Journal of Autonomous Intelligence (2024) Volume 7 Issue 5
doi: 10.32629/jai.v7i5.1331

REVIEW ARTICLE

Automatic text summarization of scientific articles using transformers—A brief review

Seema Aswani[1], Kabita Choudhary[1], Sujala Shetty[2,*], Nasheen Nur[3]

[1] *Department of Computer Science, Birla Institute of Technology and Science, Dubai, U.A.E.*

[2] *Department. of Computer Science, Birla Institute of Technology, Dubai, U.A.E.*

[3] *Department of Electrical Engineering and Computer Science, Florida Institute of Technology, Florida 32901, U.S.A.*

*** Corresponding author:** Sujala Shetty, sujala@dubai.bits-pilani.ac.in

ABSTRACT

Learning how to read research papers is a skill. The researcher must go through many published articles during the research. It is a challenging and tedious task to go through numerous published articles. The research process would sped up by automatic summarization of scientific publications, which would aid researchers in their investigation. However automatic text summarization of scientific research articles is difficult due to its distinct structure. Various text summarization approaches have been proposed for research article summarization in the past. After the invention of transformer architecture, it has created a big shift in Natural Language Processing. The models based on transformers are able to achieve state-of-the-art results in text summarization. This paper provides a brief review of transformer-based approaches used for text summarization of scientific research articles along with the available corpus and evaluation methods that can be used to assess the model generated summary. The paper also discusses the future direction and limitations in this field.

Keywords: natural language processing; long document summarization; transformers; multi-headed attention; scientific article summarization

ARTICLE INFO

Received: 10 October 2023
Accepted: 13 December 2023
Available online: 13 March 2024

1. Introduction

The Internet's web resources, such as blogs, social media networks, news, user reviews, and webpages, are enormous sources of textual data. Additionally, a plethora of textual information may be found on the numerous archives of books, novels, news stories, legal documents, scientific research articles, and biomedical records. Users consequently have to spend a lot of time searching for the information they need. They are unable to read and grasp every word in search results. Hence, it becomes essential to summarize and condense the text resources as a result. Manual text summarization is a time consuming and expensive task. In the context of the information era, Automatic Text Summarization approaches are becoming more and more important for this purpose. An automatic text summarization system's primary goal is to generate a summary in less space that captures the essential concepts from the input content[1]. Without having to read the complete document, the automatic text summarization systems assist users in understanding the key aspects of the original document[2]. The document summaries can be categorized as systems for summarizing single or multiple documents. Single document summarization represents the gist of a single document whereas Multi Document

Summarization also known as MDS represents the gist of multiple related documents. There are two categories for the Automatic Text Summarization process: extractive and abstractive summarization[3]. In Extractive Summarization the summary is created by concatenating the most pertinent sentences from the document. Sentence scoring, sentence selection, and intermediate representation of the input text typically make up the extractive summary[4] Abstractive summarization on the other hand provides a concise summary of the source article. The summary that is generated from abstractive summarization contains new sentences that might not appear in source text[5]. Hybrid Summarization is a type of summarization that incorporates both abstractive and extractive techniques[6]. From a given corpus, it implies extracting a few sentences and creating a few novel sentences[7].

Due to the dramatic growth of information generated in recent years extracting useful information is a challenging task. The researchers find it difficult to find useful information or research articles which are relevant to their field of study. They need to spend a lot of time finding and reviewing the articles of their interest. Automatic Text Summarization of these articles will help research scholars to get the gist of the articles and find the articles of their interest. When it comes to automatically summarizing documents, there is a significant distinction between generating a summary of generic text document and scientific paper. This is due to the discourse structure of scientific papers is significantly different than generic documents. Automatic scientific article summarization differs from generic text in three ways[8]. Firstly, a specific main structure differentiates scientific publications from generic material. The major problem is usually stated in the introduction, which is followed by relevant works, methods, experiments, and findings, before the conclusion includes the findings and implications. In general, scientific articles are lengthier than non-scientific ones. The summary's objective is also not unique because scholars continue to seek out fresh insights, discoveries, and answers. A significant amount of important information found in scientific publications is included in document elements, which are "an entity apart from the running text of the document[9]. Figures, tables, and pseudocodes for algorithms are the most often used document elements in scientific papers; they include the most significant experimental findings and concepts. All these components are not present in generic text. Hence the automatic text summarization of scientific articles has to be addressed. Text summarization of research articles comes under the category of long document summarization due to the length of the document. The sequence length of scientific articles is larger than the length of the generic text article.

The technique that contributes to success in many NLP tasks including text summarization is modeling the sequential context of language[10]. Recurrent Neural Networks (RNNs)[11] memorize the sequential context in precisely constructed cells. It is challenging to scale these models to large corpora due to their sequential design, which makes computing costly[12]. The Transformer[13] architecture uses self-attention and pointwise fully connected layers in place of RNN cells because they are more computationally affordable and more parallelizable. When combined with positional encoding, Transformers can capture relative token positions that are unclear and long-range relationships. As a consequence, sentences have a coarse-grained sequence representation[10]. Due to the capability of handling long-range relationships transformers have achieved state-of-the-art results in text summarization scientific research articles.

