Journal of Informatics and Web Engineering

Vol. 3 No. 3 (October 2024) eISSN: 2821-370X

Intelligent Abstractive Summarization of Scholarly Publications with Transfer Learning

Farooq Zaman[1], Munaza Afzal[1], Pin Shen Teh[3], Raheem Sarwar[4*], Faisal Kamiran[5], Naif R. Aljohani[6], Raheel Nawaz[7], Muhammad Umair Hassan[8], Fahad Sabah[9]

[1,2,5]Information Technology University, 346-B, Ferozepur Road, Lahore, Pakistan

[3,4]Manchester Metropolitan University, Manchester M15 6BH, United Kingdom

[6]Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

[7]Staffordshire University, Stock-on-Trent, United Kingdom

[8]Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Ålesund, Norway

[9]Beijing University of Technology, Chaoyang, 100021, China

**corresponding author: (r.sarwar@mmu.ac.uk; ORCiD: 0000-0002-0640-807X)*

Abstract - Intelligent abstractive text summarization of scholarly publications refers to machine-generated summaries that capture the essential ideas of an article while maintaining semantic coherence and grammatical accuracy. As information continues to grow at an overwhelming rate, text summarization has emerged as a critical area of research. In the past, summarization of scientific publications predominantly relied on extractive methods. These approaches involve selecting key sentences or phrases directly from the original document to create a summary or generate a suitable title. Although extractive methods preserve the original wording, they often lack the ability to produce a coherent, concise, and fluent summary, especially when dealing with complex or lengthy texts. In contrast, abstractive summarization represents a more sophisticated approach. Rather than extracting content from the source, abstractive models generate summaries using new language, often incorporating words and phrases not found in the original text. This allows for more natural, human-like summaries that better capture the key ideas in a fluid and cohesive manner. This study introduces two advanced models for generating titles from the abstracts of scientific articles. The first model employs a Gated Recurrent Unit (GRU) encoder coupled with a greedy-search decoder, while the second utilizes a Transformer model, known for its capacity to handle long-range dependencies in text. The findings demonstrate that both models outperform the baseline Long Short-Term Memory (LSTM) model in terms of efficiency and fluency. Specifically, the GRU model achieved a ROUGE-1 score of 0.2336, and the Transformer model scored 0.2881, significantly higher than the baseline LSTM model, which reported a ROUGE-1 score of 0.1033. These results underscore the potential of abstractive models to enhance the quality and accuracy of summarization in academic and scholarly contexts, offering more intuitive and meaningful summaries.

Keywords— Text Summarization, Deep Learning, Scholarly Publications, Computational Intelligence, Transformer, LSTM

Received: 05 June 2024; Accepted: 05 August 2024; Published: 16 October 2024

Journal of Informatics and Web Engineering
https://doi.org/10.33093/jiwe.2024.3.3.16

Published by MMU Press. URL: https://journals.mmupress.com/jiwe

1. INTRODUCTION

Automatic text summarization aims to generate a document summary without human involvement. Text summarization is an area with much promise in today's age of information overflow. In the domain of scientific literature, the rate of publications grows exponentially, which requires efficient automatic summarization tools [1].

These models offer significant practical applications for academic settings. For research paper summarization, they can be utilized by researchers and academics to generate concise summaries of scholarly papers, which facilitates quicker literature reviews and a better understanding of key findings. Additionally, institutions and publishers can leverage these models to automate the generation of abstracts for academic publications, thereby enhancing consistency and efficiency in summarizing research contributions. In terms of deployment, integrating our models with academic databases and search engines could provide users with efficient access to summarized research articles, streamlining the process of finding relevant literature. Furthermore, implementing these models through user-friendly interfaces in academic tools or platforms could greatly enhance accessibility, allowing researchers to easily generate or review summaries.

This is a challenging task because a better understanding of the language model is required to produce such a structurally correct summary that contains meaningful phrases in it [2]. Standard natural language processing (NLP) techniques for this task require domain experts to craft the set of features manually. Due to the availability of the massive amount of data, deep learning algorithms can play an important role as they learn the most important features directly from data without needing any domain expert to extract features, and they can automatically shorten longer texts and generate such summaries that can deliver the intended messages by carrying most useful and important information. Researchers have been trying to improve existing techniques for summary generation so that a machine-generated summary matches the human-made summary [3].

Text summarization methods can be organized into extractive and abstractive. In extractive text summarization, the model picks up all words from the source document to make up the new summary. While, in abstractive text summarization, the model tries to generate phrases and sentences by using those words that are not present in the source document but have similar meanings. Previous research on summarizing scientific articles has focused purely on extractive methods [4]. However, the researchers now focus on abstract rather than extractive summarization techniques. This is because the summary generated through extractive techniques usually is longer than usual. This is mainly because its sentences contain those parts from the source document which are not meaningful. Thus, it wastes time and space. Whereas a summary generated through abstractive techniques is more coherent, precise, and clear. In this study, we aim at abstractive title generation from scientific articles.

In most previous studies, summarization has been performed on news articles to generate a new title on limited datasets [5],[6]. In this study, we aim at abstractive title generation from the abstracts of scientific articles. The core objective of this work is to generate a headline that it contains the main crux of the article, providing enough relevant information in the form of a summary, just like a human-made summary. In this study, we employ the Transformer-based [7] model, which is inherently saleable for large datasets for text generation tasks. Furthermore, this study also extends the previously proposed work [8] by using the Recurrent Neural Network family (GRU; Gated Recurrent Unit) model for abstractive text summarization due to its main advantage of solving the vanishing gradient problem. Specifically, we aim to explore such a model, which will help in generating a simplified summary (in the form of headlines) from its corresponding abstracts of scientific articles using a limited scope of the dataset and this summary would be condensed and informative enough, consuming less training time and computation power. The abstracts of scientific articles are more difficult to summarize than news articles because of their compact, imprecise discourse style containing the problem, methodology, experiments/results, and conclusions [9].

Moreover, we aim to measure the performance of generated text by comparing the scores of the proposed model with the already implemented baseline model on the same hyper-parameters using the ROUGE metric [10]. Although researchers have been working towards text summarization to get state-of-the-art ROUGE results, no improvement has been seen in reducing model training time and getting a better version of summarization.

