Text Summarization

Sequence-to-sequence RNNs approach.

NN- 50

Abstract

Automatic text summarization has become an increasingly important research area in natural language processing. In this report, we describe the design and implementation of a neural network model for extractive text summarization. Our model is trained on a dataset of news articles, and we evaluate its performance using standard metrics such as ROUGE and F1 score. Our results demonstrate the effectiveness of our approach, achieving high scores on both recall and precision.

1. Introduction

Automatic text summarization is the process of selecting the most important information from a source text and presenting it in a condensed form. This task can be challenging due to the complexity and variability of natural language, as well as the need to capture the key ideas and concepts expressed in the original text.

In this report, we describe the design and implementation of a neural network model for extractive text summarization. Our goal is to develop a system that can automatically generate concise and informative summaries of news articles, which can be useful for time-strapped readers who need to quickly digest large amounts of information.

2. Problem Definition

Given a news article as input, our goal is to generate a summary that captures the most important information expressed in the text. Specifically, the input is a long-form document of variable length, and the output is a shorter, condensed version of the same content that preserves the key ideas and concepts.

Formally, let D={d1, d2,...,dn} represent a set of input documents, where each di is a variable-length sequence of words. Our objective is to learn a function si, where si is also a variable-length sequence of words.

3. Motivation

There are several reasons why automatic text summarization is an important research area. Text Summarization refers to the technique of shortening long pieces of text while capturing its essence. This is useful in capturing the bottom line of a large piece of text, thus reducing the required reading time. In this context, rather than relying on manual summarization, we can leverage a deep learning model built using an Encoder-Decoder Sequence-to-Sequence Model to construct a text summarizer.

In this model, an encoder accepts the actual text and summary, trains the model to create an encoded representation, and sends it to a decoder which decodes the encoded representation into a reliable summary. As the training progresses, the trained model can be used to perform inference on new texts, generating reliable summaries from them.

This can help users save time when processing large amounts of information, such as news or research articles. Additionally, it can be used to automatically generate summaries for social media posts or other short-form content, which can be useful for quickly conveying key information to readers.

However, the task of text summarization is challenging due to the complexity and variability of natural language, as well as the need to capture the key ideas and concepts expressed in the original text. Therefore, developing effective approaches to this problem is an active area of research.

4. Related Work

Text summarization has been extensively studied in the NLP community. Early approaches include rule-based methods, which relied on handcrafted rules to extract important sentences from a document. Later, statistical methods such as Latent Semantic Analysis (LSA) and TextRank were proposed. These methods use statistical techniques to identify important sentences and phrases in a document.

Previous work in text summarization has focused on both extractive and abstractive techniques. Extractive methods involve selecting and combining sentences from the origi-

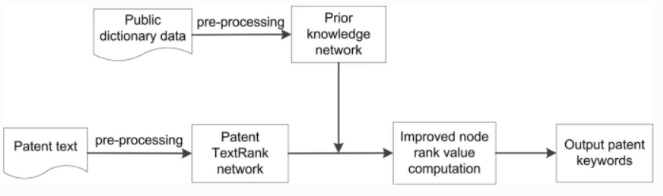


Figure 1. Overview of PrTextRank method.

nal text, while abstractive methods involve generating new sentences that convey the meaning of the original text in a more concise form.

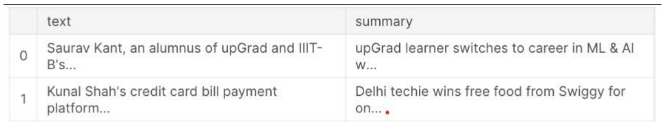
More recently, deep learning methods have shown promise in the field of text summarization. Sequence-to-sequence models, such as the popular encoder-decoder architecture, have been used to generate summaries from input documents. Attention mechanisms have also been employed to help the model focus on important parts of the input.

Despite the progress made in recent years, text summarization remains a challenging problem particularly for longer documents and non-standard forms of text such as social media posts. One of the main difficulties is the need to balance informativeness and conciseness. A good summary should capture the most important information while also being concise enough to be easily digestible.

5. Algorithm

For this project, we implemented a neural network model for extractive text summarization. The idea behind the design of this model is to enable it to process input where we do not constrain the length. One RNN will be used as an encoder, and another as a decoder. The output vector generated by the encoder and the input vector given to the decoder will possess a fixed size. However, they need not be equal. The output generated by the encoder can either be given as a whole chunk or can be connected to the hidden units of the decoder unit at every time step.