This paper provides a systematic review of application of various transformer based pretrained models on the problem of Scientific Research article summarization using available large-scale datasets. The types of articles published in PubMed are related to the field of biomedical or life sciences. Apart from this we have also used arxiv dataset. The dataset contains papers from different domains such as computer science, mathematics, physics etc. The rest of the paper is divided into the following sections: Section 2 highlights the literature review. Section 3 describes the available datasets and transformer-based models used for research article summarization and methods for summary evaluation. Section 4 presents the result of summary evaluation and Section 5 presents a conclusion and future directions in this area.

2. Literature review

Some of the earliest methods for automatically summarizing text involved the use of statistical models that could identify and replicate the most crucial terms from the text; however, because these models couldn't comprehend the context or meaning of the words, they couldn't produce new text or paraphrase it[14]. A number of issues were brought to light by earlier research on automatic summarization systems, including the requirement for intelligent systems that are able to assess and comprehend a language's semantics at a deeper level and produce meaningful sentences or descriptions from input data that resembles human language[15].

Text summarization approaches are categorized as Extractive summarization and Abstractive summarization.

2.1. Extractive summarization approaches

The primary objective of extractive techniques is to identify and highlight the most significant sentences within the given content. The most important elements of the original text are condensed into a brief summary. For extractive summarization, a number of algorithms have been developed, each of which uses a distinct method for the extraction stage and sentence ranking which are statistical analysis based, semantics-based and graph-based methods.

One of the initially proposed methods for extractive summarization is Luhn[16]. It ranks every sentence in a given text according to the frequency of the most significant words and their relative placement in the sentence using statistical analysis. To get the final summary, the sentences with the highest scores are taken out. However, this method has a drawback in that it ignores the relationships between words or phrases and concentrates only on individual words.

One of the early methods to try to model the semantic relationships between words and extract important topics from a manuscript was latent semantic analysis (LSA)[17]. The LSA approach, which models a document as a term-sentence matrix that represents the frequency of each word in each sentence of the document, was proposed by Gong and Liu[18] for the job of text summarization. Following this, in order to rank and extract the most significant sentences, it uses singular value decomposition (SVD) to extract the most significant semantic aspects of the document.

Another extractive summarization method that overcomes some of the drawbacks of previous methods is the use of graph-based algorithms, which can produce scalable and quick summarizations. TextRank[19] is among the most well-known and oldest graph-based ranking techniques. This method starts with the document being represented as a weighted graph of sentences. The document's sentences are shown as nodes, while the connections between them are shown as edges. A relationship between two sentences shows how similar they are to one another based on the content that they both overlap. Following the creation of the graph, each sentence is ranked according to its relationships with the other sentences using the PageRank centrality algorithm[20]. In the end, a summary of the input content is created by choosing the sentences that rank highest. For the algorithm's termination, the number of extracted sentences can be configured as a user-defined parameter.

2.2. Abstractive summarization approaches

Since extractive approaches use simple heuristics to extract and concatenate the most relevant sentences without taking into account grammatical or syntactical rules, they have a significant drawback in terms of the produced text's lack of readability and coherence, which is why abstractive approaches became necessary[21]. In order to produce a summary that is both fluid and cohesive, additional contextual details regarding the text's tokens are needed. As a result, a family of models that produce new sentences in a way that mimics how a human reader would paraphrase is necessary[22].

There have been many abstractive summarization models proposed in the past. These include techniques based on graphs[23], rules[24], and semantic modeling. Yet a lot of NLP tasks have been improved by current deep learning advances, which are not utilized by these previous models. Convolutional neural networks (CNN) and recurrent neural networks (RNNs) are two examples of recent abstractive summarization techniques that build on deep learning models. LSTM and GRU, which enhance the original RNNs and are also used to generate abstractive summaries. Generative adversarial networks, or GANs, are another type of neural architecture that is not dependent on CNNs and RNNs.

2.3. Transformers

Global dependencies of sequential data are modeled by Transformer[13], a deep learning model made up of many encoder and decoder layers that use the attention mechanism[25]. In particular, based on their contextual significance, the self-attention mechanism of transformer gives various input components varying weights. During the output sequence generation process, these are encoded in hidden state layers. Transformer models also employ multi-head attention, which applies attention in parallel to the incoming data in order to identify various patterns and relationships. By encoding data into hidden layers and then decoding it to produce output, Transformer employs the encoder-decoder paradigm. Due to their unsupervised pretraining on huge datasets and supervised fine-tuning, these models are semi-supervised. Generating automatic summaries using transformers achieves state-of-the-art results[26].

3. Materials and methods

This section presents various datasets available for the task of scientific research article summarization as well as discusses the transformer-based approaches used for automatic text summarization of scientific articles along with techniques used for summary evaluation.

3.1. Scientific article summarization datasets

Corpora are required in a summarizing task in order to assess the summarization system and compare it with alternative methods. The Text Analysis Conference (TAC) (https://tac.nist.gov/) hosts a set of evaluation workshops designed to advance research in Natural Language Processing and related applications by giving organizations a place to present their findings and a large test collection[2]. The TAC 2014 (https://tac.nist.gov/2014/BiomedSumm/). Summarization Track and the 2016 Computational Linguistics Summarization Shared Task have lately provided more motivation for scientific summarization. One of these tracks from 2008 through 2011 and in 2014 was the summarization track. The TAC 2014 from the summarization track contains a dataset of research papers on 20 topics each with one reference text and other cited articles included in referenced paper. The research papers are of biomedical domain and are published by Elsevier (A Dutch publishing and analytics company). All these 20 topics contain four summaries all written by expert human annotators. Discourse facets and annotated citation texts are also included in the dataset that Cohan and Goharian used[27,28].