The rest of the paper is organized as follows. Section 2 describes the literature review in detail, covering existing techniques and methods that have been used for text summarization. Section 3 proposes the methodology covering the data set collection and preprocessing details and the overall approaches used for generating the title from its abstract. The implementation detail of the proposed solution has also been discussed here. Section 4 presents

quantitative and qualitative analysis and discusses the results in detail compared to previous approaches. Section 5 presents concluding remarks on the overall research work and points out the future directions that could be followed.

2. RELATED WORK

The extractive approach involves selecting the most relevant phrases and lines from the original papers and combining them to form a summary. In this case, every line and word in the summary is directly taken from the original document. On the other hand, the abstractive approach uses deep learning techniques to generate summaries. This method creates new phrases and terms that differ from those in the original document while preserving the core ideas, similar to how one might summarize in their own words. Consequently, the abstractive approach is considerably more challenging than the extractive method.

Previous research on summarizing scientific articles has focused purely on extractive methods [4]. However, the researchers now focus on abstractive summarization techniques instead of extractive summarization techniques. Unlike extractive methods, a summary generated through abstractive techniques is more coherent, precise and clear. In this following section, we review abstractive text summarization studies.

2.1 Abstractive Text Summarization (Earliest Prominent Works)

Starting from the earliest notable works in this area, the Neural Attention Model was presented for abstractive sentence summarization in headline form [5]. The feed-forward neural network language model (NNLM) [11] was tested in news articles in the New York Times. There were grammar mistakes in the summary generated through the NNLM model. Also, word order was also changed in summary, which was generated through an Attention-Based encoder. In the same year, the hierarchical Long Short-Term Memory (LSTM) autoencoder model was presented with an attention mechanism for multi-sentence generation [12]. An autoencoder is like a neural model in which the input and output units are identical. Later, multi-sentence abstractive summarization was performed using an attentional encoder-decoder recurrent neural network [6]. For this purpose, they used an attentional GRU bi-directional neural network. The Encoder-Decoder RNN with Attention model adopted a Large Vocabulary Trick (LVT) [13] in which the decoder picks the words that exist in the source input text while decoding. In another model, they used the Feature-rich Encoder technique in which additional embedding matrices are created based on look-up embedding to capture linguistic features, i.e., TF and IDF information, named-entity tags, and parts-of-speech tags that help in determining the words that have high importance in the text. They have handled unknown or out-of-vocabulary (OOV) words using the pointer-generator network, where the decoder decides either to point (copy word from the input source text) or generate a word based on its context at each timestamp. For a good summary, it is essential to capture the keywords in the input document and find meaningful, vital sentences. To achieve this goal, the authors of this paper have also used another model where two bi-directional RNNs have been used to capture the critical keywords and key sentences from the input document where the attention mechanism operates simultaneously at the word and sentence levels. Later, in the same year, a data-driven approach for summarization of a single document was presented [14]. The framework consisted of a hierarchical document encoder and extractor based on an attention mechanism to identify the most important sentences to extract and generate an abstractive summary.

Besides this, abstractive summarization was performed with Pointer-Generator Networks [15]. This model can copy words from the source input text via pointing, which helps in reproducing information, and finally, novel words are produced through the generator. It also solved inaccuracies found in the summary; the problem of word repetition was also solved using a coverage mechanism. The only drawback with this model was that there was no strategy for selecting content, and its copying mechanism was copying too much data, even long sentences and phrases. This same year, an approach was presented for scientific and structured document summarization [16]. The approach used in this paper had four steps named segmentation of document, language models generation, term selection and sentence extraction. To better handle the redundancy of words or phrases in abstractive summary, an RNN encoder-decoder model was presented with Word Frequency Estimation (WFE) as an additional component [17]. In the same year, the authors presented a model that performed query-based abstractive summarization with an attention mechanism to handle various issues found in the basic encode-attend-decode model [18]. The model has two additions (a) a query and document attention model, which tries to understand different parts of the query at each time stamp, and (b) to solve the problem of repeating words and phrases in the generated abstractive summary, a model which is diversity-based and with an attention mechanism. Later on, a neural network model was presented for abstractive text

summarization with an innovative intra-attention mechanism that looked at the input in detail to generate an output summary [3]. The model was trained with reinforcement learning and teacher forcing to generate a readable and coherent summary. New York Times and CNN/Daily Mail datasets were used for evaluation using the ROUGE metric. For a generation of abstractive summary, authors implemented a model which was based on sequence-to-sequence encoder-deep recurrent generative decoder (DRGN) [19]. Their motivation was to present a model that could learn the hidden structure information present in the target summaries and could enhance the performance of abstractive summary generation by using this information.

2.2 Abstractive Text Summarization (Recent Prominent Works)

More recently, for abstractive text summarization, authors presented a model based on an encoder-decoder architecture with deep communicating agents [20]. In this model, the encoding task is divided into several collaborating agents who help in sub-divisions of the input source text. Encoders in this model are associated with a single decoder to generate a coherent and focused summary trained using reinforcement learning. This way of communication of multiple encoders causes to generate a summary of higher quality compared to the models used in baselines. Afterwards, researchers [21] attempted to address the Neural Abstractive Summarization of single, longer-form documents like Research papers in 2018. Earlier, these models (Neural abstractive summarization) have been used in summarizing relatively short documents. The authors presented a neural sequence-to-sequence model that was able to summarize long and structured documents effectively. The approach used in the paper consisted of a new hierarchical encoder that exhibits the structure of a document and a decoder (discourse-aware) for a summary generation. Nuances in the coherence or coverage of the summaries were not appropriately captured through the evaluation of the ROUGE metric.

Based on actor-critic approaches, an attention-based seq2seq framework was employed, for the actor, as the policy network for neural abstractive summarization [22], whereas for the critic, the maximum likelihood estimator was combined with the global summary quality estimator. An alternative training strategy was suggested for joint learning of both (the actor and critic). This solution was proposed to avoid low-quality summaries or summaries with incorrect phrases. To capture the main gist of the text, summarization and sentiment classification [42], [43] are helpful techniques but at different stages. For this purpose, a hierarchical end-to-end model was proposed [23] for learning both text summarization and sentiment classification, where sentiment classification also helps in text summarization. Its layer is placed above the layer of text summarization, from which a hierarchical structure was obtained. The model consists of three components (a) the text encoder, (b) the summary decoder and (c) the sentiment classifier. The model was named Hierarchical Summarization and Sentiment Classification (HSSC).