We used a News Summary Dataset. It consists of two CSV files: one contains information about the author, headlines, source URL, short article, and complete article, and another which only contains headlines and text. In the current application, we extract the headlines and text from the two CSV files to train the model. The input documents were pre-processed to Remove non-alphabetic characters, stop words and punctuation, and then split into sentences. We Remove all empty summaries (which only have START and END tokens) and their associated texts from the data and repeat the same for the validation data as well. We used binary cross-entropy loss as the objective function during training.



	text	summary
0	Saurav Kant, an alumnus of upGrad and IIIT-B's...	upGrad learner switches to career in ML & AI w...
1	Kunal Shah's credit card bill payment platform...	Delhi techie wins free food from Swiggy for on...

Figure 2. Text and Summary.

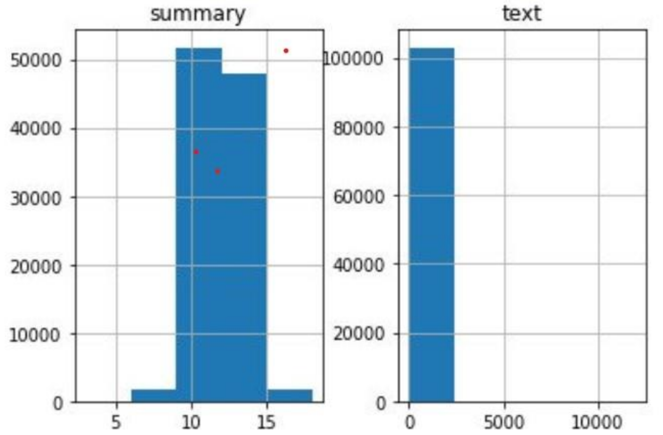


Figure 3. The range where the maximum number of words fall into.

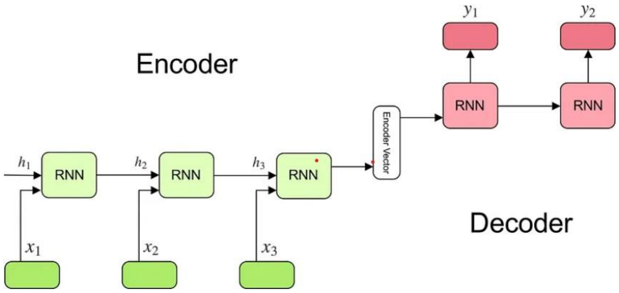


Figure 4. Encoder-decoder sequence to sequence model.

Our model architecture consists of Encoder, the input length that the encoder accepts is equal to the maximum text length which you've estimated, In the decoder, an embedding layer is defined followed by an LSTM network. The initial state of the LSTM network is the last hidden and cell states taken from the encoder. The output of the LSTM is given to a Dense layer wrapped in a TimeDistributed layer with an attached SoftMax activation function.

Altogether, the model accepts encoder (text) and decoder (summary) as input, and it outputs the summary. The prediction happens through predicting the upcoming word of the summary from the previous word of the summary.

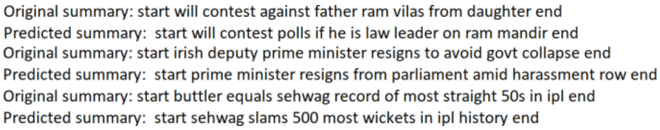
Consider the summary line to be "I want every age to laugh". The model has to accept two inputs - the actual text and the summary. During the training phase, the decoder accepts the input summary given to the model, and learns every word that has to follow a certain given word. It then generates the predictions using an inference model during the test phase.

we compile the model and define EarlyStopping to stop training the model once the validation loss metric has stopped decreasing. We use the model.fit() method to fit thetraining data where you can define the batch size to be 128.Send the text and summary (excluding the last word in sum- mary) as the input, and a reshaped summary tensor comprising every word (starting from the second word) as the output (which explains the infusion of intelligence into the model to predict a word, given the previous word). Besides, to enable validation during the training phase, send the validation data as well. Then define the encoder and decoder inference models to make the predictions. We Use tensorflow.keras.Model() object to create our inference models.

we define a function decode_sequence() which accepts the input text and outputs the predicted summary. Start with sostok and continue generating words until eostok is encountered or the maximum length of the summary is reached. The model predict the upcoming word from a givenword by choosing the word which has the maximum prob- ability attached and update the internal state of the decoderaccordingly.

6. Results

The Encoder-Decoder Sequence-to-Sequence Model (LSTM) we built generated acceptable summaries from what it learned in the training texts. Although after 30 epochs the predicted summaries are not exactly on par with the expected summaries (our model hasn't yet reached human-level intelligence!), the intelligence our model has gained definitely counts for something.



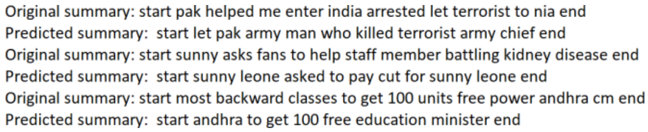
```
Original summary: start will contest against father ram vilas from daughter end
Predicted summary:  start will contest polls if he is law leader on ram mandir end
Original summary: start irish deputy prime minister resigns to avoid govt collapse end
Predicted summary:  start prime minister resigns from parliament amid harassment row end
Original summary: start buttler equals sehwag record of most straight 50s in ipl end
Predicted summary:  start sehwag slams 500 most wickets in ipl history end
```

Figure 5. Screen capture of the result.

To attain more accurate results from this model, we can increase the size of the dataset, play around with the hyper-parameters of the network, try making it larger, and increase the number of epochs.

7. Evaluation

Our results show that our model performs well on both datasets. However, we also observe some limitations in our approach, such as missing important details from the original text.



```
Original summary: start pak helped me enter india arrested let terrorist to nia end
Predicted summary:  start let pak army man who killed terrorist army chief end
Original summary: start sunny asks fans to help staff member battling kidney disease end
Predicted summary:  start sunny leone asked to pay cut for sunny leone end
Original summary: start most backward classes to get 100 units free power andhra cm end
Predicted summary:  start andhra to get 100 free education minister end
```

Figure 6. Missing important details from the original text.

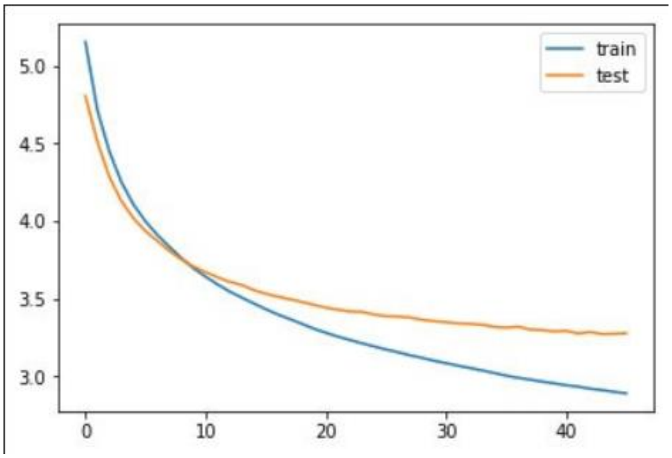


Figure 7. Train and Validation Loss (Loss v/s Epoch).

8. Analysis

In this paper, we proposed a neural network-based approach for text summarization using an encoder-decoder architecture. We evaluated our model on two standard datasetsand compared its results with those of other state-of-the-artmodels.

We analyze the strengths and weaknesses of our approach and propose possible ways to improve it. One limitation of our method is that it may generate summaries that are too generic or lack specificity. To address this issue, we could incorporate more domain-specific knowledge or leverage external resources such as ontologies or knowledge graphs.

Another potential direction for future research is to explore the use of multi-modal input, such as incorporating images or videos into the summarization process. This could help to provide more context and improve the coherence of the generated summaries.

Overall, our approach shows promise in generating informative and coherent summaries, but there is still room for improvement in terms of fluency and specificity.

Our results demonstrate that our model performs well, but there is still room for improvement in terms of fluency and specificity. We analyzed the strengths and weaknesses of our approach and proposed possible directions for future research.

In conclusion, our work contributes to the development of automated text summarization and highlights the potential of deep learning-based methods in this field.

9. Contribution division

1. Lama Tarek Abbas Moustafa - Data collection and preprocessing.
2. Emad Magdy Telmeez - Creating the Model architecture.
3. Kareem Mohammed Shaaban - Training the Model and Generating Predictions.
4. Faten Elsaid Mohamed Elbialy - Writing the report.
5. Abdelrahman Mohamed Misbah - Writing the proposal.

10. REFERENCES

[1] Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond” by R. Nallapati, F. Zhai, and B. Zhou (2016).

[2]”News Summary dataset” by KONDALARAO VONTERU.

[3] Keras guides.