In 2016 CL-SciSumm (https://github.com/WING-NUS/scisumm-corpus) 2016 in computation linguistics was released to promote the research of scientific article summarization. Every paper is formatted in XML, or Extended Markup Language, and each sentence has distinct bounds.

Microsoft Academic Search is the source of the Microsoft (http://academic.research.microsoft.com) dataset. It includes details on the authors, the place of publication, the citation sentences' attached paper, and the sentences in the article abstract.

ACL Anthology Network (AAN) (https://clair.eecs.umich.edu/aan/index.php) is a network of group of individuals who are interested in finding solutions to NLP-related issues[29]. In the discipline of computational

linguistics, it is comprised of a comprehensive manually curated networked database of citations, collaborations, and summaries.

PLOS (https://plos.org/) is a dataset of 50 scientific publications. PLOS Medicine corpus is accompanied with gold standard summaries. This summary, which was authored by the editor, takes a more comprehensive view than the article abstract.

The surveyed studies' previous datasets (between 30 and 50 articles) are small in size. As a result, a sizable dataset of the 1000 most referenced articles from AAN[29] was released. Authors have cleaned and retained an average of fifteen citation sentences for every target paper. For every target paper, they also write a gold standard summary that is typically 151 words long. This data set with 1000 referenced articles is also not large enough in size. For long documents like research papers, it is difficult to prepare large sized datasets.

The sequence models used in NLP tasks such as Named Entity Recognition (NER is a technique used in information extraction that identifies name entities and assigns them to various classifications), Machine Translation (an automatic process of translating text from one language to another), Automatic Text Summarization[30] etc. require a huge number of parameters and large amounts of training along with the ground truth. Preparing the dataset of ground truth (reference summaries) along with the long documents is a challenging task. A large-scale dataset was proposed[31] for the purpose of Scientific research article summarization to enhance the research in the field of long document summarization. The two datasets were created using the repositories of arxiv.org and PubMed.org. The article's abstract served as a baseline for assessing how well the machine-generated text summarization performed. Both the datasets were preprocessed by eliminating research publications with lengths that were too lengthy or short, as well as those without a clear structure in terms of sections of articles or without an abstract in an article[31]. The figures and tables were removed from the article and only the textual data was kept. The tables and figures of the articles are removed using the regular expressions in python. The sections kept in both the datasets are only upto the conclusion. Sections after the conclusion section of the publications were removed. For arxiv dataset unique tokens such as @xmath and @xcite were used to normalize mathematical equations and citation marks respectively. However, In PubMed dataset the citation marks were removed. **Table 1** shows the statistics of arxiv and PubMed dataset provided on huggingface datasets library. The research articles that were taken from the arxiv directory were first converted to plain text using Pandoc[31] to maintain a specific structure of article sections. The size of both the datasets is significantly larger than all other previous datasets published for research article summarization. The files in the provided data sets are in the jsonlines format. Each line in the files' jsonlines format contains a json object representing a single scientific article from ArXiv or PubMed. The article, abstract and section names of all the articles are sentence tokenized. About 3-5 % of the dataset was retained for training and testing validation and apart from that the remaining entire dataset is used for training. The datasets are made open source and are accessible via GitHub and huggingface datasets library. The size of both datasets is large. arxiv contains 215K documents in total and PubMed contains 133K documents including train, test, and validation split. These datasets are provided with an abstract as a ground-truth summary.

In order to aid in the investigation of TLDR generation, Cachola et al. [32] presented SCITLDR, a brand-new dataset comprising 5,411 TLDRs from computer science publications. Another dataset named Multi-XScience, a multi-document summarization dataset was derived from scientific articles[33]. **Table 1** shows the statistics of arxiv and PubMed datasets provided on huggingface datasets library.

Table 1. Statistics of PubMed and arxiv datasets[31].

Dataset	Train	Validation	Test	Avg. length of document in words	Avg. length of summary in words
PubMed	119924	6633	6658	3016	203
arxiv	203037	6436	6440	4938	220

3.2. Transformers based models used for scientific article summarization

There are many different approaches used in Natural Language Processing to summarize scientific articles. However, the focus of this research is on the current state-of-the-art transformers-based models that are effective with large sized documents. This section covers the design of transformers and several transformers-based model architectures used for research article summaries within the arxiv and PubMed[31] datasets.

3.2.1. Architecture of transformers

The architecture of transformers[13] is entirely built on the extended idea of attention[25]; known as a multi-layer attention network. Instead of relying on Recurrent Neural Network or Convolutional Neural Network to produce an output, the Transformer architecture uses an encoder-decoder structure[34]. An input sequence is transformed into a series of continuous representations by the encoder. The decoder gets the output from both the encoder and the decoder from a prior time step and produces an output sequence. To remember the sequence of words in the input, transformer networks are dependent upon two main concepts: Self-Attention and Positional Encoding.

The architecture of transformer is depicted in **Figure 1**. Before the input text is passed in the encoder it is converted into tokens and these tokens are then turned into vectors using word embedding technique. The word embeddings are then passed to positional embeddings which helps in assigning position vector indicating the order of each vector. After the word vector is converted into positional embedding the next step is to predict the next word in the sentence. The modules that make up the encoder and decoder are layered on top of one another several times (shown in **Figure 1** as Nx). Both feed-forward and multi-head attention layers are present in the modules. **Figure 1** depicts the model architecture of transformer. The model architecture consists of a stack of N = 6 identical layers in Encoder and Decoder. Both the Encoder and Decoder have two sublayers in each layer. The first is a multi-head self-attention mechanism, and the second is a basic feed-forward network that is fully connected positionally. A third sub-layer, which performs multi-head attention over the output of the encoder stack, is inserted by the decoder in addition to the two sub-layers present in each encoder layer.

