Comparison of Different Text-Summarization Techniques

Venkata Abhiram Chitty

vchitty@students.kennesaw.edu

Institution: Kennesaw State University

Abstract:
In today's digital world, much information is created daily. Reading, gathering, and summarising material can become laborious and prone to human error. Text summarization is an increasingly important Machine Learning task as it becomes a common way to determine relevant information from many documents. As text summarization has increased in utilization, multiple methods have been developed for this task. We will be using three different abstractive methods, Term Frequency – Inverse Document Frequency, Transformers, and Sequence-to-Sequence, for text summarization and comparing the performance of each modelling type through the similarity of each model's output to an existing summary of the corpus.

Index Terms: Text Summarization, Term Frequency-Inverse Document Frequency (TF-IDF), Natural Language Processing, Encoding, Decoding, sequence-to-sequence (Seq2Seq) model and Transformers.

Introduction:
With the growth of the internet in the past decade, we have never had more information at our fingertips, and as a result, there's a growing need for summarizations of all of this information. This has become particularly apparent in the news field as more and more people utilize social media platforms and newsfeeds to obtain their news. As a result, what previously was a lengthy and expensive process, reading articles and summarizing them effectively, is now near impossible with the sheer quantity of news articles being published regularly. Natural Language Processing (NLP) converts human language into information that computers can process for analysis. Text summarization is the NLP task of summarizing text in a few sentences that capture the concepts present in the input text. The two general ways of doing text summarization are extractive, where important words are taken from an input text and combined into a coherent summary, and abstractive, where a model is trained to generate summaries like a human would. We will overview the multiple methods for accomplishing these tasks in the following section.

Related Work:
Text is a complicated data structure to analyze. As a result, many different approaches to text summarization have been developed. The two overarching types of summarizations are extractive and abstractive summarization, where extractive summarization selects the most relevant words from a document to form the summary, and abstractive summarization is a unique sentence or set of sentences generated by a model that summarizes an inputted set of text. An early attempt of NLP was the introduction of Term Frequency-Inverse Document Frequency (TF-IDF) which weights words based on the frequency of each word and the inverse frequency of that word across multiple documents (Li, 2002). This form of extractive text summarization focuses on frequency to determine the importance of text to summarize a larger document. While the focus of this paper is on abstractive summarization, as abstractive summarization has proven to be an effective form of text summarization that also provides meaningful and human-readable output, we will be utilizing TF-IDF as a baseline form of extractive text summarization to compare our abstractive methods to.

A particular difficulty in text summarization is training human semantic knowledge of text into the model (Shanmugam, 2023). The solution to this problem is utilizing the attentional recurrent neural networks encoder-decoder model, which attempts to map single-word sequences to another for the summarization task (Nallapati, 2016). This modelling type was first utilized for machine translation but saw promising results when utilized for text summarization. While only some models performed excellently, this innovative modelling technique generally showed it could generate summaries that, while not perfect, could be indistinguishable from a human-written summary. This model type is called a sequence-to-sequence (S2S) model due to the direct mapping of words from one sequence to a new sequence. To further improve summarization modelling, Long Short-Term Memory-Convolutional Neural Networks (LSTM-CNN) were added to the S2S framework, allowing models to improve in how words were mapped by allowing models to have prior words used as inputs for the prediction of upcoming words. So, instead of only one word being mapped at a time to a predicted outcome word, that word and any before it is used to predict the next outcome word, some researchers have compared the effectiveness of LSTM models to that of Large Language Models (LLMs) such as T5, BART, and BART-Large and found that comparatively LSTM performs well in recall, but not as well in precision and F-measure (Öykü, 2023).

Other text summarization methods have also been developed, such as using Latent Dirichlet Allocation for topic modelling (Onah, 2022) and Text-Representing Centroids (Ruamsuk, 2022), which are not the focus of this study, but have shown promising results for this field.

Dataset:
The dataset contains a total of 11,491 rows. Each row has two columns: article and highlight. The article column includes the text content of online articles covering various topics such as sports, entertainment and technology. The texts in this column consist of vocabulary, language structures, numeric statistics, and mentions of specific people and places. On the other hand, the highlighted column provides sentences summarising each article's main points. It focuses on the aspects, including statistics and important entities mentioned in the article, while omitting minor details, elaborations and examples.

Data Preprocessing:

1. Handling missing values:
   Insert empty strings '' into any missing article or highlight value. As a result, there are no nulls, which might interfere with further processing.

2. Change to lowercase:
   Use the Lower () function to convert all text to lowercase characters. This lowers the normalization and amount of vocabulary.

3. Take the numbers out:
   For text, use regular expressions re. Sub (r'\d+', '') to eliminate all numbers from 0 to 9. To summarise, numbers don't have any meaningful semantic value.