A deep learning topic-aware model was proposed to handle the automatic text summarization with the help of involving topic information in the convolutional sequence-to-sequence (ConvS2S) model [24]. For optimization, self-critical sequence training (SCST) was used. Summaries generated through this model were full of information, consistent, and varied. Later, in the same year [25], the abstractive summary was generated for Wikipedia articles using a model that has its roots in transformer network [7], but its primary focus was on multi-document summarization. In this paper, the authors have used a transformer decoder with an attention mechanism (memory-compressed) (TDMCA).

A novel framework was introduced named Abstract Meaning Representation (AMR), where source input is parsed into AMR graphs, then these graphs are converted into a summary graph, and finally, the text is generated from the summary graph [26]. In this method, a structured prediction algorithm is used, which converts input semantic graphs into output (single summary) semantic graphs. In the same year, the researchers proposed a simple model that could accurately select content from input documents in the form of phrases to be included in the output for abstractive text summarization [27]. The content selector found in this model could be used for bottom-up attention that limited the words to be copied from the input document to output text as a summary. Previously, summarization has been performed for single documents, but later on, researchers tried to explore neural abstractive methods for multi-document summarization. Authors proposed an approach based on the encoder-decoder network in which all documents of a document set are treated as input, then the encoder converts these documents into a representation of the document set, and finally, the decoder generates a summary [28].

Recently, LSTM-CNN, an abstractive text summarization framework, proposed ATSDL that could generate entirely new sentences using semantic phrases [29]. This framework was different from already existing methods, and it was

a two-step process; in the first step, key phrases are extracted from the input sentences and in the second step, summaries are generated using the deep learning model (LSTM). Later in the same year, a Hybrid learning model was proposed for Abstractive Text Summarization (HATS), which could read a document as humans do. Earlier, no such methodology was adopted for summarization, so this strategy took the lead in its work. Their model was divided into three phases; an attentional network, an encoder-decoder network and a generative adversarial network module, which were synchronized together to read like humans. The summary generated through this process was fluent, containing important information [30]. A mixture model was proposed to enhance the machine-generated abstractive summary in its generalized form with its roots in deep learning techniques and semantic data transformations [31]. This model helped generate general summaries, which the model further transformed to be readable by humans. This model also solved out-of-vocabulary and rare word problems. Very recently, BERT based extractive text summarization has been performed where the authors of the paper have called this discourse-aware neural extractive summarization model DISCOBERT, which has its basis in BERT [32]. The summary generated through this model is short in length and conveys important and less redundant information. DISCOBERT with discourse unit has helped minimize text redundancy in the generated summary. In the same year, the authors performed both extractive and abstractive summarization using a new dataset (proceedings of 41st International ACM SIGIR 2018) which has human-written goal summaries for the summary evaluation. This dataset consisted of abstracts which the humans and academic publications have written. LSTM-NN was used to deal with the semantic and syntactic features required for summarization [33].

To capture the main concepts of the input text for summarization, an encoder-decoder model with a double attention pointer network (DAPT) has been proposed [34]. This model (DAPT) works in three steps: 1) first, core concepts of the text are taken by the self-attention mechanism from the encoder part, 2) after which main content is generated by the soft-attention and pointer network, which is more coherent and consistent, 3) in the last step, both strategies combine to generate a fluent summary. Later on, a novel network was proposed for abstractive text summarization named the Hierarchical Human-like deep neural network for Abstractive Text Summarization (HH-ATS), which could grasp the necessary information by reading from the input document like humans do and could generate relevant summary [35].

The significant difference in our methodology compared to other studies is to improve the ROUGE scores using less training time and data yet to generate impressive summaries (titles) from our encoder with greedy-search decoder GRU neural network model and Transformer models. We found that training our GRU model was 1.5 times faster than the baseline model (LSTM), and the Transformer model was 10 times faster than the baseline model (LSTM).

3. DATA AND METHODOLOGY

We used two models in this study to generate abstractive titles from scientific documents. The first model is based on GRU, which belongs to the recurrent family of algorithms. The second model employs Transformer architecture which eliminates bottleneck of recurrency by leveraging attention mechanism during training [7].

3.1 Dataset and Preprocessing

We evaluate the models on a data set comprised of five million abstracts of scientific articles with their corresponding summaries in the form of titles. Dataset can be downloaded from google repository, which contains 5 million papers in the biomedical domain, having both the versions of the dataset as raw dataset and pre-processed dataset. The dataset is present in compressed format, and before accessing the dataset, its files have been uncompressed. The dataset is transformed into a training, validation, and testing split. This dataset is extracted from MEDLINE (Medical Literature Analysis and Retrieval System Online, or MEDLARS Online), which is a bibliographic database of life sciences and biomedical information. It contains metadata of scientific articles of almost 25 million papers in the biomedical domain in XML format. Compiled by the National Library of Medicine (NLM), MEDLINE is available on the Internet without any cost, and it can be searched via PubMed. To compare the results, our model is trained on the same training data of scientific journal articles [9] reported by previous work on abstractive text summarization using recurrent neural networks (RNNs), which is present in the form of title-abstract pairs (title-gen). As extensive computational power or machine is required to train a model on whole 5 million training records, so we limited our model training to 100K training records and 1K testing records for model evaluation.

The data files are processed to pair the abstract of a paper to its title to form a title-gen dataset for summarization (title generation from its abstract), where figures, tables or headings in the body were skipped. Several pre-processing techniques were also applied, like tokenization and conversion of uppercase letters to lowercase. Besides, URLs found in the data were removed, and all numbers were replaced with a hash (#). Furthermore, all those pairs of titles and abstracts were excluded from data where the abstract length or title length was not in the range of 150-370 tokens and 6-25 tokens, respectively. We have applied the following further pre-processing steps to data: (i) We included a start and end token to each sentence so that the model could understand when to start and stop predicting, (ii) To clean the sentences, special characters were also removed, (iii) We created a word index and reverse word index, which is a mapping of dictionaries from word → id and id → word), and (iv) Each sentence was also padded to reach a maximum length.

3.2 Methodology

Since abstractive summarization is a sequence-to-sequence (seq2seq) problem where a source length and target length vary, to solve such problems, an encoder-decoder neural network is used. Encoder-Decoder architecture works well when the input source length is standard. This is because a basic encoder-decoder's performance declines as an input sentence's length increases. To overcome the memory and performance limitations of encoder-decoder architecture, an attention mechanism is introduced to predict a word by looking at a few specific parts of the input sequence instead of focusing on the entire sequence. In this research, we compare the results of Attentional Encoder-Decoder Recurrent Neural Networks, our proposed model, i.e., encoder with greedy-search decoder GRU, and Transformer-based model with that of baseline work, i.e., encoder-decoder LSTM [8]. This study aims to summarize a huge content of the source (abstract), with the help of a deep learning model, by compressing its main content into its shorter version, like title/headline generation, using vocabulary that is not present in the source document.

