# Extractive vs Abstractive Summarization

Text summarization is the process of reducing the length of a text document by recomposing or resequencing sentences. The process of summarization itself is defined as reducing the length of a document to at least half of its original length while keeping the contextual and semantic meaning intact. I've analyzed and compared two different approaches to summarization: the extractive and the abstractive approach.

The abstractive approach hopes to reconstruct sentences in a certain way to shorten the document and preserve the meaning, while being syntactically consistent. The underlying models most of the time contain a sequence2sequence model, sometimes paired with a CNN or a similar supporting structure to help overcome its shortcomings: repetition, and semantic irrelevance.

The extractive approach is based on splitting the document into multiple sentences and choosing our pick of the most important, representative sentences to convey all the main points of the original document. My current implementation is based on Ozsoy et al. which can be found here.

The main idea of the extractive method is to create some kind of algebraic, transitionary representation for the input text, where each sentence is given a vector representation. Then this representation is used to mathematically deduce the list of the k most important sentences where k is a user supplied parameter.

The exact method recommended by Ozsoy et al. is dubbed the "cross-method" which consists of a pre-processing step on the SVD of the input matrix and a modified version of the selection process presented in Steinberger and Jezek. The discussion of the results of this method suggests that the cross-method performs better than its predecessors when working on sufficiently large texts, but it lacks good results in shorter pieces. It also lacks behind graph

based approaches like SentenceRank. My reproduction of this method has agreed with the findings of the paper, where short texts -especially short texts with long, running sentences- are problematic examples. The user-given shrinking ratio causes the algorithm to pick only a single sentence in certain cases which forcefully causes the summary to lack some important points from the document.

The abstractive methods provide better accuracy and better results, however due to their nature of being supervised models they -sometimes- require large amounts of data to reach their target accuracy. We should also consider the fact that since most abstractive methods are based on seq2seq models, which carry a smaller memory footprint compared to traditional MT models which have to hold phrase tables.

On the other hand the extractive implementations , while being easier to implement and unsupervised, put out summaries that sometimes clearly lack information. In addition to this shortcoming they also lack the capability of combining information from multiple sentences when the input document is relatively short.

I believe the extractive method has proven to be useful in certain aspects of daily life, like the summarization of long news articles or scientific papers. However, they do not seem capable of providing query-based summaries or summaries on shorter documents. Abstractive methods seem to be the stronger alternative, but still presents a higher barrier of entry due to large data requirements and the model training process. I believe more accessible abstractive methods or a combination of a graph based extractive method with an abstractive method might provide different overall behavior which could prove to be interesting.

Sources:

https://medium.com/luisfredgs/automatic-text-summarization-with-machine-learning-an-overview-68ded5717a25

https://towardsdatascience.com/abstractive-summarization-using-pytorch-f5063e67510

https://www.youtube.com/watch?v=MqugtGD605k

Github: https://www.github.com/CaglarDeniz/datalab