
- **puntuactionCorrector.py:** API call to the Huggingface model used for punctuation correction:
- **text_to_sentences.py:** divides the text into chunks of uniform length;
- **requirements.txt**

The project is ready to run on Streamlit, but is designed to extract from it, if necessary, the individual parts that compose it.