4. Take the punctuation out:
   To create a translation table, punctuation marks are eliminated using. translate and make trans. Since punctuation adds no semantic significance, it can be omitted.

5. Take out any extra whitespace:
   Any leading or trailing whitespace is removed using .strip() so that only words remain.

6. Tokenize the text into words:
   To break phrases up into individual words, use the tokenizer word_tokenize function. Tokens provide further processing.

7. Keep only alphabetic words:
   Conditional list comprehension excludes non-alphabetic words like punctuation and digits, leaving only words that start with an alphabet.

8. Remove stopwords:
   Since stopwords from the NLTK library don't contain any useful information, they are filtered out using set differences.

9. Lemmatize terms:
   WordNetLemmatizer is used to lemmatize words to their most basic root form to minimize the amount of vocabulary.

10. Join lists to string:
    The list of tokens is joined back to strings using " ". This produces cleaned text for feeding to models.

Methodology:

For this summarization we have used three different models namely Term Frequency - Inverse Document Frequency (TF-IDF), Sequence2Sequence and Transformers. We have tested these three different models for our dataset.

TF-IDF Algorithm:

TF-IDF Model Term Frequency-Inverse Document Frequency (TF-IDF) is an extractive summarization technique. This is a technique to quantify words in a set of documents. Text vectorization converts words within a text document into important numbers. Tf-IDF measures a term's importance within a document about a collection of documents. This algorithm has two components – Term Frequency and Inverse Document Frequency.

The Term Frequency (TF) is computed as the number of times a term appears in a document relative to the total number of words.

$$TF = \frac{No.of\ times\ the\ term\ appers\ in\ the\ document}{Total\ no\ of\ terms\ in\ the\ document}$$

The Inverse Document Frequency shows how many documents in the corpus contain that term. Words specific to a small subset of documents are given a higher importance value than words used in all documents. The formula for calculating the value is as follows:

$$IDF = Log(\frac{No\ of\ documents\ in\ the\ text}{No\ of\ documents\ in\ the\ text\ contain\ the\ term})$$

The TF and IDF scores are multiplied to determine the term's TF-IDF.

TF-IDF = TF * IDF

Sequence-to-Sequence (Seq2Seq) Model:

The Seq2Seq model is a neural network architecture consisting of an encoder and a decoder. The encoder processes the input sequence (news articles), and the decoder generates the corresponding summary. Our Seq2Seq model, equipped with multiple Long Short-Term Memory (LSTM) layers, captures sequential dependencies. Before training, we prepared the data by tokenizing and padding text and summary sequences. A rare word analysis guided vocabulary decision, enhancing the model's understanding. During training, we split the data, engaging in three epochs with early stopping to mitigate overfitting. Evaluation through learning curve plots showcased the model's convergence and performance on training and validation sets. Training loss demonstrated a steady decline, indicating effective learning, while validation loss remained stable, signifying generalization.

Transformers:
Text summarization using transformers is a process where advanced machine learning models, specifically transformer-based models, condense longer texts into shorter, coherent summaries. Here is a brief overview of how it works:

1. Transformers Architecture:
   A particular kind of deep learning model called a transformer was first described in the 2017 study "Attention Is All You Need" by Vaswani et al. They are particularly well-suited for natural language processing tasks, including text summarization.

Unlike previous models that processed text sequentially, transformers use an attention mechanism to weigh the significance of different words in a sentence, enabling parallel processing and capturing context more effectively.

2. Process of Summarization:
    - Input: The model receives a long text as input.
    - Understanding Context: Using self-attention mechanisms, the transformer analyzes the entire text, understanding context, semantics, and the relationships between words and phrases.
    - Generating Summary: The model then generates a summary. This can be done in two ways:
        - Extractive Summarization: Selecting key sentences or phrases directly from the text and stitching them together to create a summary.
        - Abstractive Summarization: creating new sentences—often by rewording or substituting words—that encapsulate the main ideas of the source material.

3. Pre-trained Models: Several pre-trained transformer models are available for text summarization, such as BERT, GPT (like OpenAI's GPT-3), and T5. These models can produce precise and well-organized summaries because they have been trained on enormous volumes of text data.

4. Fine-tuning for Specific Needs: These models can be further fine-tuned on specific types of text (like scientific articles, news stories, etc.) to improve their accuracy in those domains.

5. Challenges:
    - Coherent and maintains the original context and meaning.
    - Handling Lengthy Documents: Longer documents might pose a challenge regarding memory and processing requirements.
    - Bias and Sensitivity: Ensuring the summary does not introduce or propagate biases in the training data.

