

Automatic Text Summarization using Deep Learning

Automatic text summarization is the data science problem of creating a short, accurate, and fluent summary from a longer document.

Summarization methods are greatly needed to consume the ever-growing amount of text data available online. In essence, summarization is meant to help us consume relevant information faster.

Summarization is the task of condensing a piece of text to a shorter version, reducing the size of the initial text while at the same time preserving key informational elements and the meaning of content. Since manual text summarization is a time expensive and generally laborious task, the automatization of the task is gaining increasing popularity and therefore constitutes a strong motivation for academic research.

In general, there are two different approaches for automatic summarization: **extraction** and **abstraction**.

- Extractive summarization picks up sentences directly from the document based on a scoring function to form a coherent summary. This method work by identifying important sections of the text cropping out and stitch together portions of the content to produce a condensed version.
- Abstractive summarization methods aim at producing summary by interpreting the text using advanced natural language techniques in order to generate a new shorter text — parts of which may not appear as part of the original document, that conveys the most

critical information from the original text, requiring rephrasing sentences and incorporating information from full text to generate summaries such as a human-written abstract usually does.

Here in this model I have implemented ***Extractive Summarization***.

Libraries Imported:-

- ❖ Numpy
- ❖ Panda
- ❖ NLTK (Natural Language Toolkit)
- ❖ Scikit-learn

Key Steps involved:-

- Imported the required libraries.
- Read the dataset as a csv file using Pandas.
- Tokenized text into sequences stored in a list.
- Used GloVe Word Embeddings (GloVe word embeddings are vector representation of words. These word embeddings will be used to create vectors for our sentences.).
- Extracted all the Word vectors.
- Did a few text pre-processing like removing unnecessary punctuations,etc.
- Removed stopwords from sentences (Stopwords are words unnecessary for being included in the summary).
- Used cosine similarity approach by creating an empty similarity matrix for this task and populate it with cosine similarities of the sentences.
- Applied Page Rank Algorithm where the nodes of this graph will represent the sentences and the edges will represent the similarity scores between the sentences.

Download [GloVe Word Embedding](#) files before running the model in your system.