ISSN: 0974-5823 Vol. 6 No. 3 December, 2021

International Journal of Mechanical Engineering

SURVEY AND ANALYSIS OF DIFFERENT TRANSFORMER MODELS FOR ABSTRACTIVE TEXT SUMMARIZATION

Abhijeet R. Raipurkar, Tithi Agrawal, Suhasini Aney, Karen Abraham, Shubham Saboo and Atharva Nimbalwar

Department of Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur

Abstract

Quintillions of data is generated on a daily basis and this surfaces the need for the summarization of this data. Generating precise and fluent summaries of lengthy articles manually is a very strenuous task. Hence automated summaries are needed, two techniques are used for automated summaries- Extractive and abstractive. Extractive summarization uses keywords and important sentences to construct the summary whereas abstractive summarization understands the data to be summarized and presents the summary. This makes it much more complicated.

The encoder decoder architecture is generally used for abstractive text summarization. The selection of encoder-decoder architecture provides us with certain choices of designing our encoder and decoder with standard RNN/ LSTM/ GRU, bidirectional RNN/LSTM/GRU, Transformer, BERT/GPT-2 architecture.

This paper briefs about the different transformer architectures - T5, BART and Pegasus and their functioning. Finally, a comparative analysis of these models when used on the same data is presented. The summaries generated by these models are compared to a manually generated summary and ROUGE1, ROUGE2 and ROUGEL values are weighed.

The purpose of this review on abstractive text summarization is to render a complete understanding of the elements of recent abstractive text summarization models as well as to provide an instinct of the challenges with these systems.

Keywords: Abstractive text summarization, Natural Language Processing, ROUGE, Transformers.

1. INTRODUCTION

1.1. Text Summarization

There is a huge quantity of data which is growing everyday. This data is unorganised, there is an acute need that this data's size is reduced and summarized in a succinct manner. The purpose of automatically producing text summaries is to have the summaries which are on par with human written documents. Data reduction alone is insufficient, the generated summaries must be precise and consistent.

1.2. Why Summarize text?

1. Summaries reduce the time invested in reading.
2. Summaries make it easier to discover information.
3. These algorithms are less partial than a majority of the human summarizers.
4. It also increases the indexing's efficiency.
5. QnA becomes easier as personalized information is provided via summaries.
6. Automatic summarization aids corporations in the processing of large volumes of texts.

1.3. Literature Review

One of the most essential features of Natural Language Processing (NLP) is Text Summarization. So to understand text summarization, it is critical that we comprehend the history of NLP. Machine Translation(MT) is the origin of NLP for translating Russian language into English & vice versa, during the second world war. In the beginning, text summarization was done using rule-based algorithms, called 'importance evaluator', that worked based on ranking parts of a text according to their importance.[1](Allahyari, 2017)

Text summarization was a significant advancement in NLP. A neural network is trained on a corpus of articles and then adjusted using feature fusion to generate a summary of the article's highest-ranking sentences. The NN learns what sentences and phrases must be taken into consideration in the summary. During feature fusion the neural network is trimmed and the hidden layer unit activations are collapsed into discrete values with frequencies. The main traits are then generalised that must be included in the phrases that will build the summary. Ultimately, the modified NN ranks the sentences to determine which ones will be included in the summary. The diversity-based approach in extractive summarizer, calculates sentence diversity and attempts to eliminate repetitive sentences from the final summary.

In 2016, Text summarization using seq2seq model outperformed other models and demonstrated state-of-the-art performance amongst other models, where an attentional encoder-decoder RNN[3](Graves, 2013), which was actually established for machine translation, outperformed other models and demonstrated state-of-the-art performance among other models developed at the time.

In the realm of text summarization, more models were developed which aided in the creation of a more abstractive summarised result. One such model's foundation is standard feed forward Network Neural Language Model (NNLM) which is used to estimate the contextual probability of the following word, also known as next word prediction model.With the introduction of BERT there was a broad range of progress in NLP tasks. BERT introduced pretrained language models that perform as a State-Of-The-Art model in NLP applications using a transfer learning approach. With all of its transfer learning features, BERT has left a lasting impression in the text summarization field. Another recent method of abstractive summarization is PEGASUS[11](Zhang, 2019), which combines gap sentence generation (GSG) and masked language model (MLM) to achieve a state-of-the-art result with a lesser sample size.T5 (Text-to-Text-Transfer-Transformer), a recent Google release, claims to surpass existing high-end algorithms such as BERT, GPT2, and others on NLP tasks like text classification, question answering, text summarization etc.