6. Applications: Text summarization is used in various fields such as news aggregation, legal document analysis, academic research, and more.

In conclusion, text summarization using transformers signifies a noteworthy development in natural language processing., offering a way to efficiently condense large volumes of text while retaining key information and context.

We used T5 as a pre-trained transformer model for our text summarization.

Result and Discussion:
The term frequency-inverse document frequency, or TF-IDF, algorithm is a straightforward extractive summarization technique that is surprisingly efficient. It consistently covers subjects indicative of the entire document by choosing sentences frequently containing keywords. That said, these sentences are not guaranteed to make sense when combined. Thus, readability is worse than in summaries that humans write.
Sequence-to-sequence models based on long short-term memory (LSTM) show promise for generating more fluent abstractive summaries. Their performance is heavily dependent on training with a large in-domain dataset, though, and many hyperparameters need to be adjusted. LSTM performance would only fall well short of human baselines with these fine-tunings.

Table 1: Comparison of Methods			
	TF-IDF	LSTM	Transformer
Rouge1	0.16304636	0.016471161	0.293877551
Rouge2	0.050668204	0.001097394	0.271604938
RougeL	0.158332693	0.013428105	0.285714286
RougeLsum	0.160542834	0.013573511	0.285714286

Transformer-based models provide strong out-of-the-box performance with minimal parameter tweaking, using self-attention to identify important content. Further fine-tuning is helpful, but strong pretraining methods like BERT allow for a reasonable summarization ability to emerge quickly. Less control over the specifics of the summarization policy is the trade-off.

Conclusion:
TF-IDF produced fragmented summaries with poor flow when various summarization strategies were tested on an article and highlights dataset. However, it offered representative coverage through keyword matching. Following extensive fine-tuning on in-domain training data, Seq2Seq models demonstrated strong performance close to human fluency and coherence. In the meantime, the pre-trained transformer model showed decent readability right out of the box and could identify important content without needing further optimization.

Recommendation:
Though significant configuration and training data are required to fully realize the potential of the Seq2Seq approach, it is advised for applications where fluency is paramount, such as replicating human summaries. The zero-shot transformer model can generate reasonably coherent summaries more quickly for quick and flexible content summarization across various corpora. This better aligns it with scenarios that optimize deployment speed at the expense of some linguistic quality degradation compared to human performance on the task. It all boils down to balancing available resources, performance requirements, and time constraints. However, a comparative analysis using representative data identifies the advantages and disadvantages to help choose the approach that best fulfils the needs of the target summarization use case.

References:

Abdelrahman, Amany A., et al. "Extractive text summarization of long documents using word and sentence encoding." *2023 5th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, 2023, https://doi.org/10.1109/niles59815.2023.10296690.

K, Manojkumar V, et al. "An experimental investigation on unsupervised text summarization for customer reviews." *Procedia Computer Science*, vol. 218, 2023, pp. 1692–1701, https://doi.org/10.1016/j.procs.2023.01.147.

Li, Ping Jing, et al. "Improved feature selection approach TFIDF in text mining." *Proceedings. International Conference on Machine Learning and Cybernetics*, 4 Nov. 2002, https://doi.org/10.1109/icmlc.2002.1174522.

Mercan, Öykü Berfin, et al. "Abstractive text summarization for resumes with cutting edge NLP transformers and LSTM." *2023 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2023, https://doi.org/10.1109/asyu58738.2023.10296563.

Mohammad Masum, Abu Kaisar, et al. "Abstractive method of text summarization with sequence to sequence RNNS." *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 6 July 2019, https://doi.org/10.1109/icccnt45670.2019.8944620.

Nallapati, Ramesh, et al. "Abstractive text summarization using sequence-to-sequence RNNS and beyond." *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 19 Feb. 2016, https://doi.org/10.18653/v1/k16-1028.

Onah, Daniel F., et al. "A data-driven latent semantic analysis for automatic text summarization using LDA Topic Modelling." *2022 IEEE International Conference on Big Data (Big Data)*, 2022, https://doi.org/10.1109/bigdata55660.2022.10020259.

Ruamsuk, Yanakorn, et al. "Generating and evaluating text summarisations using text-representing Centroids(TRC)." *2022 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C)*, 2022, https://doi.org/10.1109/ri2c56397.2022.9910272.

Shanmugam S, P., et al. "Abstractive text summarisation using keywords with transformers model." *2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, 2023, https://doi.org/10.1109/iconstem56934.2023.10142867.

Song, Shengli, et al. "Abstractive text summarization using LSTM-CNN based Deep Learning." *Multimedia Tools and Applications*, vol. 78, no. 1, 2018, pp. 857–875, https://doi.org/10.1007/s11042-018-5749-3.