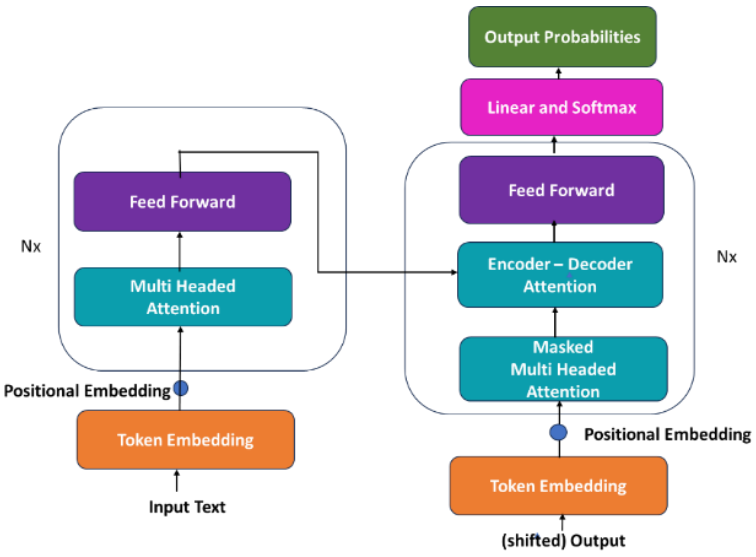

Figure 1. Model architecture of transformer[13].

Figure 2 demonstrates the model's multiple-head attention bricks. The self-attention is termed as Scaled Dot Product attention[13] in **Figure 2**. To generate the query (Q), key (K), and value (V) vectors, the input is fed into three connected layers. The attention function is computed on a set of queries bunched in matrix Q. The K and V matrices have keys and values bunched together. The output matrix is computed as:

$$\text{Attention (Q, K, V)} = \text{softmax}\left(\frac{Qk^T}{\sqrt{d_k}}\right)V$$

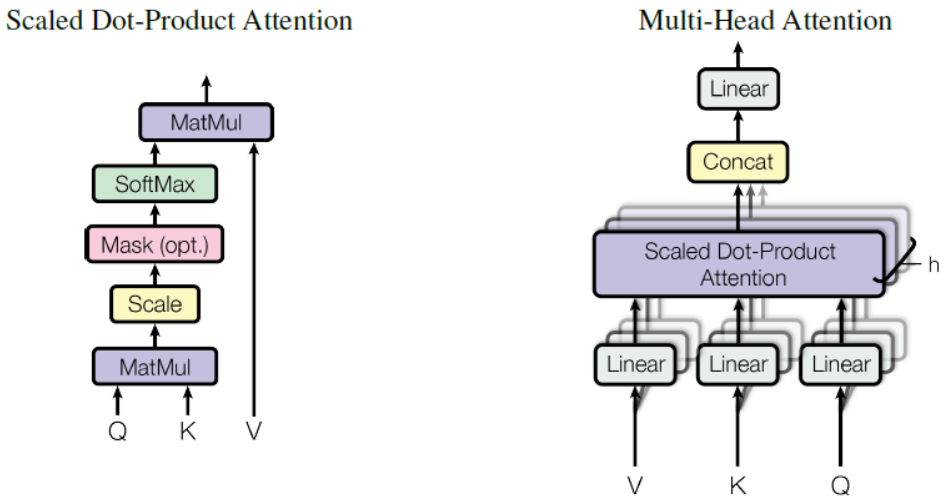

Figure 2. (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel[13].

The Q in the equation depicts a matrix containing Query, K contains key which is the vector representation of input sequence in words and V represents values. The model can focus on several different important components at once due to the multi-head attention framework.

3.2.2. Transformers based models used for research article summarization

The problem of Scientific Research Article Summarization comes under the category of Long Document Summarization. There are several transformer-based models that can handle long sequences of large documents and give state-of-the-art results. In this section below we briefly discuss some of the pre-trained transformers-based model architecture.

- **PEGASUS**—PEGASUS stands for Pre-training with Extracted Gap Sentences[35] for Abstractive Summarization. A new self-supervised aim is provided for the model, which pretrains big Transformer-based encoder-decoder models on enormous text corpora. The Pegasus Model works on two main objectives: GSG (Gap Sentence Generation) and MLM (Masked Language Modeling). The model masks the important lines in the input document. On the output end a single sequence of line/sentence is generated from the masked line. Additionally, selecting only pertinent sentences works better than selecting sentences at random. There are 12 downstream tasks of text summarization on which the model is evaluated. The model was able to produce state-of-the-art results on various datasets. The model is also able to handle the long sequence and summarize it efficiently. The two objectives of GSG and MLM worked well on various downstream summarization tasks. The model is trained on the corpus of C4[36,37].

For selecting gap sentences from a document without replacing them, the model considers three main ways: Random, Lead and Principal. The Random way of selecting sentences randomly chooses m sentences uniformly. The Lead strategy of selection selects first m sentences. Principal selection selects top m scored sentences[35]. The model was published with two variants $PEGASUS_{BASE}$ and $PEGASUS_{Large}$.