1.4. Summarization techniques:

1.4.1. Extractive Text Summarization

This revolves around the selection of pieces of sentences from the original document to form a complete new summary. Ranking is done on the basis of relevance of phrases to choose only from the most suitable to the implication of the source.[6](Kalchbrenner, 2013)

1.4.2. Abstractive Text Summarization

Abstractive text summarization, creates new phrases and sentences to capture complete meaning of the data to be summarized. It is used to form a semantic representation of the document. Then words from the general vocabulary which are deemed appropriate are selected to prepare a brief summary that accumulates the crux of document's ideas.[6](Kalchbrenner, 2013)

2. METHODOLOGY:

2.1 ABSTRACTIVE TEXT SUMMARIZATION

2.1.1 Tasks at the core of abstractive summarization approaches:

1. Information extraction extracts needed information from using phrases-they maybe noun or verb phrases. Another method to extract important information is by employing query-based extraction.

2. For the purpose of content selection, a subset of important phrases from the extracted text are selected to include in the resulting summary.

3. The surface realization task combines selected words or phrases in an ordered sequence by using grammatical rules and lexicons (vocabulary along with its related knowledge on linguistic significance and usage).

2.1.2 Three domains of Abstractive Text Summarization:

1. The structure-based approach methods encode data from text documents based on certain arrangements, for example, templates or other structures like trees, ontology, lead and body, rules (classes and lists), and graphs.

2. Semantic-based methods work on identifying noun and verb phrases by applying linguistic/semantic illustration of a text document as an input to the natural language generation system. These systems include multimodal semantic-based techniques, information item-based methods, semantic text representation, and semantic graph methods.

3. There are many advancements in text summarization where deep learning concepts like sequence to sequence models are known to be the foundation of most of the recent studies.

2.2 ENCODER DECODER ARCHITECTURE

The selection of encoder-decoder architecture provides us with certain choices of designing our encoder and decoder with standard RNN/ LSTM/ GRU, bidirectional RNN/LSTM/GRU, Transformer, BERT/GPT-2 architecture, or the very recent BART model.

2.2.1 RECURRENT NEURAL NETWORK (RNN)

- RNN does not have a regular feedforward neural network architecture. There are feedback loops that allow information to persist in these networks. They introduce the concept of memory in neural networks. Owing to their feedback nature, these networks learn information based on the context.[3](Graves, 2013)

- The architecture of RNN suits very well with tasks relating to sequential data. A novel network consisting of two RNNs as encoder and decoder was first proposed for statistical machine translation tasks.[3](Graves, 2013).

2.2.2 LONG SHORT-TERM MEMORY (LSTM)

- It is a very special type of RNN because it solves the problem of long-term dependencies. For example, if the next word in a sequence is to be predicted and the correctly predicted word depends on past information, RNN is not capable of retaining information at length.[5](Hochreiter, 1997)

- This is where (long-term gaps/dependencies) LSTMs come into practice. LSTM can learn information for longer periods.

2.2.3 GATED RECURRENT UNIT (GRU)

- GRU is a variant of LSTM because there is a similarity in the design of both. It tackles the problem of vanishing gradient in recurrent neural networks.The design of GRU has an update gate and a reset gate.

- The update gate deals with information that goes into the memory and helps the model to decide which of the past information needs to be memorized to be passed on.

- The reset gate deals with information that flows out of the memory and helps the model to decide the past information which can be forgotten.

2.2.4 BI-DIRECTIONAL RNN/LSTM/GRU

- Bidirectional neural networks consider two sequences for predicting the output, one in the forward direction and the other in the reverse direction.

- It implies that with bidirectional networks we can make predictions of the current state by using information from previous time steps as well as later time steps. So, the network can capture a richer context and is capable of solving problems more effectively.

2.2.5 TRANSFORMERS

- Transformers were a breakthrough introduced by Google for sequence learning tasks.
- Transformers are based entirely on attention mechanisms thus eliminating the requirement for recurrent as well as convolutional units. The transformer architecture consists of encoders and decoders stacked up.
- The encoder and decoder blocks are made up of attention units and feed-forward units. The encoder part is a stack of six encoder units and the decoder part is a stack of six identical decoder units.
- Each encoder unit has a multi-head attention unit as well as a feedforward unit. Each decoder unit has an additional masked multi-head attention unit in addition to the feedforward unit and the multi-head attention unit.
- The functioning of the transformer starts with the word embeddings of the input sequence. The word embeddings are forwarded to the first encoder which is then transformed and passed on to the following encoder. This is repeated many times until it gives the output.
-