3.2.1 GRU Model with Attention Layer

The overall approach for the GRU has been explained below with the help of Figure 1. The encoder reads source text as input text and generates hidden states at each time step. The encoder's combined state is treated as a first input to the decoder. The decoder's output is called the decoder's 1st hidden state. A score (scalar), which is a dot product between the encoder's hidden states and the decoder, is obtained by an alignment model/score function. The final scores are then fed to a softmax layer. The alignment/annotation vector is obtained by multiplying the softmax score with each of its encoder hidden states. The alignment vectors are added together to generate the context vector. For the next decoder step, the generated word from the former decoder time step (pink) and context vector from the current time step are passed as input after concatenation.

3.2.2 Fine-tuning Pre-trained Transformer Model

The overall approach for the transformer has been explained below with the help of Figure 2. Since Transformer models do not use recurrency, therefore, the sequence between the input text is captured with the help of positional embeddings. The positional embeddings are concatenated with word embeddings and further feeds into the encoder. The encoder consists of multi-head attention with a feed-forward layer. The encoder states are further passed to the decoder, consisting of masked multi-head attention followed by multi-head attention and feedforward. Unlike the recurrent model, the output generates more than one token sententiously.

In this paper, we have fine-tuned three pre-trained language models, including ProphetNet, Pegasus, and BART, that have achieved state-of-the-art performance for the summarization task. The hyperparameters are given in Section 3.2.3.

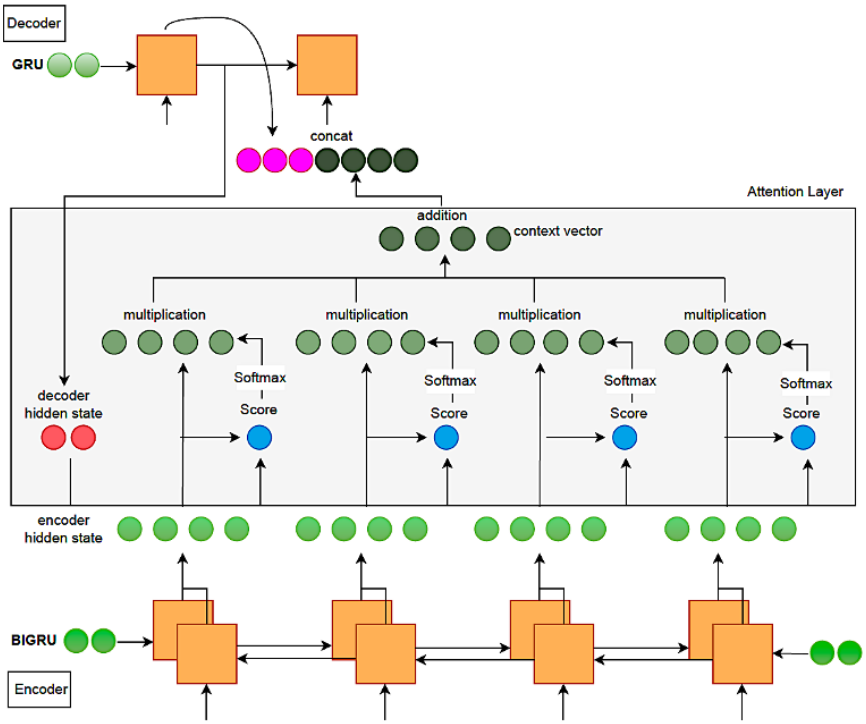

Figure 1. GRU Architecture With Attention Layer Used In The Study

ProphetNet is a sequence-to-sequence pre-trained model that introduces an innovative n-stream self-attention mechanism combined with future n-gram prediction. Unlike traditional sequence-to-sequence models, which are optimized for one-step-ahead prediction, ProphetNet is optimized for n-step-ahead prediction. This approach involves predicting the next n tokens simultaneously, using prior context tokens at each time step. The model is specifically designed to consider future tokens through the future n-gram prediction, which helps to prevent overfitting on local correlations. ProphetNet achieved a new state-of-the-art performance in abstractive summarization after being trained on two datasets: a base-scale dataset (16GB) and a high-scale dataset (160GB).

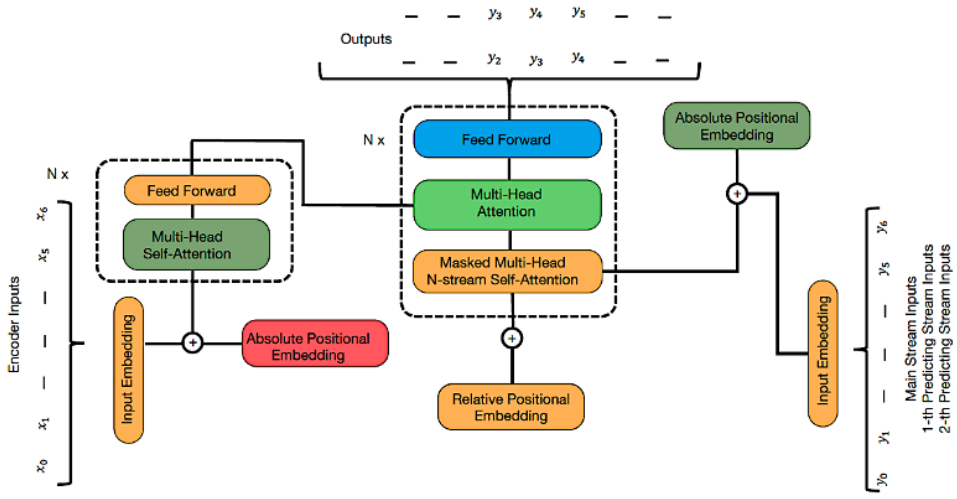

Figure 2. Fine Tuning Of Pretrained Models

The pretraining process for Pegasus is intentionally designed to mimic summarization. This is done by masking or removing key sentences from an input document and then generating a single output sequence from the remaining sentences, similar to how an extractive summary works.

A denoising autoencoder named BART is utilized to pretrain sequence-to-sequence models. The training process for BART begins by corrupting the text using a random noise function, after which the model is trained to reconstruct the original text. BART features a Transformer-based neural machine translation architecture, which, despite its straightforward design, can be seen as a generalization of earlier pretraining methods such as GPT and BERT, due to its bidirectional encoder and left-to-right decoder. It introduces an innovative in-filling technique that involves randomly reordering sentences and replacing long sections of text with a single mask token. BART performs optimally when fine-tuned for text generation tasks.

3.2.3 Experimental Setup

Since abstractive text summarization is a structured prediction problem, we used the neural attentive encoder-decoders framework [36], best suited for solving such problems. The quality of the resultant summary has been measured with the ROUGE metric [10] on the test set.

For abstractive text summarization (title generation from abstract of scientific article), we have implemented our models using GPU enabled Keras and TensorFlow libraries on Google Colaboratory, which is a cloud-based free online Jupyter notebook environment that allows training deep learning models on either CPUs, GPUs, and TPUs. We have implemented all the models, including the LSTM encoder-decoder model [36] with an attention mechanism [37] used in the baseline paper [8] and compared it with our proposed models, our GRU architecture is similar to the baseline paper except that we are using an encoder with greedy-search decoder GRU units instead of LSTM. A recurrent neural network (RNN) [38] is a powerful and expressive framework to handle sequential data where input and output lengths may vary. In an encoder-decoder RNN, the encoder processes the input sequence and transforms it into an internal representation known as the context vector. Conversely, the decoder produces the output sequence by interpreting the encoded input from the encoder. Typically, the decoder is trained to anticipate the next word in the output, using both the context vectors and all previously predicted words as reference.

The model used in our research was trained with the same hyper-parameter settings as defined in the foundational paper [8], except that fewer input/output vocabularies are used. For comparison, we have used the same architecture to train both models on the same provided dataset. The first model is RNN Encoder with beam-search Decoder LSTM [39] used in the baseline paper, while the other is our proposed model, to which we refer as RNN Encoder with greedy search Decoder GRU, both working with attention mechanism [37].

In this study, our training process and choices of hyper-parameter are similar to those used in the baseline paper, except that we have trained both the models on 100K training records only instead of training them on whole training data (5 million) as it is a computationally extensive task which requires a GPU enabled machine along with many additional days for training to complete successfully. Our train/evaluation split is 100K/1K. Specifically, we have trained both multi-layer deep recurrent models for our research, each of which has two layers containing 1000 hidden units, with 500-dimensional word embeddings, and input/output vocabularies were limited from 80K to 25K due to memory limitation. Our hyper-parameters can be summarized as follows: (a) both models were trained for 11 epochs, (b) the optimizer was chosen as Adam [40], (c) the batch size is 64, and (d) the decoder beam search size is 20 and (e) use of sparse categorical cross-entropy as the loss function as it overcomes memory issues if any. Training our model on 100K training records takes about 6-8 days on GPU-enabled Google Colaboratory Jupiter notebook. The training steps are detailed as follows: (i) During the training phase, when the input is processed by the encoder, it generates both the encoder output and the encoder's hidden state, (ii) This information is then transferred to the decoder, which receives the encoder output, the encoder's hidden state, and the decoder input, (iii) The decoder produces the predictions along with its own hidden state, (iv) The hidden state from the decoder is fed back into the model, and the predictions contribute to the calculation of the loss, (v) Through the use of teacher forcing, the subsequent input to the decoder is determined [41], and (vi) Teacher forcing involves providing the output (target) word to the decoder as the next input, which aids in reducing the loss function at each step of decoding in GRU and LSTM-based models as in Equation (1).

$$\text{Loss}_{GRU\_LSTM} = -\sum_{k=1}^{n} \log p(\, y_k \mid y_1, \ldots, y_{k-1}, x_1, \ldots, x_n \,) \tag{1}$$

Here, $y = \{y_1, y_2, \dots, y_n\}$ is defined as the output sequence of the ground truth while $x = \{x_1, x_2, \dots, x_n\}$ is a given input sequence. In the last step, gradients are calculated and applied to the optimizer and back-propagate.

For the Transformer model, we used the following loss function: the combination of language model loss and future n-gram loss is shown on Equation (2).

$$\text{Loss}_{Tranformer} = -\alpha_0 \left(\sum_{t=1}^{T} \log P\,\theta(y_t \mid y_{<t}, x) \right) - \sum_{j=1}^{m-1} \alpha_t \cdot \sum_{t=1}^{T-j} \log P\,\theta\left(y_{t+j} \mid y_{<t}, x\right) \quad (2)$$

4. RESULTS AND DISCUSSION

This section evaluates and compares our models through quantitative and qualitative analysis. The output of our model is abstractive summary in the form of one-liner headline generation carrying the most relevant textual information.

4.1 Automatic Evaluation

We evaluated the performance of both recurrent models studied under this research on the provided dataset using the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric [10], which is a recall-based metric, frequently used for summarization and has also been used by DUC (Document Understanding Conferences) for their tasks. This metric is used to determine the quality of machine-generated summaries by comparing them with ground truth summaries, which have been created by humans, by counting the overlapping units such as n-gram, word sequences, and word pairs between them (Equation (3)).

$$\text{ROUGE} = \frac{\text{Number of overlapping words}}{\text{Total number of words in reference summary}} \quad (3)$$

It has several automatic evaluation measures that could be used to measure the similarity between summaries. Our reporting uses the ROUGE-1, ROUGE-2, and ROUGE-L measures. Here, ROUGE-1 and ROUGE-2 are unigram and bigram overlap to measure similarity between ground truth and machine generated summaries and assess informativeness, while ROUGE-L measures the longest common subsequence, a means of assessing fluency. For each ROUGE measure, a recall (R) score and a precision (P) score are balanced using the F1 score. The fundamental way to compute the F1 score is to count a harmonic mean of precision and recall, as shown in Equation (4).

$$F1 = 2 \times \frac{P \times R}{P + R} \quad (4)$$

ROUGE computes the overlap of words between predicted output and actual output. We run ROUGE on our generated and gold summaries for 1K test records. Below is the Inner detail of the evaluation procedure.

The evaluate function operates similarly to the training loop, but without employing teacher forcing. The prediction process continues until the model predicts the end token, and attention weights are preserved at each time step. At each step, the decoder receives its previous predictions, the hidden state, and the encoder output as input.

Furthermore, it has been observed that our model (attentional GRU encoder without beam-search decoder) is quite effective as compared to the baseline model in predicting headlines from the abstracts of scientific articles it was trained on when the greedy-search decoder is used instead of the beam-search decoder. Generally, our proposed model seems to capture the gist of the text by introducing completely new words while generating the title/headline of an abstract. However, on the other hand, the baseline model (encoder with beam search decoder LSTM) shows poor performance as there are multiple repetitive words in the generated summary/headline.

4.2 Discussion

Results of the baseline model and produced by our models (attentional encoder with greedy-search decoder GRU) and Transformer model are given in Table 1. Our model 1 achieved Rouge-1, Rouge-2, and Rouge-L at 23.4%, 3.5% and 23.3%. Whereas model 2 achieved Rouge-1, Rouge-2, and Rouge-L at 28.8%, 11.08% and 25%. Model 3 achieved Rouge-1, Rouge-2, and RougeL at 31.2%, 17.4% and 32.9% while model 4 achieved Rouge-1, Rouge-2, and Rouge-L at 44.4%, 22.8% and 38.3%.

Table 1. Rouge-F1 Metric Results For The Title-gen Dataset: R-1, R-2, R-L represent the ROUGE-1/2/L metrics

Model	ROUGE-1	ROUGE-2	ROUGE-L
Baseline: LSTM	0.1030	0.00285	0.1466
Model 1: GRU with Attention	0.23361	0.03464	0.23321
Model 2: Transformer-ProphetNET	0.28809	0.11086	0.25002
Model 3: Transformer-PEGASUSLARGE	0.31241	0.17450	0.32904
Model 4: Transformer-BART	0.44392	0.22769	0.38256

We have shown the summarization performance measured in the ROUGE score, obtained on 100K training records and computed on 1K test records. It is clear from the table that our proposed model outperforms the baseline model. Furthermore, we also observed that evaluation through-beam search decoding on test records slows down the output generation. Finally, the Transformer-based models such as ProphetNET, PEGASUSLARGE, and BART outperform our solution; however, they are computationally expensive.

We have shown system-generated summaries (titles) of both the models in Figure 3. along with the ground truth summary (original title), and it can be seen clearly that our model can produce a legible summary having the same context as of the ground truth title, even though both titles do not match by words. While on the other hand, the same words or phrases often get repeated in the summary generated through the baseline model.

Source Document (Abstract of Scientific Article)
<start>measurements of cardiac performance for humans at various ages is influenced by the variable examined, the population and techniques employed, and the factors that co-vary with age, including the presence of disease and physical conditioning. Interstudy differences in the extent to which occult coronary disease is present in older subjects and in the level of physical conditioning among subjects may underlie the variable perspectives contained in the literature of how aging affects cardiovascular function. In carefully screened, highly motivated, but not athletically trained community-dwelling subjects, resting cardiovascular parameters are not age-related except for systolic blood pressure, which increases with age. During vigorous exercise, the mechanisms used to achieve a high level of cardiac output shift from a dependence on a catecholamine-mediated increase in heart rate and inotropy to a dependence on the frank starling mechanism. One reason for the age difference in cardiovascular response to exercise may be a diminished responsiveness to beta-adrenergic stimulation in these subjects. In other elderly subjects who cannot exercise due to high work loads, a decline in stroke volume, as well as heart rate at peak exercise, has been observed. Whether the inability of these individuals to augment stroke volume is caused by a decrease in the ability of the heart to increase diastolic filling, by a decrease in systolic pump function caused by an increased afterload, by intrinsic myocardial contractile defects, or by a greater diminution of the cardiovascular response to beta-adrenergic stimuli is presently unknown. <end>
Ground truth Summary (Actual Title): <start>cardiovascular response to exercise in younger and older men. <end>
Baseline LSTM Model: influence, and human. influence of rats in young and is the influence of rats and function and function and function and function and function and function and function and function and function and function and DNA.
GRU with Attention layer: heart rate variability during exercise. <end>
Transformer model: age-related changes in cardiovascular function in healthy and elderly subjects.

Figure 3. An Example Of A Headline Generated By The Models Studied In This Research And Their Comparison With The Ground Truth Summary

5. CONCLUSION

With ever-increasing textual content every day, there is a need for a method that can automatically summarize the entire content in the precise form, usually in a headline, conveying the most valuable and essential information. In this research work, we have applied the attentional encoder-decoder recurrent neural network to generate an abstractive summary in the form of single-line headline generation instead of lengthy summaries due to the complexity and various limitations such as colossal memory requirements, long training time or even architectures inefficiency with long output sequences. For summarization, we have trained two recurrent models (an encoder with beam-search decoder LSTM and an encoder with greedy-search decoder GRU units and attention), along with the Transformer model, to generate headlines using scientific articles' texts from the MEDLINE dataset. After comparing models on the same hyper-parameters and identical training records of the dataset, results reveal that our proposed models outperform the baseline model on the evaluation metric with promising results, as shown in the results and discussion section. For evaluation, ROUGE F1 metric is used. Furthermore, it has also been observed that the summary generated from the baseline model has poor quality as there are word repetitions in the summary, as exhibited by the baseline model.

After showcasing our models' results, we believe that there is still room for many improvements in future work that could serve as a continuity of this work. Although this research provides a better approach to saving computational cost while training on a subset of the training dataset, we still believe it is challenging to train such models on this complete (5 million) dataset because of the large memory and computation requirements. Additionally, as part of future work, a plan is needed to focus on building more robust models by generating summaries comprising multiple sentences. Along with it, one more challenge is left for the future: to better handle rare or unknown words in the source content. While our work has been focusing on text summarization of documents in English, we expect this work to be carried over to other languages (supporting multi-lingual summarization). Moreover, we plan to implement an attention mechanism from scratch in future work.

ACKNOWLEDGEMENT

The authors received no funding from any party for the research and publication of this article.

AUTHOR CONTRIBUTIONS

Farooq Zaman: Experimental Setup, Writing - Methodology
Munaza Afzal: Conceptualization, Data Curation, Writing - Draft preparation
Pin Shen Teh: Experimental Setup, Writing - Review & Editing
Raheem Sarwar: Supervision, Writing - Results and Discussion
Faisal Kamiran : Supervision, Validation
Naif R. Aljohani: Supervision, Writing - Literature Review
Raheel Nawaz: Supervision, Writing - Introduction
Muhammad Umair Hassan: Writing - Preliminaries, Conclusion & Future Recommendation
Fahad Sabah: Writing - Literature Review

CONFLICT OF INTERESTS

The authors declare that there are no conflicts of interest regarding the publication of this paper.

ETHICS STATEMENTS

This work did not involve human subjects, animal experiments, or data collected from social media platforms. Therefore, the corresponding ethical statements regarding informed consent, animal ethics permissions, and social media data compliance are not applicable.

All authors confirm that there are no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] B. Dorr, D. Zajic, and R. Schwartz, "Hedge Trimmer: A parse-and-trim approach to headline generation," *HLT-NAACL 03 Text Summarization Workshop*, 2003, doi: 10.3115/1119467.1119468.

[2] H. Jing, "Using Hidden Markov Modeling to Decompose Human-Written Summaries," *Computational Linguistics*, vol. 28, no. 4, pp. 527–543, 2002, doi: 10.1162/089120102762671972.

[3] R. Paulus, C. Xiong, and R. Socher, "A Deep Reinforced Model for Abstractive Summarization," *arXiv (Cornell University)*, 2017, doi: 10.48550/arxiv.1705.04304.

[4] A. Nenkova, "Automatic Summarization," *Foundations and Trends® in Information Retrieval*, vol. 5, no. 2, pp. 103–233, 2011, doi: 10.1561/1500000015.

[5] A. M. Rush, S. Chopra, and J. Weston, "A Neural Attention Model for Abstractive Sentence Summarization," *arXiv (Cornell University)*, 2015, doi: 10.48550/arxiv.1509.00685.

[6] R. Nallapati, B. Zhou, C. D. Santos, C. Gulcehre, and B. Xiang, "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond," SIGNLL Conference on Computational Natural Language Learning, 2016, doi: 10.18653/v1/k16-1028.

[7] A. Vaswani et al., "Attention Is All You Need," *arXiv (Cornell University)*, 2017, doi: 10.48550/arxiv.1706.03762.

[8] N. I. Nikolov, M. Pfeiffer, and R. H. R. Hahnloser, "Data-driven Summarization of Scientific Articles," *arXiv (Cornell University)*, 2018, doi: 10.48550/arxiv.1804.08875.

[9] R. Sarwar, M. Perera, P. S. Teh, R. Nawaz, and M. U. Hassan, "Crossing Linguistic Barriers: Authorship Attribution in Sinhala Texts," *ACM Transactions on Asian and Low-Resource Language Information Processing*, 2024, doi: 10.1145/3655620.

[10] C.-Y. Lin, Information Sciences Institute, and University of Southern California, "ROUGE: A Package for Automatic Evaluation of Summaries," 2004. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/07/was2004.pdf

[11] Y. Bengio, R. Ducharme, and P. Vincent, "A Neural Probabilistic Language Model," *Journal of Machine Learning Research*, vol. 3, pp. 932-938, 2000, doi: 10.1162/153244303322533223.

[12] J. Li, T. Luong, and D. Jurafsky, "A Hierarchical Neural Autoencoder for Paragraphs and Documents," *Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, doi: 10.3115/v1/p15-1107.

[13] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, "On Using Very Large Target Vocabulary for Neural Machine Translation," Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, doi: 10.3115/v1/p15-1001.

[14] J. Cheng and M. Lapata, "Neural Summarization by Extracting Sentences and Words," *54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, doi: 10.18653/v1/p16-1046.

[15] A. See, P. J. Liu, and C. D. Manning, "Get To The Point: Summarization with Pointer-Generator Networks," *55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jan. 2017, doi: 10.18653/v1/p17-1099.

[16] J. M. Conroy and S. T. Davis, "Section mixture models for scientific document summarization," *International Journal on Digital Libraries,* vol. 19, no. 2–3, pp. 305–322, 2017, doi: 10.1007/s00799-017-0218-6.

[17] J. Suzuki and M. Nagata, Cutting-off Redundant Repeating Generations for Neural Abstractive Summarization, *arXiv (Cornell University),* 2017. doi: 10.18653/v1/e17-2047.

[18] P. Nema, M. M. Khapra, A. Laha, and B. Ravindran, "Diversity driven attention model for query-based abstractive summarization," *55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* 2017, pp. 1063–1072.

[19] P. Li, W. Lam, L. Bing, and Z. Wang, "Deep Recurrent Generative Decoder for Abstractive Text Summarization," *arXiv (Cornell University),* 2017. doi: 10.48550/arxiv.1708.00625.

[20] A. Celikyilmaz, A. Bosselut, X. He, and Y. Choi, "Deep Communicating Agents for Abstractive Summarization," *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers),* 2018, doi: 10.18653/v1/n18-1150.

[21] A. Cohan et al., "A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents," *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers),* 2018, doi: 10.18653/v1/n18-2097.

[22] S. K. Abbas et al., "Vision based intelligent traffic light management system using Faster R-CNN," *CAAI Transactions on Intelligence Technology,* 2024, doi: 10.1049/cit2.12309.

[23] S. Ma, X. Sun, J. Lin, and X. Ren, "A Hierarchical End-to-End Model for Jointly Improving Text Summarization and Sentiment Classification," *arXiv (Cornell University),* 2018, doi: 10.48550/arxiv.1805.01089.

[24] L. Wang, J. Yao, Y. Tao, L. Zhong, W. Liu, and Q. Du, "A Reinforced Topic-Aware Convolutional Sequence-to-Sequence Model for Abstractive Text Summarization," *Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18),* Jul. 2018, doi: 10.24963/ijcai.2018/619.

[25] P. J. Liu et al., "Generating Wikipedia by Summarizing Long Sequences," *arXiv (Cornell University),* 2018, doi: 10.48550/arxiv.1801.10198.

[26] F. Sabah, Y. Chen, Z. Yang, M. Azam, N. Ahmad, and R. Sarwar, "Model optimization techniques in personalized federated learning: A survey," *Expert Systems With Applications*, vol. 243, p. 122874, 2024, doi: 10.1016/j.eswa.2023.122874.

[27] S. Gehrmann, Y. Deng, and A. Rush, "Bottom-Up Abstractive Summarization," *Conference on Empirical Methods in Natural Language Processing,* 2018, doi: 10.18653/v1/d18-1443.

[28] A. Bakar, R. Sarwar, S.-U. Hassan, and R. Nawaz, "Extracting Algorithmic Complexity in Scientific Literature for Advance Searching," *Journal of Computational and Applied Linguistics*, vol. 1, pp. 39–65, 2023. doi: 10.33919/JCAL.23.1.2.

[29] S. Song, H. Huang, and T. Ruan, "Abstractive text summarization using LSTM-CNN based deep learning," *Multimedia Tools and Applications,* vol. 78, no. 1, pp. 857–875, 2018, doi: 10.1007/s11042-018-5749-3.

[30] M. Yang, Q. Qu, W. Tu, Y. Shen, Z. Zhao, and X. Chen, "Exploring Human-Like Reading Strategy for Abstractive Text Summarization," *AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 7362–7369, Jul. 2019, doi: 10.1609/aaai.v33i01.33017362.

[31] M. U. Hassan, S. Alaliyat, R. Sarwar, R. Nawaz, and I. A. Hameed, "Leveraging deep learning and big data to enhance computing curriculum for industry-relevant skills: A Norwegian case study," *Heliyon*, vol. 9, no. 4, p. e15407, Apr. 2023, doi: 10.1016/j.heliyon.2023.e15407.

[32] J. Xu, Z. Gan, Y. Cheng, and J. Liu, "Discourse-Aware Neural Extractive Text Summarization," *58th Annual Meeting of the Association for Computational Linguistics*, Jan. 2020, doi: 10.18653/v1/2020.acl-main.451.

[33] B. Mutlu, E. A. Sezer, and M. A. Akcayol, "Candidate sentence selection for extractive text summarization," *Information Processing & Management*, vol. 57, no. 6, p. 102359, 2020, doi: 10.1016/j.ipm.2020.102359.

[34] Z. Li, Z. Peng, S. Tang, C. Zhang, and H. Ma, "Text Summarization Method Based on Double Attention Pointer Network," *IEEE Access*, vol. 8, pp. 11279–11288, 2020, doi: 10.1109/access.2020.2965575.

[35] M. Yang, C. Li, Y. Shen, Q. Wu, Z. Zhao, and X. Chen, "Hierarchical Human-Like Deep Neural Networks for Abstractive Text Summarization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 6, pp. 2744–2757, 2020, doi: 10.1109/tnnls.2020.3008037.

[36] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," *Advances in Neural Information Processing Systems*, vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

[37] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *arXiv (Cornell University)*, 2014, doi: 10.48550/arxiv.1409.0473.

[38] J. L. Elman, "Finding Structure in Time," Cognitive Science, vol. 14, no. 2, pp. 179–211, 1990, doi: 10.1207/s15516709cog1402_1.

[39] A. Graves, A. -r. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural networks," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645-6649, doi: 10.1109/ICASSP.2013.6638947.

[40] S.-U. Hassan et al., "Exploiting tweet sentiments in altmetrics large-scale data," *Journal of Information Science*, vol. 49, no. 5, pp. 1229–1245, 2022, doi: 10.1177/01655515211043713.

[41] R. J. Williams and D. Zipser, "Experimental Analysis of the Real-time Recurrent Learning Algorithm," Connection Science, vol. 1, no. 1, pp. 87–111, 1989, doi: 10.1080/09540098908915631.

[42] T. A. Khan, R. Sadiq, Z. Shahid, M. M. Alam, and M. M. Su'ud, "Sentiment Analysis using Support Vector Machine and Random Forest," *Journal of Informatics and Web Engineering*, vol. 3, no. 1, pp. 67–75, 2024, doi: 10.33093/jiwe.2024.3.1.5.

[43] H. Ng, M. S. Jalani, T. T. V. Yap, and V. T. Goh, "Performance of Sentiment Classification on Tweets of Clothing Brands," Journal of Informatics and Web Engineering, vol. 1, no. 1, pp. 16–22, 2022, doi: 10.33093/jiwe.2022.1.1.2.

BIOGRAPHIES OF AUTHORS

Farooq Zaman is a PhD Scholar in AI Lab, Information Technology University, Lahore. He received M.Phil (2017) degree in Computer Science from the Department of Computer Science, Quaidi-Azam, University, Islamabad, Pakistan. Prior to joining Scientometrics Lab, he was serving as a visiting faculty at QuaidI-Azam, University, Islamabad, Pakistan. His research interests are in the area of text summarization, text simplification and machine translation. email: phdcs18002@itu.edu.pk.

	Munaza Afzal has completed her Master in Data Science from Information Technology University Lahore. Her research focuses on text summarisation systems machine learning. She can be contacted at email: msds17051@itu.edu.pk.
	**Pin Shen Teh** is a Senior Lecturer at Manchester Metropolitan University. His research focuses on practical applications of machine learning, biometrics systems, cybersecurity and metaverse in education. He can be contacted at email: p.teh@mmu.ac.uk
	**Raheem Sarwar** is a Senior Lecturer at Manchester Metropolitan University. His research focuses on practical applications of machine learning, authorship attribution and higher education. He can be contacted at email: R.Sarwar@mmu.ac.uk
	**Faisal Kamiran** is the Chairperson of the Computer Science Department, the Dean of Engineering, and the Director of the Data Science Lab (DSL), Information Technology University. He is also the Co-Founder and the President of ADDO AI. He received the World Bank Research Innovation Award on his data science work, in April 2017. His research interests include fairness-aware data analytics and machine learning to safeguard the rights of the deprived communities and individuals, ICTD, social media analytics, text mining, and so on.. He can be contacted at email: faisal.kamiran@itu.edu.pk
	**Naif Aljohani** is an Associate Professor at the Faculty of Computing and Information Technology (FCIT) in King Abdul Aziz University, Jeddah, Saudi Arabia. His research focuses on big data analytics. He can be contacted at email: nraljohani@kau.edu.sa
	**Raheel Nawaz** is Pro V C - Digital Transformation in Executive Office at Staffordshire University, United Kingdom. His research focuses on digital transformation, applied artificial intelligence and high education.
	**Muhammad Umair Hassan** is Assistant Professor at NTNU, Norway. His research interest include artificial intelligence and computer vision. He can be contacted at emial: muhammad.u.hassan@ntnu.no
	**Fahad Sabah** is Senior Lecturer in Superior University, Lahore Pakistan. He research focuses on artificial intelligence and federated learning. He is currently pursuing his PhD at BJUT, China. He can be contacted at email: fahad.sabah@emails.bjut.edu.cn