- **PEGASUS-X:** The model PEGASUS-X[38] is an extension of PEGASUS model. This model is specifically designed to support long document summarization. To accommodate inputs of up to 16K tokens, the PEGASUS-X model adds additional extended input pretraining to the original PEGASUS model. PEGASUS-X performs well on lengthy input summarizing tasks equivalent to much bigger models. The model requires only a small number of additional parameters and training without model parallelism. To summarize long documents the model makes use of three key strategies:

During pretraining, we employ a Global-Local architecture[38] with block staggering, a significant number of global tokens, and huge block sizes.

1) A further level of extended input pretraining with 4096 token inputs for 300,000 steps was performed.
2) Depending on the objective, input sequences up to 16384 input tokens were expanded.

Apart from using all the parameters of PEGASUS two new parameters were introduced: Global Token Embeddings[38] and a layer for layer normalization. Due to its mechanism of handling large sequences the model can achieve comparable results on the research article summarization for PubMed and arxiv datasets.

- **T5:** T5 stands for Text-to-Text Transfer Transformer[37]. The model is based on the concept of Transfer Learning[39]. The architecture of the T5 model is same as the transformer model having 12 blocks of Encoder and Decoder. Each of these contains multi headed attention, feed forward neural network and encoder-decoder attention which is elective. The model differs from a general transformer architecture in two ways:

  1) The representation of Input and Output
  2) The Training dataset: Colossal Clean Crawled Corpus(C4)[37]

Before feeding in the input to the model, input is processed with a prefix of a specific text to text downstream tasks like text summarization, language translation, text classification etc. The denoising aim and C4 dataset were used to pretrain the model, which was based on a BERT-base size encoder-decoder transformer. The three main objectives of the model are: BERT-style Masked language modeling, language modeling and deshuffling. The deshuffling is a strategy in which the input is randomly shuffled, and the model tries to predict the original text. Different variants of the model with different numbers of parameters are made available. The model generates an abstractive summary of the input. In case the problem of summarization focuses on multiligual dataset then T5 variant mT5[40] can be used.

- **Long T5**: LongT5[41] is an extended version of the T5 model. The pretraining strategies for Long T5 are taken from PEGASUS model. The model architecture works on two key concepts of attention mechanism:

  1) Transient-Global attention—A mechanism that allows to attend all the input words.
  2) Local attention—A mechanism that allows to attend only subset of input words.

The model gives good results on the tasks of long sequence inputs. The model can handle input length of 16K tokens. Although LongT5 is an extended version of T5 but it does not use a prefix for a specific task. Due to the nature of model's capability of handling long sequences the model works really well on arxiv and PubMed datasets and gives comparable results.

- **BART:** Bidirectional and Auto Regressive Transformers is what the BART[42] model is known as. BART is a denoising autoencoder that was trained as a sequence-to-sequence model. This implies that a refined BART model can accept one text sequence as input and output another text sequence. Text that is "corrupted" or "noisy" in the BART training data will be mapped to text that is clean or original. The noising strategies used for BART are Masking Tokens, Deleting Tokens, Infilling Text and Rotation of the Document. Same as BERT model[43] BART uses bi-directional Encoder and GPT[44] like autoregressive decoder. The model limits on handling very large length of input sequences and does not work well with sequence data of large length. The model has different variants like BART-base and BART-large available to use as required**.** In case the problem of summarization focuses on multiligual dataset then BART variant mBART[45] can be used.
- **BART-LSG:** BART-LSG[46] stands for BART for Long Sequence Generation. To enable the model to handle large sequences the model works on three mechanisms. Attending Global Tokens: Previous

research has suggested enhancing block-sparse attention with a small collection of "global tokens" that attend to the entire sequence and so enabling long-range interactions in the encoder[47]. This works really well in-terms of attending large sequences. Strided Attention Window: For simple introduction of long-range connections in local attention models, sliding-attention with overlap is used. Each token's receptive field would exponentially grow due to the stacked layers in the encoder. Pooling Layer: The pooling operations are used in the models so that it can have fewer key and value pairs. BART-LSG model gives the SOTA results on the text summarization of the scientific articles.

- **Big Bird:** BigBird is a sparse-attention transformer that can handle substantially longer sequences than other transformer-based models like BERT[48]. The model has adopted three different attention mechanisms in the architecture: Random Attention, Sliding Window Attention and Global attention.

    1) **Random Attention**: A (i, ·) =1 for r randomly selected keys indicates a sparse attention system in which each query attends over r random numbers of keys.
    2) **Sliding Window Attention:** A window of width W that restricts the query node's ability to attend to only its peers inside the key nodes and the key node's immediate neighbors inside the window. This is known as a sliding window attention.
    3) **Global Attention:** This attention mechanism incorporates the importance of global tokens. A token attends every other token in an input sequence.

Applying global, random, and sparse attention has been demonstrated to be computationally more efficient for longer sequences while roughly achieving the same results as complete attention. Big Bird has demonstrated enhanced performance on a variety of long document natural language processing tasks, including question answering and summarization, because of its capacity to handle longer contexts. To summarize all the different transformers-based models **Table 2** is provided below. It compares the pretraining objective, pretraining corpora, total number of parameters and supported token length.

Table 2. Summary table of model details.