2.2.6 BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMER-GENERATIVE PRE-TRAINED TRANSFORMER (BERT-GPT)

- Google introduced BERT in 2019. It allows the application of a pre-trained language model to a variety of NLP tasks.[2](Devlin, 2018)
- Open AI introduced GPT in 2018, which can be used to pre-train a language model on a large body of text. This can further be fine-tuned on a variety of particular tasks.
- When it comes to architecture, both the models are similar in the sense that both are transformer-based, but the distinguishing factor between the two is that training is unidirectional in GPT whereas BERT can perform bidirectional training.
- Another point of difference is that GPT is a multilayer transformer decoder whereas BERT is a multilayer transformer encoder. GPT-2 has an autoregressive nature, i.e., each token has a context of the previous words, but BERT is not autoregressive and hence, employs all surrounding context at a time.[2](Devlin, 2018)
- BERT comprises two modules namely, pretraining and fine-tuning. It is trained with two tasks known as Masked Language Model (MLM) for bidirectional prediction and Next Sentence Prediction (NSP) for sentence-level understanding.

2.3. Dataset

From here on out, the basic experimental setup is outlined, assessment metrics, and numerous models are analysed. Apart from this, findings from the research will be compared with the models' performance. The dataset is derived from a text categorization dataset, which consists of BBC news website documents referring to articles featured in the paper [4](Greene, 2006)

2.4. Preprocessing

This set of data includes large news stories as well as short summaries for comparison. Following that, the raw dataset was cleaned using a variety of pre-processing techniques, including:

Lower casing means converting input text to the same casing format so that all characters with different cases are handled the same.

Punctuation elimination, HTML tags and links elimination - To standardise the content, remove punctuation, tags and links which are of no significance to the text for the summarization purpose.

Remove stopwords and frequently recurring words - Terms like "my" and "but" that are frequently used in a text but add value while summarizing should be removed.

Stemming means to convert the derived words to their root form like 'staying' to 'stay'.

2.5. Different transformer architectures

2.5.1. Bidirectional and Autoregressive Transformers (BART)

BART comprises two major components, a bidirectional encoder and a decoder. It is quite similar to BERT but pretrained on "facebook/bart-large-cnn" and then uses a tokenizer. This tokenizer is based on the GPT-2 tokenizer. The encoder is fairly similar to BERT and the decoder similar to GPT-2. The decoder used in BART is autoregressive in nature and this when regulated can be used for text summarization(NLP task).

It uses denoising as the pre-training purpose. 6 layers in each, the encoder and the decoder are used in the base model of BART, whereas this number becomes 12 when it comes to the large model. Fine-tuning BART is helpful in applications such as sequence classification, token classification, sequence generation, and machine translation as the representations produced by it are extensively used by these applications.[7](Lewis, 2019)

2.5.2. T5 (Text-to-text transformer)

T5 was trained on a huge amount of text in transfer learning before fine tuning on a downstream task.

Seq-to-seq technique is used. Through crossed-attention layers, the encoded input is transmitted on decoder. Output generated by the decoder is of autoregressive nature. A sequence of tokens is given to the encoder to be mapped to a series of embeddings.[9](Raffel, 2019)

The encoder is made up of two parts: a self-attention layer and a feed forward network. Before proceeding to each self-attention layer, there is a general attention mechanism which differentiates encoder from decoder; or else, their structures are alike.

As a result, previously developed outputs can be utilised. The decoder's output is then transferred to a second dense layer, with softmax as the activation function.The input embedding matrix consists of weights from this layer's outputs.

2.5.3. Pegasus

PEGASUS, developed by Google, expands to Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence models. The most important lines from the input data are extracted and they are then compiled as separate outputs. Also, choosing the most relevant sentences is better than randomly selecting sentences.[11](Zhang, 2019)

This model is one of the most preferable models for abstractive summarization because it is similar to the ways humans generate a summary by reading the entire document and then producing a summary. The model is pre-trained on the newspaper CNN/DailyMail datasets.

3. RESULTS:

Quantitative Analysis: The Rouge scores for all the three models are compared and the models have used the same data while doing so. The results state that Pegasus has performed better than the other models.

Table 1: Comparison of the rouge scores of the three models

MODELS	ROUGE-1	ROUGE-2	ROUGE-L
T5	0.327122	0.087318	0.173913
BART	0.245524	0.066838	0.143223
PEGASUS	0.351351	0.217391	0.243243

Qualitative Analysis: The summaries generated by different models are compared against a manually generated summary. A news article about The US President on Covid-19 has been summarized.