Model	Pretraining Objective	Corpora	No. Parameters	No. of Tokens
$PEGASUS_{LARGE}$	Masked Language Modeling and Gap Sentence Generation	C4, Hugenews	567 M	1024
PEGASUS-X	Masked Language Modeling and Gap Sentence Generation	C4, Hugenews	568 M	16k
$BART_{LARGE}$	Denoising objective	BOOKCORPUS, CCNEWS, OPENWEBTEXT, STORIES	406 M	1024
$BART—LS_{LARGE}$	T5 span Denoising, Pegasus—Primary Sentence Prediction, Model based Denoising	C4, Real News, Stories, C4	406 M	16 k
$T5_{LARGE}$	Masked Language Modeling and De-shuffling sentences	C4 Corpus	220 M	512
$LongT5_{LARGE}$	Principle Sentence Generation	C4 Corpus	780 M	16 k
Big Bird	Masked Language Modeling	Books, CCNews, Stories, Wikipedia	-	4096, can be extended to 16 k

3.3. Methods of summary evaluation

This section describes different techniques used to evaluate machine generated summaries. There are different approaches used for the evaluation of summaries generated using the NLP models and comparing it with the ground truth summaries.

1) **ROUGE:** The full form of ROUGE is Recall Oriented Understudy for Gisting Evaluation[49]. It is a collection of measures for assessing automatic text summarization. It compares an automatically generated summary to a collection of reference summaries, which are human produced. ROUGE is one of the widely used approach to evaluate the summaries. Suppose a reference summary is created by number of human annotators and denoted as reference summary set (RSS); then the ROUGE-N Score can be calculated as:

$$ROUGE - n = \frac{\Sigma_{C \in RSS} \Sigma_{gram_n \in C} Count_{match}(gram_n)}{\Sigma_{C \in RSS} \Sigma_{gram_n \in C} Count(gram_n)}$$

where $Count_{match}(gram_n)$ is the maximum number of *n*-grams occurring in a ground truth summary and a reference summary and $Count(gram_n)$ is number of *n*-grams occurring in the reference summary.

2) **BLEU Score**: For an automatic evaluation of machine generated summary BLEU[50] score is used. The score always lies between zero to one where zero indicates no overlap between the ground truth and machine generated summary and 1 indicates a perfect overlap between the ground truth and machine generated summary.

The BLEU score is calculated as:

$$\text{BLEU} = \underbrace{\prod_{i=1}^{4} min(1, \exp(1 - \frac{\text{reference-length}}{\text{output-length}}))}_{\text{brevity penalty}} \underbrace{(\prod_{i=1}^{4} precision_i)^{1/4}}_{\text{n-gram overlap}}$$

where,

$$precision_i = \frac{\sum_{\text{snt} \in \text{Cand-Corpus}} \sum_{i \in \text{snt}} min(m^i_{cand}, m^i_{ref})}{w^i_t = \sum_{\text{snt'} \in \text{Cand-Corpus}} \sum_{i' \in \text{snt'}} m^{i'}_{cand}}$$

Here,

m^i_{cand} xi is count of i-gram between ground truth and reference summary.

m^i_{ref} xxx is count of i-gram in reference summary.

$w^i_t m_{max}$ is total number of i grams in ground truth summaries

As highlighted in the formula it consists of two parts;

1) Brevity Penalty—A penalty that penalizes the machine generated summary if it's too short compared to the ground truth.
2) N-gram overlap: Count of overlap of unigrams, bigrams or trigrams with the ground truth summary.

4. Results and discussion

This section highlights the results of summarization on the two datasets. The most widely used metric used for the evaluation of automatic text summarization is ROUGE Score because it offers a means of evaluating the quality of machine-generated summaries in comparison to reference summaries. The overlap of n-grams is taken into consideration, which aids in encapsulating the summary's key points. **Tables 3** and **4** compare the ROUGE Scores of different state-of-the-art transformers-based models used for scientific article summarization.

In **Table 3** text summarization results of arxiv dataset are highlighted. BART-LS outperforms compared to all other models. In **Table 4** ROUGE scores of PubMed summarization on different pretrained models are highlighted. It shows that for ROUGE-1 BART-LS gives the best result and for ROUGE-2 and ROUGE-3 Long T5 gives the best result.

Table 3. ROUGE scores of arxiv summarization on different pretrained models.

	arxiv		
Approach	R-1	R-2	R-L
PEGASUS-Large	44.6	17.2	25.8
PEGASUS-X	50.0	21.8	44.6
BART–LS	50.2	22.1	45.4
Long T5	48.3	21.9	44.2
BigBird	46.6	19.02	41.7

Table 4. ROUGE scores of PubMed summarization on different pretrained models.

	PubMed		
Approach	R-1	R-2	R-L
PEGASUS-Large	45.09	19.5	27.4
PEGASUS-X	51.0	24.7	46.6
BART – LS	50.3	24.3	46.3
Long T5	50.2	24.7	46.6
BigBird	46.3	20.6	42.3

The BART model for long sequence generation model achieves the highest result compared to all other models. This is mainly due to the pretraining objective on which BART-LS is trained on; as well as the corpora used for pretraining. BART-LS and LongT5 both the models are trained with large number of parameters and are specifically trained for handling long sequence input. The reason why BART-LS outperformed in comparison with all other models is also due to its architecture of attending global tokens as well as introducing strided window attention.

5. Conclusion and future works

This paper presents a review of the state-of-the-art models used for the scientific article summarization on two big datasets: arxiv and PubMed. The transformers-based models are able to capture the long range context dependencies, hence they perform better than all other models previously proposed for text summarization. According to the results based on ROUGE score, BART-LS surpasses all the transformer-based models presented in the literature in terms of summary generated for the arxiv and PubMed datasets. This is due to its ability of efficiently handling large length sequential data of scientific research articles. Although both datasets only include the text content of the scientific article and ignore the mathematical equations, images/graphs and tables of results and comparisons which hold the most important information about the scientific article. All the approaches have used text-to-text summarization approach. From the survey we have found out that almost every model's performance is assessed by ROUGE score only. Hence there is a large dominance of the use of ROUGE score for summary evaluation. So, proposing new metrics for summarization evaluation remains an open area of research.

Future research is needed to annotate the dataset which should consider the images and tables to generate the summary which gives a good overview of research article that includes the important findings of the paper as well. There is also a need of new evaluation methods that can be used to assess the performance of the model. As per our knowledge so far there is no approach developed for scientific article summarization that considers the images, tables, graphs and mathematical expressions. This is an open area where researchers can extend their work in this domain.

Acknowledgments

The authors express their profound gratitude to all the reviewers for their comments and suggestions on improvements of the article.

Conflict of interest

The authors declare no conflict of interest.

References

1. Radev DR, Hovy E, McKeown K. Introduction to the Special Issue on Summarization. Computational Linguistics. 2002, 28(4): 399-408. doi: 10.1162/089120102762671927
2. Hima Bindu Sri S, Dutta SR. A Survey on Automatic Text Summarization Techniques. Journal of Physics: Conference Series. 2021, 2040(1): 012044. doi: 10.1088/1742-6596/2040/1/012044
3. Kadry S, Yong H, Choi J. Applied sciences Improved Text Summarization of News Articles Using GA-HC. 2021.
4. Joshi A, Fidalgo E, Alegre E, et al. SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. Expert Systems with Applications. 2019, 129: 200-215. doi: 10.1016/j.eswa.2019.03.045
5. Wang Q, Liu P, Zhu Z, et al. A Text Abstraction Summary Model Based on BERT Word Embedding and Reinforcement Learning. Applied Sciences. 2019, 9(21): 4701. doi: 10.3390/app9214701
6. Sharma G, Sharma D. Automatic Text Summarization Methods: A Comprehensive Review. SN Computer Science. 2022, 4(1). doi: 10.1007/s42979-022-01446-w
7. Rush JW, Alexander M., Sumit Chopra. A Neural Attention Model for Sentence Summarization Alexander. Conference on Empirical Methods in Natural Language Processing, 2017. 5(3): 379-389.
8. Teufel S, Moens M. Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status. Computational Linguistics. 2002, 28(4): 409-445. doi: 10.1162/089120102762671936
9. Bhatia S, Caragea C, Chen HH, et al. Specialized Research Datasets in the CiteSeerX Digital Library. D-Lib Magazine. 2012, 18(7/8). doi: 10.1045/july2012-bhatia
10. Zhang JG, Li JP, Li H. Language Modeling with Transformer. In: Proceedings of the 2019 16-th International Computer Conference on Wavelet Active Media Technology and Information Processing, December 2019. pp. 249–253.
11. Tomas Mikolov SK, Karafiat M, Burget L. Recurrent neural network based language model. Proceedings of the Annual Meeting of the Association for Computational Linguistics, September 2020. pp. 8093–8104.
12. Luo H, Jiang L, Belinkov Y, et al. Improving neural language models by segmenting, attending, and predicting the future. ACL 2019—57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, pp. 1483–1493, 2020.
13. Vaswani A, Shazeer N, Parmar N. Attention is all you need. ArXiv 2023; arXiv:1706.03762.
14. Anitha J, Raahavi M, Rehapriadarsini M, Sudarshana SS. Abstractive Text Summarization. Journal of Xidian University, 2020. 14(6): 854–857. doi: 10.37896/jxu14.6/094
15. Nenkova A. Automatic Summarization. Foundations and Trends® in Information Retrieval. 2011, 5(2): 103-233. doi: 10.1561/1500000015
16. Luhn HP. The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development. 1958, 2(2): 159-165. doi: 10.1147/rd.22.0159
17. Deerwester S, Dumais ST, Furnas GW, et al. Indexing by latent semantic analysis. Journal of the American Society for Information Science. 1990, 41(6): 391-407. doi: 10.1002/(sici)1097-4571(199009)41: 6<391: : aid-asi1>3.0.co, 2-9
18. Gong Y, Liu X. Generic text summarization using relevance measure and latent semantic analysis. In: SIGIR Forum (ACM Special Interest Group on Information Retrieval), pp. 19–25, 2001.
19. Mihalcea R, Tarau P. TextRank: Bringing order into texts. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004—A meeting of SIGDAT, a Special Interest Group of the ACL held in conjunction with ACL 2004, vol. 85, pp. 404–411, 2004.
20. Page LMR, Brin S. The Page Rank Citation Ranking: Bringing Order to the Web.
21. Saggion H, Poibeau T. Automatic Text Summarization: Past, Present and Future To cite this version: HAL Id: hal-00782442 Automatic Text Summarization: Past, Present and Future. 2016.
22. El-Kassas WS, Salama CR, Rafea AA, et al. Automatic text summarization: A comprehensive survey. Expert Systems with Applications. 2021, 165: 113679. doi: 10.1016/j.eswa.2020.113679
23. Ganesan K, Zhai CX, Han J. Opinosis: A graph-based approach to abstractive summarization of highly redundant opinions. In: Coling 2010—23rd International Conference on Computational Linguistics, Proceedings of the Conference, vol. 2, no. August, pp. 340–348, 2010.

24. Genest PE, Lapalme G. © In: 50th Annual Meeting of the Association for Computational Linguistics, ACL 2012—Proceedings of the Conference; July 2012; Jeju Island, Korea. pp. 354–358.
25. Bahdanau D, Cho KH, Bengio Y. Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings, pp. 1–15, 2015.
26. Giarelis N, Mastrokostas C, Karacapilidis N. Abstractive vs. Extractive Summarization: An Experimental Review. Applied Sciences. 2023, 13(13): 7620. doi: 10.3390/app13137620
27. Cohan A, Goharian N. Scientific document summarization via citation contextualization and scientific discourse. International Journal on Digital Libraries. 2017, 19(2-3): 287-303. doi: 10.1007/s00799-017-0216-8
28. Cohan A, Goharian N. Scientific article summarization using citation-context and article's discourse structure. In: Conference Proceedings—EMNLP 2015: Conference on Empirical Methods in Natural Language Processing; 1 September 2015; pp. 390–400.
29. Jha R, Abu-Jbara A, Radev D. A system for summarizing scientific topics starting from keywords. ACL 2013—51st Annual Meeting of the Association for Computational Linguistics. In: Proceedings of the Conference, vol. 2, pp. 572–577, 2013.
30. Khurana D, Koli A, Khatter K, et al. Natural language processing: state of the art, current trends and challenges. Multimedia Tools and Applications. 2022, 82(3): 3713-3744. doi: 10.1007/s11042-022-13428-4
31. Cohan A, Dernoncourt F, Kim DS, et al. A discourse-aware attention model for abstractive summarization of long documents. In: NAACL HLT 2018—2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies—Proceedings of the Conference, vol. 2, pp. 615–621, 2018.
32. Cachola I, Lo K, Cohan A, Weld DS. TLDR: Extreme summarization of scientific documents. Find. Assoc. Comput. Linguist. Find. ACL EMNLP 2020. pp. 4766–4777. doi: 10.18653/v1/2020.findings-emnlp.428
33. Lu Y, Dong Y, Charlin L. Multi-XScience: A large-scale dataset for extreme multi-document summarization of scientific articles. In: EMNLP 2020—2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, pp. 8068–8074, 2020.
34. Gupta A, Chugh D, Anjum, et al. Automated News Summarization Using Transformers. Lecture Notes in Electrical Engineering, vol. 840, pp. 249–259, 2022.
35. Zhang J, Zhao Y, Saleh M, Liu PJ. PEGASUS: Pre-Training with extracted gap-sentences for abstractive summarization. In: 37th International Conference on Machine Learning, ICML 2020, vol. PartF16814, pp. 11265–11276, 2020.
36. Dodge J, Sap M, Marasović A, et al. Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus. In: EMNLP 2021—2021 Conference on Empirical Methods in Natural Language Processing, Proceedings, no. Table 1, pp. 1286–1305, 2021.
37. Raffel C. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 2020. 21: 1–67.
38. Phang J, Zhao Y, Liu P. Investigating Efficiently Extending Transformers for Long Input Summarization. 2022.
39. Zhuang F, Qi Z, Duan K, et al. A Comprehensive Survey on Transfer Learning. Proceedings of the IEEE, 2021, 109(1): 43–76. doi: 10.1109/JPROC.2020.3004555
40. Xue L, Constant N, Roberts A, et al. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In: NAACL-HLT 2021—2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, pp. 483–498, 2021.
41. Guo M, Ainslie J, Uthus D, et al. LongT5: Efficient Text-To-Text Transformer for Long Sequences. In: Findings of the Association for Computational Linguistics: NAACL 2022—Findings, pp. 724–736, 2022.
42. Lewis M, Liu Y, Goyal N, et al. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880, 2020.
43. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL HLT 2019—2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies—Proceedings of the Conference, vol. 1, no. Mlm, pp. 4171–4186, 2019.
44. Brown TB. Language models are few-shot learners. Adv Neural Inf Process Syst, 2020.
45. Liu Y, Gu J, Goyal N, et al. Multilingual Denoising Pre-training for Neural Machine Translation. Transactions of the Association for Computational Linguistics. 2020, 8: 726-742. doi: 10.1162/tacl_a_00343
46. Xiong W, Gupta A, Toshniwal S, et al. Adapting Pretrained Text-to-Text Models for Long Text Sequences. 2022.
47. Ivgi M, Shaham U, Berant J. Efficient Long-Text Understanding with Short-Text Models. Transactions of the Association for Computational Linguistics. 2023, 11: 284-299. doi: 10.1162/tacl_a_00547
48. Zaheer M. Big bird: Transformers for longer sequences. Adv Neural Inf Process Syst, 2020.
49. Lin CY, Hovy E. Automatic evaluation of summaries using N-gram co-occurrence statistics. In: Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003. June 2003, pp. 71–78.

50. Wentzel G. Funkenlinien im Röntgenspektrum. Annalen der Physik. 1922, 371(23): 437-461. doi: 10.1002/andp.19223712302