Reference Summary:

"The US has over 637,000 confirmed coronavirus cases and over 30,826 deaths. Trump said some states may be able to return to normalcy earlier than expected. New guidelines to reopen the country will be announced on Thursday after he speaks to governors."

Table 2: The summaries generated by the three models

T5	"The us has over 637,000 confirmed Covid-19 cases and over 30,826 deaths. president Donald Trump predicts some states will reopen the country in april, he said. 'we'll be the comeback kids,all of us,' the president says."
BART	'The US has over 637,000 confirmed Covid19 cases and over 30,826 deaths, the highest for any country in the world. Trump said new guidelines to reopen the country would be announced on Thursday after he speaks to governors. 'We'll be the comeback kids, all of us,' he said.'"
PEGASUS	"'We want to get our country back.'The Trump administration has previously fixed May 1 as a possible date to reopen the world's largest economy, but the president said some states may be able to return to normalcy earlier than that."

4. DISCUSSION

In recent years, transformers have been the go-to models for the purpose of abstractive text summarization. The most popular transformer models are T5, BART and Pegasus.

T5 (Text-to-Text-Transfer-Transformer), a recent Google release, claims to surpass existing high-end algorithms such as BERT[2](Devlin, 2018), GPT2, and others on NLP tasks like question answering, text classification, text summarization etc. Another transformer model by Facebook is BART. The latest released model for abstractive summarization is PEGASUS, which combines masked language model (MLM) and gap sentence generation (GSG) to achieve a state-of-the-art result with a lesser sample size. All these three models were used to generate abstractive text summaries on the same input text.

The metric used for evaluation of the summaries generated is ROUGE(Recall-Oriented Understudy for Gisting Evaluation) score. The various system generated summaries are compared to manual summaries often known as the reference summaries.

ROUGE-1 is the measure of the number of overlaps of unigrams in the system generated summary and the reference summary.

ROUGE-2 is the measure of the number of overlaps of bigrams in the system generated summary when compared to the reference summary.

ROUGE-L is the measure of a sequence of words that is common to both reference and system generated summary and is longest possible. The matches may or may not be successive matches.

On comparative analysis of these 3 models- T5, BART and Pegasus, it was found that Pegasus outperforms the other two models and has higher rouge scores. It also generates better summaries for large input texts.

5. CONCLUSION:

The pre-trained models, which were based on the transformer architecture, were executed for the purpose of summarization. The conclusion drawn from this analysis was that finely tuned transformers gave good results. The ROUGE scores[8](Lin, 2004) were computed for the summaries generated by each of the models and weighed against each other for precision, recall, and f-measure. The findings suggest that Pegasus gave results that outperformed the other two models.

Future Scope of this research could be to implement a crossover of these models to improve text summarization in terms of accuracy and coherence of the outlines.

6. REFERENCES:

- Allahyari M, Pouriyeh SA, Assefi M, Safaei S, Trippe ED, Gutierrez JB, Kochut K (2017) 'Text Summarization Techniques: A Brief Survey', CoRR abs/1707.02268:

- Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018) 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', Available at: https://arxiv.org/abs/1810.04805.

- Graves A (2013) 'Generating Sequences With Recurrent Neural Networks', CoRR abs/1308.0850:

- Greene D, Cunningham P (2006) 'Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering', *In Proceedings of the 23rd International Conference on Machine Learning Association for Computing Machinery*, New York, NY, USA, pp 377–384

- Hochreiter S, Schmidhuber J (1997) 'Long Short-Term Memory' available at . https://doi.org/10.1162/neco.1997.9.8.1735

- Kalchbrenner, N. and Blunsom P. (2013) 'Recurrent Continuous Translation Models', Association for Computational Linguistics, pp.1700–1709. Available at: https://aclanthology.org/D13-1176.pdf

- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L. (2019) 'BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension' Available at: https://arxiv.org/abs/1910.13461 [Accessed 9 Nov. 2021].

- Lin, Chin-Yew (2004) 'ROUGE: A Package for Automatic Evaluation of summaries', *Proceedings of the ACL Workshop: Text Summarization Branches Out 2004.*

- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J. (2019). 'Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer', Available at: https://arxiv.org/abs/1910.10683.

- Zhang, J., Zhao, Y., Saleh, M. and Liu, P.J., (2019) 'PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization'