Computer Speech and Language 71 (2022) 101276

Contents lists available at ScienceDirect

Computer Speech & Language

journal homepage: www.elsevier.com/locate/csl

Deep reinforcement and transfer learning for abstractive text summarization: A review

Ayham Alomari [a], Norisma Idris [a,*], Aznul Qalid Md Sabri [a], Izzat Alsmadi [b]

[a] Department of Artificial Intelligence, Faculty of Computer Science and Information Technology, Universiti Malaya, 50603 Kuala Lumpur, Malaysia
[b] Department of computing and cyber security, Texas A&M, San Antonio, Texas, United States

ARTICLE INFO

Keywords:
Abstractive summarization
Sequence-to-sequence
Reinforcement learning
Pre-trained models

ABSTRACT

Automatic Text Summarization (ATS) is an important area in Natural Language Processing (NLP) with the goal of shortening a long text into a more compact version by conveying the most important points in a readable form. ATS applications continue to evolve and utilize effective approaches that are being evaluated and implemented by researchers. State-of-the-Art (SotA) technologies that demonstrate cutting-edge performance and accuracy in abstractive ATS are deep neural sequence-to-sequence models, Reinforcement Learning (RL) approaches, and Transfer Learning (TL) approaches, including Pre-Trained Language Models (PTLMs). The graph-based Transformer architecture and PTLMs have influenced tremendous advances in NLP applications. Additionally, the incorporation of recent mechanisms, such as the knowledge-enhanced mechanism, significantly enhanced the results. This study provides a comprehensive review of recent research advances in the area of abstractive text summarization for works spanning the past six years. Past and present problems are described, as well as their proposed solutions. In addition, abstractive ATS datasets and evaluation measurements are also highlighted. The paper concludes by comparing the best models and discussing future research directions.

1. Introduction

Text Generation (TG) tasks are one of the most NLP challenging tasks because it requires an automated understanding of the text and an accurate semantic and lexical analysis of its words [1]. These tasks include Machine Translation (MT), Image Captioning (IC), Automatic Text Summarization (ATS), among others.

ATS is the process of generating a short text that covers the main parts of a longer document. A good summary considers important aspects, such as readability, coherency, syntax, non-redundancy, sentence ordering, conciseness, information diversity, and information coverage [2]. The two main methods of ATS are extractive (selection and combination), and abstractive (paraphrasing). The abstractive type of summarization is more challenging and requires understanding at a higher-level to ensure the fluency, saliency, coherency, information correctness, and novelty of the resulting summary [3]. The combination of both methods recently showed promising results by first extracting most of the highlights and then reformulating them to create a final abstractive summary.

Recently, Deep Learning (DL) approaches have shown significant progress in various NLP applications. Specifically, deep neural sequence-to-sequence models have been used extensively in various TG tasks.

Deep encoder-decoder architecture is the most widely used method for applying sequence-to-sequence models. Despite significant

* Corresponding author.
E-mail address: norisma@um.edu.my (N. Idris).

https://doi.org/10.1016/j.csl.2021.101276

Available online 10 August 2021

improvements in the results of ATS applications, deep sequence-to-sequence models suffer from several problems. For example, they cannot handle long-term dependencies efficiently. Moreover, they cannot be parallelized due to their sequential nature. Other issues include low-novelty, exposure-bias problem, loss/evaluation mismatch, and lack of generalization.

Subsequently, Reinforcement Learning (RL) approaches were used to overcome some problems of deep sequence-to-sequence models and improve the quality of their results. RL's research in ATS has mainly focused on developing novel rewards to allow models to refer to principles other than ground-truth summaries.

Nowadays, NLP research is witnessing a golden era by leveraging Transfer Learning (TL) and Pre-Trained Language Models (PTLMs). The self-attention architecture, Transformer [4], is the first transduction model based on the self-attention mechanism to represent its source data and results without the use of sequence-based Recurrent Neural Networks (RNNs) or Convolutional Neural Networks (CNNs). The self-supervised architecture, Transformer, has solved the previous problems of sequence-to-sequence models. It can efficiently handle long-dependencies using a parallel-based design that allows parallelization, which drastically increases

Table 1
Text Summarization Categories.

Category	Types	Description	Example
Form	Extractive Summary	Highlighting	[27]
	Abstractive Summary	Rewriting	[28]
	Compressive Summary	Deleting unimportant words/sentences from an extractive summary	[24]
Length	Short Document Summary	Generating short summaries from short documents	[18]
	Long Document Summary	Generating long summaries from long documents	[29]
Output	Text Summary	The summary is generated in a normal text	[28]
	Other Formats Summary	e.g., numbered, bulleted summary, word graph	[30]
Input	Single-Document Summary	Summarizing one document	[28]
	Multi-Document Summary	Summarizing clusters of documents of the same topic	[24]
Context	Generic Summary	Summaries based on standard generic datasets (news datasets)	[16]
	Domain-Specific Summary	Summaries based on some domain dataset as scientific datasets	[31]
	Query-based Summary	Summaries based on a query, such as search engines	[32]
Language	Multi-Lingual Summary	The source document is in several languages, and the summary is also generated in these languages	[33]
	Mono-Lingual Summary	The language of source and target documents are the same	[28]
	Cross-Lingual Summary	The source and summary documents are written in two different languages	[34]
Personal	Personalized Summary	Contains the specific information that the user desires and prefers	[35]
	Update Summary	Considering the user has the basic information about a topic, summarizing only the current updates regarding that topic	[36]
Scientific Articles Summarization	Title Generation	Based on the article's abstract as an input	[37]
	Abstract Generation	Based on the article's body as an input	[38]
	Introduction summarization	Extracting salient sentences in the introduction	[39]
	Citation summarization	Generating related work section	[40]
	Highlight statements	Generating highlight statements of articles	[41]
	Multi Article Summarization	Summarizing multi scientific articles	[42,43]
	Scientific Survey Generation	Generating a survey of many papers on a specific problem or domain	[44]
	Presentation slides Generation	Automatically creating presentation slides of a specific article	[45]
Medical Reports Summarization	Radiology Reports Summarization	Automatically summarizing of radiology findings into natural language impression statements	[46]
	Clinical Abstractive Summarization	Extracting clinical findings from clinical reports	[47]
Others	Email-based summary	Summarizing, sorting, and archiving email conversations	[48]
	Guided summary	Incorporating different kinds of important information from the source text to guide the generation process, control the output, and improve the quality of results	[49]
	Sentiment-based summary	Summarizing user's opinions, feelings, or reviews based on social networking sites, forums, blogs, etc.	[50]
	Comparative Summarization	Summarizing the differences, similarities, or hierarchical relations among multiple documents	[51]
	Evolutionary Timeline Summarization	Generating a summary for dates in a timeline for the evolution of events	[52]
	Microblog summarization	A multi-document summarization which has been widely studied in Information Retrieval (IR)	[53]
	Spoken data summarization	Such as summarizing meetings and conservations with fixed templates and procedures	[54]
	Opinion Summarization	Generating summaries with sentiment and without redundancy	[55]
	Community Answer Summarization	Summarizing millions of question-answer pairs contributed by community users, such as QUORA and StackOverflow	[56]
	Student responses to reflection prompt	Summarizing student responses to reflection prompts for large courses such as introductory STEM, MOOCs.	[57]
	Movie Script Summarization	Highlighting major scenes representative of the story and its progression	[58]
	Entity description in knowledge graphs	To link structured data such as Google's Knowledge Graph, Facebook's Open Graph, and W3C's Linked RDF Data with webpages text and form a unified Web	[59]
	Source Code summarizing and description	Writing brief, natural language descriptions of source code behaviour throughout the program's run-time	[60]

computation speed and generates more accurate summaries with higher scores in the ROUGE metric [5], which is the most widely used abstractive ATS metric.

Based on this architecture, many large-scale PTLMs are trained on large corpora of texts developed by large organizations, such as Google's BERT [6], PEGASUS [7], T5 [8], and Switch [9], Facebook's BART [10], and Open AI's Generative Pre-Training (GPT) [11], GPT-2 [12], and GPT-3 [13]. Essentially, PTLMs are trained to estimate the combined probability of any token arrangements. PTLMs have shown remarkable success in a wide range of natural language understanding and generative tasks. The great benefit is that these PTLMs can be fine-tuned to solve a vast amount of downstream NLP tasks, including abstractive text summarization, utilizing the learnt universal language representations.

The contributions of this study can be summarized as follows:

1 *ATS overview*: We provide a full review of the ATS, which includes:
a A taxonomy with different categories.
b A brief history of models' evolution.
c Evaluation measurements review.
d Comparisons of datasets.
e Comparisons and relationship of the ATS and MT research fields.
2 *Models comprehensive review*: We collected an abundance of resources on the main topics of this study and provide a comprehensive review of SotA research works, starting with Deep neural sequence-to-sequence models, then RL approaches, and finally TL architectures, including PTLMs.
3 *Challenges*: We analyze the past and current challenges faced by researchers in the areas this study focuses on as well as their proposed solutions.
4 *Comparisons*: We provide different kinds of comparisons for models that have been investigated from different perspectives: theoretical, practical, and the experimental results. Subsequentially, the top-performing models are highlighted.
5 *Future trends*: We suggest and discuss possible future research directions.

The following sections are organized as follows; Section 2 describes the background of ATS, DL, RL, and TL. In Sections 3, 4, and 5, details of how the DL, RL, and TL approaches are being applied in abstractive ATS are discussed. Details of abstractive ATS datasets are described in Section 6. Abstractive ATS evaluation metrics and comparisons between the top-perfrming models are presented in Section 7. Section 8 discusses some abstractive ATS research challenges, presents the relationship between ATS and MT research, and highlights trends in future research. Section 9 concludes this review.

2. Background

2.1. ATS taxonomy

There are two main types of ATS which are extractive and abstractive. In the extractive type of ATS, the summary concatenates the original document's most important sentences without any modification. On the other hand, abstractive summaries are completely new sentences with the same meaning as the ideas of the original document. Technically, the extractive type can be loosely defined as binary classification, i.e., whether the source article's token is to be included in the generated summary. On the other hand, abstractive summarization requires more sophisticated approaches to generate a novel text. Extractive method is considered to be more simplistic, thus, its research is being explored more widely in contrast to the abstractive ATS research.

Other categories can be categorized based on different perspectives, including length, input, output, purpose, language, and more. Table 1 summarizes the different categories and types of ATS.

Some types of ATS can be combined, while others are incompatible. For example, several researchers nowadays demonstrate that the hybrid model of extractive and abstractive types of ATS achieves better results than the stand-alone abstractive ATS model using sequence-to-sequence techniques [14–18], and RL approaches [19–23]. Another example, Wang et al., [24] combined compression, query-based, and multi-document summarizations into one summarizer. On the other hand, search engines, which are query-based summarizations, cannot summarize multi-documents even with advanced information retrieval tools.

Generally, most researchers work on the English language, news context, and text output types of ATS. This study focuses on research works related to abstractive, short- and long- document, single-document, textual, generic, and English types of ATS. From Table 1, some of these categories were collected by [2,25], and [26] survey papers.

2.2. Why abstractive ATS?

In recent years, vast quantity of documents is available on the web, and the number is growing exponentially over time, which creates the problem of information overload. Hence, reading only the most important information while skipping the minor parts in a short time will save the time and effort for readers to read various "summarized" documents for a given field. For example, in the field

of scientific articles, a large number of research papers are published daily, for example, Pubmed publishes 1.5 papers per minute[1]. Moreover, hot research areas, such as COVID-19 research, require speeding up the process of summarizing details and findings to bridge the gap between researchers.

Currently, ATS is one of the most popular research areas in NLP. ATS can be categorized into two main types. The first being extractive, i.e., highlighting, and the second being abstractive, i.e., paraphrasing. Clearly, abstractive summarization is more challenging as it deals with higher-level aspects such as semantic representation, surface realization, and content organization [25]. Moreover, abstractive ATS requires an automated understanding of the input text beyond the meanings of words and sentences, which is a challenging task. Generally, the main requirements for abstractive summaries include fluency, saliency, coherency, information correctness, and novelty [3].

Compared to extractive summarization, the abstractive type of ATS is distinguished by its readability, consistency, and conciseness. However, it is more complex, requires higher computational costs, and has the tendency to produce trivial summaries with factual mistakes, redundant information, and major information loss. However, recent advances in abstractive ATS research have shown impressive results that overcome these shortcomings, particularly with the use of modern approaches such as PTLMs and the knowledge-enhanced mechanism. Additionally, a multi-document summarization experiment made by Genest et al., [61] to compare human-written summaries showed that the performance of pure extractive summaries is very low compared to abstractive summaries.

Recently, there has been an increasing interest in utilizing both the extractive and abstractive types of ATS, by combining them together to enhance the quality of the resulting final abstractive summary. This mixed-method is applied first using DL-based techniques; pointer-generator [14–18] and knowledge-enhanced [49,62,63], which are discussed in Section 3.1.3 and Section 3.1.5, respectively and then by RL-based fusion techniques [19–23,64], discussed in Section 4.2. In this article, based on the final result generated, we use the term "abstractive ATS models" to refer to both mixed models and pure abstractive ATS models as the end result of both models is the same, which is an abstractive summary with new phrases.

In general, humans tend to generate abstractive forms using the mixed approach when writing their summaries. They first extract the most salient parts and kinds of information from the text and then paraphrase them into a readable and coherent form. There is no doubt that human-generated summaries are more reliable and efficient because human cognition and experience play a primary role in the summarization process. Also, human summarizers can efficiently interpret a document, prioritize important sentences, and create diverse coherent paraphrased versions of the summary [65]. However, manual text summarization is time and effort consuming.

These concerns sparked interest in the research of abstractive ATS. That is why massive amounts of research have been conducted in the abstractive text summarization field, covering all its aspects. However, abstractive ATS research still requires more effort in attempting new strategies to reach the level of human-generated summaries.

2.3. Deep neural sequence-to-sequence

In recent years, deep neural sequence-to-sequence models have proven to be the best models for TG tasks, including MT, IC, Question Answering (QA), and ATS. Neural sequence-to-sequence models are used to map the input sequence from one form to another to generate the desired results. For example, in MT [66] the text sequence is changed from one language to another. In ATS [28], the input is a long text sequence, and the output is a short length sequence.

The Encoder-Decoder architecture is the main approach for modelling sequence-to-sequence models. The encoder network, which is a set of nodes, takes the input sequence timesteps (tokens) and produces some encryption (hidden representation) which can be understood by the decoder network, which is also a collection of nodes, to generate the desired output based on that representation.

Specifically, in neural sequence-to-sequence models applied by encoder-decoder architectures, the input sequence is understood by the encoder and converted into a fixed-length vector representation (at the word-, sentence-, or document-level). Then it is forwarded to a decoder that translates the vector to generate a new version of the sequence. Fig. 1 describes the basic encoder-decoder architecture. For TG tasks, usually, we have two sentences; a source sentence, x, with n tokens, and a target sentence, y, with m tokens such that $x = (x_1, x_2, x_3, \dots, x_n)$ and $y = (y_1, y_2, y_3, \dots, y_m)$. The pair $(x,y) \in (X,Y)$ is called the sentence pair, where X is the source domain and Y is the target domain.

For abstractive ATS, x and y tokens come from the same vocabulary, and $n > m$. In architectures based on Recurrent Neural Networks (RNNs), discussed in the next section, the tokens of x are fed into the encoder sequentially to produce a sequence of hidden state vectors, hi:

$$h_i = RNN(x_i,\ h_{i-1}) \tag{1}$$

On the other hand, the decoder takes these encoder hidden vectors, $h = \{h_1, h_2, \dots, h_n\}$ as feature representations of the whole input sentence. It generates the output sentence by training to estimate the conditional probability $P(y|x,\theta)$, where θ is the parameter to be learned by the model, using the log-likelihood objective function O as:

$$O(\theta, (X,Y)) = \sum_{(x,y)\in(X\ ,Y)} logP(yx,\ \theta) \tag{2}$$

However, this scenario can only work efficiently for short sequences and may struggle when dealing with long sequences, such as

[1] https://tac.nist.gov/2014/BiomedSumm/index.html

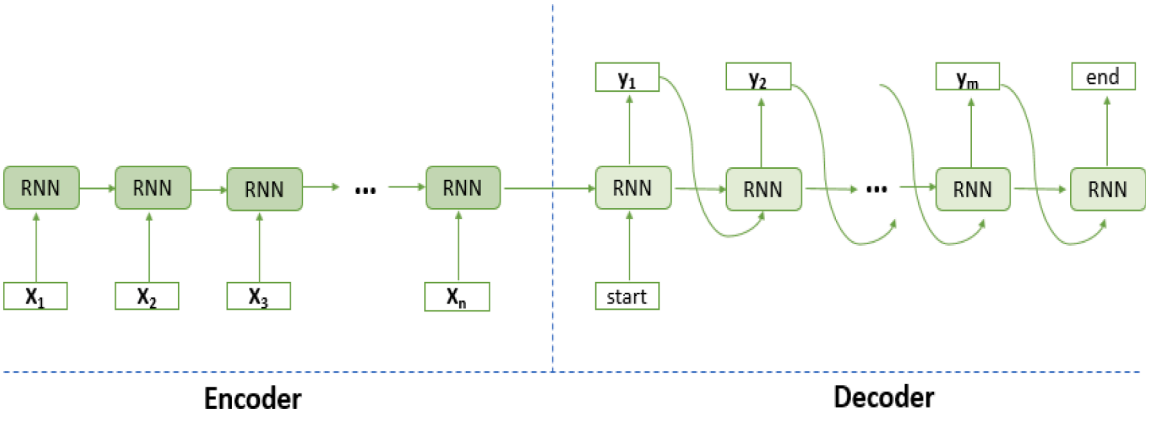

Fig. 1. Basic Encoder-Decoder Architecture.

summarizing or translating long articles, such as scientific papers.

Basic deep neural sequence-to-sequence models of [67] and [68] were the first models that applied on the task of English to French translation. This basic architecture, illustrated in Fig. 1, has been strengthened over the past years to solve different issues using various mechanisms, such as the mechanisms of attention, copy, coverage, and others. These mechanisms greatly improved TG tasks outcomes, especially MT and ATS tasks. Section 3 describes in detail these mechanisms in the field of ATS.

2.4. Recurrent, convolutional, or attention?

Sequence data are time-based data where the arrangement of its components, i.e., timesteps, is critical, including number sequences, text sequences, video frame sequences, and audio sequences. For example, the order of the words that compose a sentence plays an essential role in understanding the whole sentence.

RNNs [69] are primarily designed in a sequence nature, so they are the most appropriate DL architectures for encoding and processing sequence data such as text. RNNs can handle sequence dependencies between input tokens (characters, words, sentences, etc.) as well as the variable-length input for unstructured data such as text. Moreover, RNNs are able to generate feedback to prior layers to maintain memory inputs. Therefore, the output of some layers can be fed back into the inputs of previous layers, giving RNNs the ability to analyse sequential data.

Unidirectional RNNs can only capture the context based on the token's history, and they do not consider all the context required from the future aspect. This may lead to misleading predictions and, as a result, incorrect information generation. In human perception, to predict an unknown word in a sentence, the context of both sides is taken into consideration to ensure the correct prediction. Therefore, using bidirectional RNN allows each hidden state to be aware of the contextual information from both directions, i.e., past and future contexts, in order to improve the results. Al-Sabahi et al., [70] proposed a bidirectional encoder-decoder model that impressively improves the results of generated summaries and overcomes the problem of generating wrong information.

However, vanilla RNNs suffer from vanishing and exploding gradient problems during training, and therefore are unable to model long-term dependencies when the sequence length increases. Some variants of RNN, called canonical architectures, i.e., Long Short-Term Memory (LSTM) [71] and Gated Recurrent Units (GRU) [67], have been found to model long-term dependencies more accurately and overcome the gradient-vanishing problem [72] using gating mechanisms. Note that GRU is a simplification of LSTM with simpler resources, less complexity and operating cost, and faster training and execution. However, the exploding gradient problem could be solved via other techniques such as gradient norm clipping strategy [73].

Nevertheless, LSTM and GRU still struggle to handle very long-term memory modeling in very long documents, and thus cannot copy or recall information from a long-distance past. Recently, the Rotational Unit of Memory (RUM) [74] has been proposed as a representation unit for RNNs that unifies the properties of unitary learning and associative memory to handle memory copying and memory recall tasks better than LSTMs and GRUs. RNN based on RUM module models can better handle long-term dependencies enhancing ROUGE results when replacing LSTM and GRU. In particular, models that incorporate RUM units lead to larger gradients during training resulting in more stable training and better convergence [75].

Most researchers choose to use RNNs for NLP tasks, especially for language modelling tasks. RNN variants have gained popularity in modelling sequence data due to their sequential nature and their ability to capture unlimited and unstructured contexts. However, training with RNN is limited to parallel processing on sequence elements due to their temporal dependencies between hidden states [76]. Furthermore, they must also preserve the hidden state of the whole past [77] because of their sequential dependence nature [78], and they are still restricted to handle long-term dependencies for very long documents due to their linear path length between the output and any of the input tokens [65].

Therefore, the feedforward architecture, Convolutional Neural Networks CNNs [79], especially hierarchical CNNs, can be used instead as fundamental networks to model sequential data [28,76–78,80].

CNNs could efficiently address previous RNN's issues by computing each element in the sequence in parallel during training and

evaluation, which, as a result, will lead to improved efficiency and performance of the model [80]. Other advantages of using CNNs with sequence-to-sequence models include their ease of optimizing, their linear computational complexity, and more efficient propagation of gradient signals [78].

Dauphin et al. [81] apply gated CNNs to language modelling and prove that CNNs allow parallelization over sequential data and achieve competitive results with robust gated recurrent LSTM models. Moreover, Zhang et al., [77] apply a hierarchical CNN framework for abstractive ATS, outperforming most RNN models, as shown in Section 7.2.1.

For ATS, both RNN and CNN architectures are used to create abstractive and extractive summaries. For example, IBM has developed an abstractive summarization model based on RNN [16]. Also, Facebook has developed a similar model based on CNN [28].

More recently, significant improvements in the quality of results and ROUGE scores have been obtained using the Transformer architecture [4]. This fully attention-based architecture has been developed based on attention mechanisms only without any use of RNN or CNN nodes. Various models can utilize Transformer, such as PTLMs, as an encoder and/or decoder.

Essentially, Transformer is used to model similarities between input tokens regardless of their positions in parallel by attending each token in the input sequence independently using the self-attention mechanism. Therefore, Transformer efficiently solved the problems of RNN, i.e., sequential nature and long-term dependencies, as discussed in Section 5.

As a result, models that are based on the attention-based architecture, Transformer, outperformed RNN- and CNN-based sequence-to-sequence models and showed SotA results on various NLP tasks in less training time, including abstractive ATS as illustrated in Section 7.2. However, more efficient Transformers and attentions have recently been introduced to overcome the Transformer limitations of quadratic memory complexity, the large number of operations required, and the fixed-length context prerequisite. These improvements are discussed in Section 5.1.

Moreover, combining the advantages of attention and recurrence has recently emerged as a research trend. Many kinds of work have been done in NLP research to modify the architecture of internal Transformers and combine it with recurrent units, i.e., GRUs and LSTMs. This process showed significant improvements in terms of speed and results in various NLP tasks [82–86], as shown in Section 5.2.4.

2.5. Beyond DL

The popularity of DL nowadays mainly arises after the availability of numerous annotated datasets and the availability of computational resources that facilitate the use of parallel processing through modern Single Instruction-Multiple Data (SIMD) hardware accelerators such as GPUs and TPUs. GPUs and TPUs make the processing faster, cheaper, and more powerful.

Despite the significant improvements resulting from using deep sequence-to-sequence models and their various mechanisms described in Section 3.1 in NLP tasks, RNN-based models are still limited to various weaknesses. For example, they cannot work in parallel due to their sequential design nature. Thus, they can't leverage the full power of GPUs and TPUs. Moreover, they cannot efficiently handle long-range dependencies of long document inputs, resulting in low-quality summaries. Other problems include low-novelty, exposure-bias problem, loss/evaluation mismatch, and lack of generalization.

Therefore, integrating RL and TL approaches can efficiently solve these problems, as discussed in Section 4 and Section 5, respectively.

RL rewarding can help improve abstractive ATS results in its various aspects, such as ROUGE scores, quality, factual consistency, novelty, readability, coherency, syntax, non-redundancy, sentence ordering, conciseness, information diversity, information coverage, saliency, and entailment [87]. This is discussed in Section 4.

Moreover, as discussed in Section 5, utilizing Transformers' parallelization allows models to take advantage of the full power of GPUs and TPUs and increase training speed even with massive datasets. PTLMs pre-train the Transformers on huge corpora and transfer their acquired knowledge to downstream tasks such as abstractive ATS. As a result, the universal and rich semantic and contextual features of word embeddings acquired by PTLMs prove that they can assist in enhancing the quality of the resulting summaries. This widespread success of Transformers and PTLMs for various NLP tasks, including abstractive ATS, makes them the backbone of recent NLP research.

2.6. Abstractive ATS models evolution history

In the past years, many attempts have been made to generate acceptable and readable ATS. Luhn [88] has made the first attempt to generate an automated text summary in 1958, followed by several works. Some of them have led to acceptable and moderate outcomes, and some have failed. However, all these works are far from human summaries quality levels.

In early research work, non-neural models were applied with handcrafted features to generate ATS. These models include statistical-based approaches [89], topic-based approaches [90], graph-based approaches [91], and discourse-based approaches [92].

Next, the era of neural-based models emerged with machine-learning-based architectures, including supervised techniques [93,94] and unsupervised techniques [95].

Subsequently, the field of abstractive ATS flourished and became more appealing to researchers when the use of DL increased, especially deep sequence-to-sequence models, and demonstrated significant improvements in other similar areas of NLP such as MT and IC. These models use techniques and mechanisms that can solve several research problems and greatly improve the results. Examples include attention mechanism [28], pointer-generator mechanism [16], coverage mechanism [18], bidirectional architectures [70], and knowledge-enhanced mechanism [49], as discussed in detail in Section 3.

Besides, RL approaches showed further improvement in results when combined with deep sequence-to-sequence models. Some of

these RL-based models address various common limitations that cannot be solved by DL-based models using policy-learning [96]. Others propose fused extractive-abstractive models which notably enhance the results, while others develop new abstractive ATS related rewards and metrics, as discussed in detail in Section 4.

Finally, due to TL's breakthrough in the NLP field, particularly the significant success and popularity of Transformer architecture [4] and the massive PTLMs [6,7,10–13], the quality of the results has improved dramatically and become closer to human-generated summaries with rapid computations. To this end, several pre-training objectives appropriate for abstractive ATS have been proposed. This is discussed in detail in Section 5.

Fig. 2 summarizes the evolution of text summarization models over the past years.

2.7. Short- vs long- document ATS

Most of the early and current abstractive ATS research works focus on short-length abstractive ATS by applying experiments on datasets of news articles such as DUC, Gigaword, CNN.DM, and XSum. For example, as discussed in Section 6, the task of the Gigaword corpus is to produce a headline given the first sentence of a document. DUC has two main tasks, generating short and very short summaries. CNN/DM, the most popular dataset used to compare research progress in the field of abstractive ATS, is designed to generate short summaries (about 50 words) in light of a news article (approximately 700 words). Finally, XSum consists of an average of ~400 words inputs accompanied by one-sentence summaries of almost 20 words. Refer to Table 8 for more details.

Recently, there has been increased interest in summarizing very long documents, which has led to the development of three popular long-document datasets for the abstractive ATS task: arXiv [97] , PubMed [97], and BIGPATENT [98]. Those consider scientific articles and patent descriptions consisting of thousands of words accompanied by summaries of hundreds of words. Refer to Table 9 for more details.

Long summarization is more intricate as it requires more complex hardware requirements and more efficient approaches. Therefore, if the models developed for short-length summaries are applied to very long documents, it would certainly lead to trivial and incoherent summaries that involved many problems.

[20,99,100] use RL approaches to generate more coherent summaries for long documents by dividing the document into several encoders or submodels, as discussed in Section 4.1. However, the complexity of such models will be increased dramatically when dealing with much longer documents as these models still handle the entire input document at once, precluding the possibility of parallelization [75].

In particular, as discussed in Section 2.4, LSTM [71] and GRU [67] were essentially proposed to handle long-term dependencies in long documents more efficiently. However, they still struggle with very long documents containing thousands of words such as scientific papers due to the linear path length between the output and any of the input tokens. To this end, RUMs [74], CNNs [79], and Transformers [4] can be alternatives because they can better deal with long-term dependencies [75]. However, RUMs are limited to

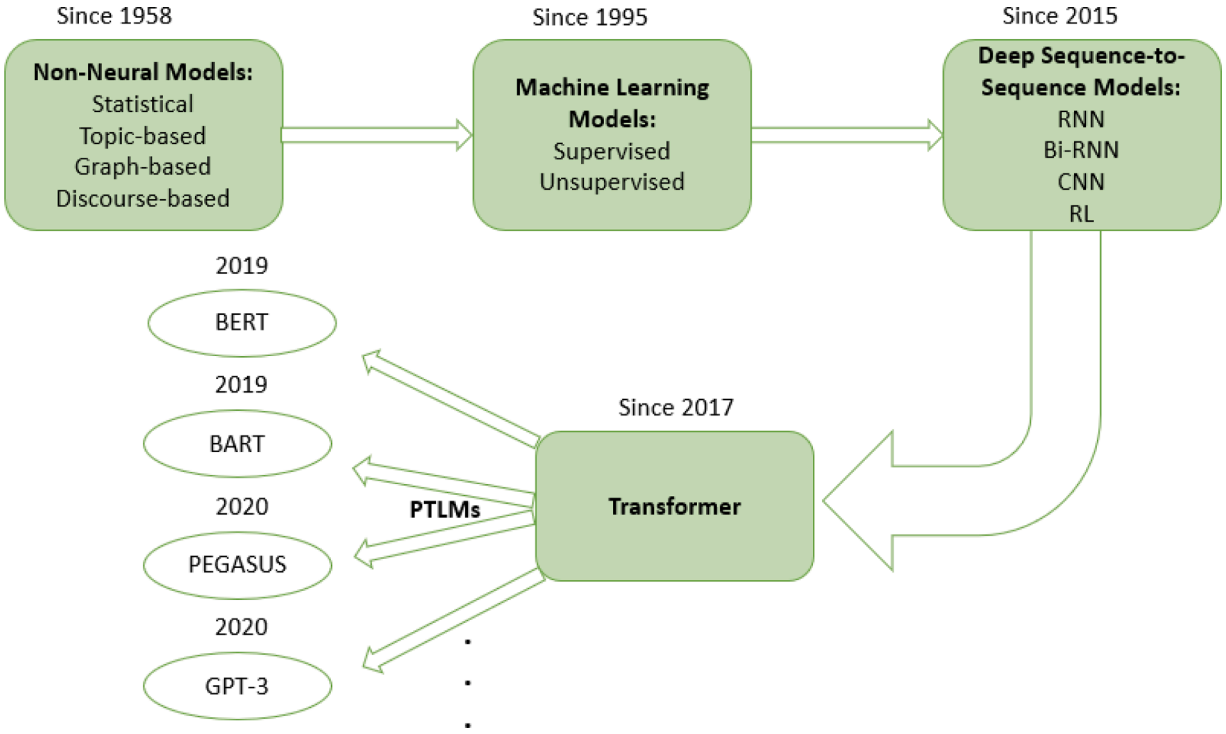

Fig. 2. The History of the Evolution of ATS Models.

parallel processing.

CNNs and Transformers can work in parallel and be used to learn very long-range dependencies efficiently in very long documents due to their logarithmic or constant path length, making gradient flow much easier [65]. However, Transformer is preferable because its self-attention mechanism is able to model similarities between the words regardless of their positions in parallel, as discussed in Section 5.1. Consequently, [7,29,65,75] designed Transformer-based models to handle long documents achieving SotA results. These models are discussed in Section 5.2.

Ultimately, this study explores the research progress of both short- and long-document abstractive ATS. SotA models for both genres are discussed in Section 7.2.1 and Section 7.2.2.

3. Abstractive ATS using DL approaches

Deep neural sequence-to-sequence models [68] using the encoder-decoder architecture, shown in **Fig. 1**, have proven to be the best choice to model TG tasks, such as abstractive ATS to shorten a long input text. However, utilizing this basic framework results in generating trivial summaries with several issues, such as difficulties in dealing with long-term sentence dependencies and Out of Vocabulary (OOV) words. Other issues include generating low-quality summaries that lose some essential details or include inaccurate factual details with repetitive phrases. Therefore, ATS researchers found some techniques and mechanisms to overcome these challenges and improve the quality of results, such as attention mechanism, copy mechanism, coverage mechanism, and knowledge-enhanced mechanism.

3.1. Mechanisms

The following subsections discuss some of the prominent deep sequence-to-sequence techniques and mechanisms that have been added to the basic encoder-decoder architecture and showed significant improvements in the abstractive ATS results.

3.1.1. Attention

The idea of attention is inspired by the human understanding of an object or text by focusing only on certain parts. For example, instead of paying attention to all parts when looking at an image, a person will concentrate only on specific details to better understand that image. Similarly, it is possible to allow a model to focus only on specific kinds of information that are considered most important in achieving a better understanding.

In sequence-to-sequence modelling, encoding the entire source text into a fixed-length vector requires large memories and leads to the problem of long-term dependencies that negatively affects the performance of the model. Alternatively, the model can utilize the attention mechanism which dynamically searches for the most relevant parts by using a dynamically changing context in the decoding process [14]. Therefore, before generating a word, the attention mechanism is used to compute word weights to determine how much attention should be paid to each input word. This idea was first exploited by [66] to translate English statements into French by means of automatic alignment, followed by image caption generation by [101], short text conversation by [102], and then abstractive ATS by Rush et al., [28]. Moreover, to improve results by better handling name-entities and long sentences, Luong et al. [103] suggested global and local attentions for the MT task. The global attention pays attention to all the words of the source input, regardless of its length, while the local attention focuses only on a selective subset of input positions at each time step.

For abstractive ATS, Rush et al. [28] utilize the basic attention mechanism of [66]'s probabilistic model by proposing a neural soft

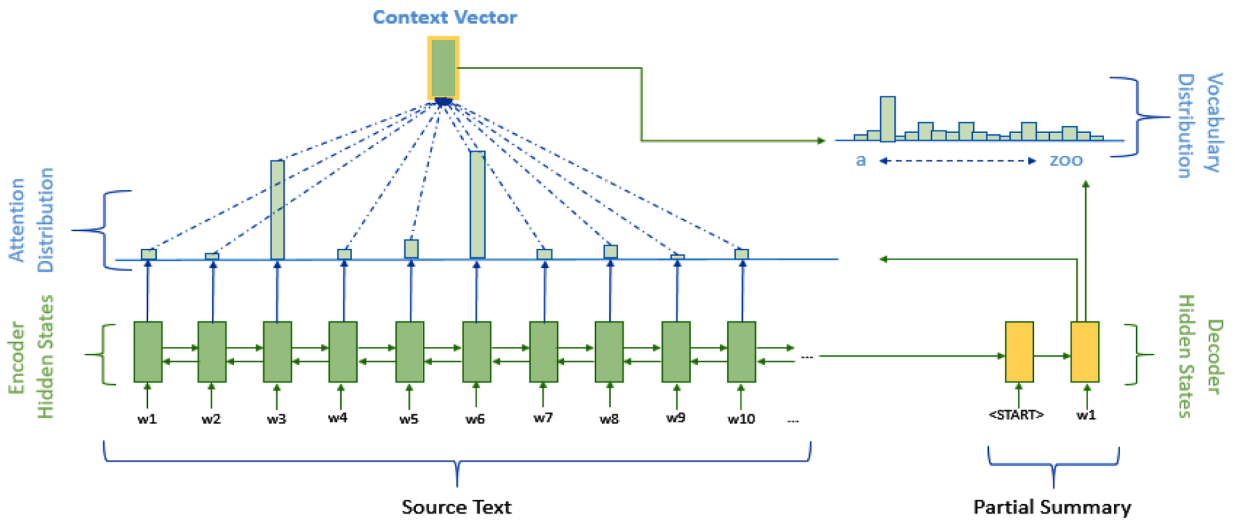

Fig. 3. Basic Sequence-to-Sequence model with attention [18].

attention model with a control layer. The model is concerned only with the words most likely to be added in the generated summary rather than looking at each word in the long input sequence. This mechanism saves more memory, improves performance, and achieves more efficient results.

In formal terms, from [66]'s point of view, as shown in **Fig. 3**, the following formula calculates the weighted sum of the encoder hidden states, i.e., the context vector :

$$cv = \sum_i a_i^t h_i \tag{3}$$

where a^t is the attention distribution over the source words, which is calculated using the softmax function as:

$$a^t = softmax\left(v^T tanh(W_h h_i + W_s s_t + b_{att})\right) \tag{4}$$

where *v, Wh, Ws*, and b_{att} are the parameters to learn, and s_t is the decoder state in timestep *t*.

Rush et al.'s model [28] consists of a convolutional attention-based encoder and a feed-forward decoder. Their experiments have achieved the best results at that time on two sentence-level and abstractive headline generation corpora, Gigaword and DUC-2004. Besides, multiple research efforts have conducted on abstractive ATS proposing several types of attention, such as attention over a hierarchical model [104], hierarchical attention [16], forced attention [105], graph-based attention [3], intra attention /self attention [99], and pointer-generator multi-head attention [106].

Recently, with the emergence of Transformers, many forms of attention have been proposed to enhance performance and reduce memory and computations. One of the commonly used forms of attention is the multi-head self-attention used by the standard Transformer [4], which uses multiple self-attention layers running in parallel to learn various attention distributions over the source sequence. Moreover, Sparse fixed and stride attentions are also proposed for better dealing with long sequences and reduce memory complexity. Furthermore, other types of attention are proposed and discussed in Section 5.1.

3.1.2. Beam search

During testing, sequence-to-sequence models generate fully-formed word sequences by searching over output sequences using either greedy search or beam search algorithms [107]. In most sequence-to-sequence models, after determining the probabilities of source salient tokens by the attention layer, the beam-search algorithm can be used during decoding to favour hypotheses, attend salient entities [108], shrink the search space, and reduce computational complexity [62].

More specifically, the greedy search algorithm tends to directly select the word with the greatest probability at each time step, which results in an incoherent sentence with inaccurate order of the words generated, i.e., low-readability [78]. On the other hand, the beam search algorithm ensures that coherent and more readable sentences are generated with the correct word order. The graph-based beam search algorithm considers (*m*) highest possibilities of the next word, which is called beam size and then selects the proposition that has the best probabilities combined.

However, diversity and local optima search are the main limitations of the beam search algorithm. For the first problem, Vijayakumar et al., [109] proposed Diverse Beam Search (DBS) that yields diversified outputs by optimizing for a diversity-augmented objective. Multiple ATS research efforts have been performed using DBS to produce more informative summaries [110] and increase abstractive ATS novelty [111]. For the second problem, He et al., [62] suggested a prediction model (value network) based on RL approaches that can predict long-term rewards to be received in the future.

3.1.3. Pointer-generator / copy

In abstractive ATS attentional sequence-to-sequence models [16,28,104,112–115], the decoder may select inaccurate phrases from its vocabulary during the paraphrasing process. This may result in a summary containing incorrect information. Often, summaries must contain constant kinds of information from the original text, such as quotes and named-entity objects, to preserve the meaning of the sentence.

Another problem with attentional sequence-to-sequence models is the inability to handle rare or unseen words, which results in generating the annoying unknown token (UNK) when locating such words, the problem which is called the OOV words problem. This problem occurs when the system encounters words during the testing phase that are not previously seen in the training phase, that is, out of its vocabulary.

As a result, the summaries generated from the attentional sequence-to-sequence models may include some inaccurate words that may change the meaning of the sentence or produce the UNK token if the model doesn't find any corresponding words in its vocabulary. Therefore, the researchers worked to solve these problems by proposing the copy mechanism using pointing networks that allow the model to borrow some words from the source input text when needed, resulting in a combination of extractive and abstractive summary.

Vinyals et al. paper [116] was the first to introduce this idea, proposing a pointer network over sequence-to-sequence learning in general, which can copy but not generate. Moreover, [117] proposed a copiable model for the MT task to solve the problem of OOV words.

For abstractive ATS, Gu et al., [14] and Gulcehre et al., [15] were the first researchers that pointed the problem of OOV word on two different datasets, LCSTS (Chinese language) and Gigaword (English language), respectively. Then, Nallapati et al. [16] introduced a hierarchical attentional bidirectional model with a switching generator/pointer network to solve the OOV word problem for the abstractive headline generation task. The switching mechanism in the pointer generator model can balance the transcription of source

words and creativity. Subsequently, Zeng et al. [17] enhanced the copy mechanism using the read-again mechanism by reducing the vocabulary size to achieve more performance and produce a better OOV word representation using a softmax pointer mechanism.

See et al. [18] enhanced previous work on the copy mechanism by proposing a hybrid pointer generator over the attention model that is capable of copying and generating words. Their model showed improved results over CNN/DM dataset and increased accuracies in the generated summaries. Specifically, Fig. 4 depicts how the proposed hybrid pointer generator is integrated with the attention-based sequence-to-sequence model shown in Fig. 3.

Specifically, to calculate the generation probability, P_{gen} for timestep t:

$$p_{gen} = \sigma\left(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr}\right) \tag{5}$$

Where *wh, ws, wx,* and *bptr* are parameters to be learned, and $p_{gen} \epsilon [0,1]$

$$P(w) = p_{gen} P_v(w) + \left(1 - p_{gen}\right) \sum_{i:w_i=w} a_i^t \tag{6}$$

In this case, if w is not included in the source text, then $\sum_{i:w_i=w} a_i^t$ is *zero*, but if it is not included in the vocabulary, then $P_v(w)$ equals to *zero*. Therefore, this demonstrates the ability of the model to efficiently overcome the OOV words problem.

However, copy-mechanism-based models generate summaries that include many repeated sentences. For this problem, the coverage mechanism [18] has been proposed. Moreover, since the model is trained to copy specific words from the source text, this reduces the novelty levels of the generated abstractive summaries. This issue is efficiently solved using RL approaches discussed in Section 4.

3.1.4. Coverage

Attentional sequence-to-sequence models with copy mechanisms [14–18] have efficiently addressed OOV words and inaccurate details problems by taking advantage of combining extractive and abstractive objectives. However, these models failed to solve the repeated information problem. Therefore, it was imperative to develop a mechanism that keeps track the information that has already been generated in the summary. Therefore, the coverage mechanism has been developed.

Tu et al. [118] and Mi et al. [119] revealed the idea of coverage mechanism by addressing two major problems of Bahdanau's attention-based MT models [66], namely, over-translation, where some words are translated multiple times, and under-translation, where some words have never been translated. Tu et al. [118] appended a coverage vector to the attention model in intermediate representations to be updated after each attentive reading by a decoder to track attention history and help adjust future attentions.

Afterward, See et al. [18] adapted Tu et al.'s model [118] to the ATS problem by adding a coverage vector, *ct*, to the attention-model to avoid repeated attention by allowing the model to pay more attention to not summarized pieces and less attention to the parts actually summarized. In fact, ATS coverage mechanism is more flexible and less restrictive than MT coverage which requires a one-to-one translation ratio and a uniform coverage. Unlike previous work that focused on generating headlines based only on one or two sentences, the distinct challenge of this work is multi-sentence summarization, which means longer-text abstractive summarization using the CNN/DM dataset.

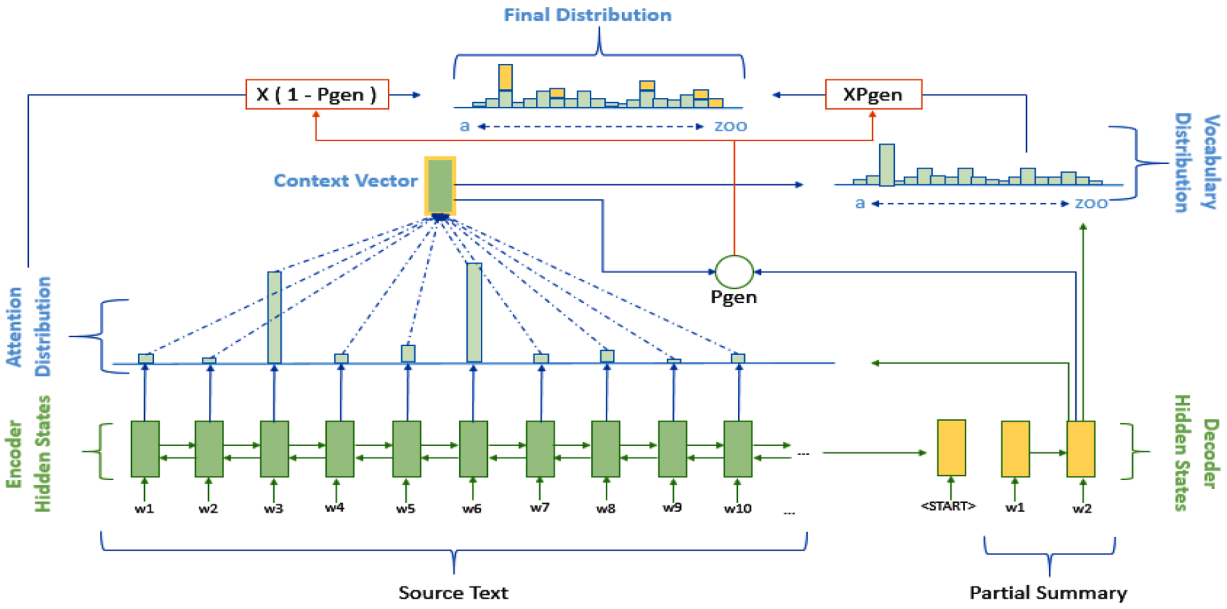

Fig. 4. Sequence-to-Sequence model with attention and hybrid pointer generator [18].

Table 2
A comparison of ATS DL-based sequence-to-sequence models. SGD stands for Stochastic Gradient Descent.

Research	Year	Dataset	Encoder	Decoder	Optimizer	Beam size	Main Contribution(s)	Problem(s) Solved
Rush et al., [28]	2015	Gigaword DUC	CNN	Feed-Forward NN	SGD	50	Attention Mechanism	Long-term Dependencies with Long sentences.
Lopyrev [113]	2015	Gigaword DUC	RNN	RNN	RMSProp	_	Attention Mechanism with RNN for encoder and decoder	Long Sequences.
Chopra et al., [114]	2016	Gigaword DUC	CNN	RNN	SGD	10	Attention mechanism (RNN Decoder)	Long Sequences.
Gulcehre et al., [15]	2016	Gigaword	Bidirectional GRU-RNN	Uni-Directional RNN	Adadelta [134]	_	Copy mechanism Pointing method	Rare/Unknown words (OOV problem).
Gu et al., [14]	2016	LCSTS	Bidirectional RNN	RNN	SGD	10	Copy mechanism (COPYNET)	Rare/Unknown words (OOV problem).
Nallapati et al., [16]	2016	Gigaword DUC CNN/DM	Bidirectional GRU-RNN	Uni-Directional GRU-RNN	Adadelta	5	Hierarchical attention Switching Generator/Pointer Temporal attention Large-Vocabulary Trick CNN/DM dataset	Rare/Unknown words (OOV problem). Inaccurate factual details.
Takase et al., [115]	2016	Gigaword DUC	AMR-RNN	AMR-RNN	SGD	_	Abstract Meaning Representation (AMR)	The usefulness of incorporating structural syntactic and semantic information into novel attention-based encoder-decoder models.
Miao and Blunsom [105]	2016	Gigaword	Variational Autoencoder	Variational Autoencoder	Adam	5	Forced-attention mechanism Variational autoencoders generative model Forced sentence compression Pointer network RL	Training the discriminative models on a big labelled dataset.
Zeng et al., [17]	2016	Gigaword DUC	GRU LSTM	LSTM	SGD	10	Read again mechanism Copy mechanism	Suboptimal representations of words. Large vocabulary. Slow decoding time.
Chen et al., [120]	2016	CNN LCSTS	GRU	GRU	SGD Adadelta	5	Coverage mechanism (Distraction) for long text	Low performance of long text summarization.
See et al., [18]	2017	CNN/DM	Bidirectional LSTM-RNN	Uni-Directional LSTM-RNN	Adagrad [135]	4	Hybrid Pointer Generator Coverage Mechanism	Inaccurate factual details. Repeated statements.
Al-Sabahi et al., [70]	2018	CNN/DM	Bidirectional LSTM	Bidirectional LSTM	SGD	4	LSTM-based bidirectional encoder-decoder model Reverse the input sequence order in the encoder Bidirectional beam search algorithm	Considering the token context only from the history side. Generating wrong information.
Li et al. [63]	2018	CNN/DM	Bidirectional LSTM	LSTM	Adagrad	-	Knowledge-enhanced mechanism Incorporate keywords Guide the generation process and control the summary's content	Hard to control generation Lack of key information
Cao et al., [49]	2018	Gigaword	Bidirectional GRU	GRU	Adam	6	Knowledge-enhanced mechanism Incorporate fact descriptions as triples and tuples relations Reduce generating fake information Improve summaries faithfulness and informativeness	Generating summaries with fake facts
Zhang et al., [77]	2019	Gigaword DUC CNN/DM	CNN	CNN	Adadelta	5	Utilize CNNs to allow parallelization over text data	Sequential nature of RNNs

Specifically, the coverage vector, c^t, can be shown as the sum of attention distributions over all previous decoder tokens:

$$c^t = \sum_{taptara_dummy_entity_aptara_dummyx27;=0}^{t-1} a^{taptara_dummy_entity_aptara_dummyx27;} \tag{7}$$

Here, the Bahdanau's attention equation (Eq.3) is modified as follows:

$$e_i^t = v^T \tanh\left(W_h h_i + W_s s_t + w_c c_i^t + b_{att}\right) \tag{8}$$

To penalize attendance to the same parts, a new coverage loss function is defined as follows:

$$cl_t = -logP\left(w_t^*\right) + \lambda \sum_i \min\left(a_i^t, c_i^t\right) \tag{9}$$

Where λ is a hyperparameter.

Later, Chen et al. [120] use the coverage mechanism by addressing higher-level abstraction by summarizing long text, i.e., thousands of words, at the document-level. In their work [120], the researchers have utilized the coverage mechanism from a different point of view: they used history to impose distraction, i.e., coverage, to avoid repetition, in order to gain a better understanding of the whole document by distracting the model to traverse between the document contents and to improve the model performance. They implement their model on CNN and LCSTS datasets.

3.1.5. Knowledge enhanced / generation guide

Deep sequence-to-sequence models along with its various techniques and mechanisms have significantly enhanced the summaries generated in terms of informativeness, readability, and quality. However, these models only rely on the input text representation to generate the results. This leads to many problems such as generating unfaithful summaries that lack basic information from the original text, generating semantically irrelevance summaries of the original text information, generating summaries with fake information, and difficulty in controlling the generation process and the summary length.

The knowledge-enhanced mechanism (a.k.a generation guide mechanism), which was first proposed by He et al., [62] for MT, aims to introduce prior knowledge to the model by incorporating other useful information to be entered with the input sequence to guide the generation process and control the output to boost the performance of the results. Generally, the knowledge-enhanced mechanism encodes some forms of useful information, then the resulting representation is fed into the decoder along with source text embeddings and attention context vectors.

Different kinds of guidance information can be extracted from the source text and entered into the model to increase the informativeness, readability, and faithfulness of its generated summaries. Examples of such information include keywords ([63,121,122]), length tokens ([122–125]), salient sentences [121,122,126]), facts descriptions and semantic dependencies represented by graphs and relations ([49,122,127,128]), and template-based reference summaries ([129])

Li et al. [63] utilize this mechanism for the abstractive ATS task by incorporating input text keywords as key information to guide the generation process to control the summary's content to include the key information of the source text. In their work, an extractor model extracts the article keywords first, then the generation-guide mechanism is used in another model to predict the summary's long-term value. This mechanism ensures that the source text keywords are included in the resulting summary. Moreover, Cao et al., [49] solve the problem of generating summaries with fake facts by extracting fact descriptions from the source text first, representing them by triples and tuples relations, and then encoding them into the model to control the output to be aware of these extracted descriptions during the generation process. Their results show a noticeable reduction in generating fake information by 80%, and an improvement in summaries faithfulness and informativeness.

Recently, as shown in Section 5.2.4, this idea is further exploited by incorporating PTLMs and extra multi-guided information to enhance the accuracy and quality of the resulting summaries [121,122]. As a result, these models show impressive enhancements in ROUGE scores, as shown in detail in Section 5.2.4 and Section 7.2.1.

As a branch of the knowledge-enhanced mechanism, the two-stage learning strategy [130,131] has recently attracted the research field of abstractive ATS. The candidate summaries generated by one model are used as input to another model to ultimately generate the best output. This model combination mechanism significantly boosts the performance of abstractive ATS models as recently shown by [132,133], discussed in Section 5.2.4.

3.2. DL Research Contributions and Comparisons

During the past six years, a vast amount of deep sequence-to-sequence research work has been conducted for abstractive ATS resulting in various models with different features and parameters. Table 2 compares some chosen research works that utilizes DL-based approaches in practical terms and summarizes their contributions.

4. Abstractive ATS using RL approaches

Research work on RL has grown in recent years. RL can achieve good performance in many areas, such as pattern recognition, speech recognition, computer vision, and NLP [136]. In RL, an agent interacts with the environment and learns, by trial and error, the optimal policy for sequential decision-making to maximize a future cumulative reward. This reward can be a developer-defined metric

based on the task being solved. For abstractive ATS, examples of such rewards may include maintaining saliency, directed logical entailment, and non-redundancy [87]. More specifically, most RL-based text summarization researchers use rewards to encourage the system to produce more qualified summaries, while others develop more suitable metrics to train their models [23,137].

Generally, based on the investigation, it is observed that researchers utilized RL in abstractive summarization in three different areas: i) To solve various problems that deep sequence-to-sequence models cannot solve, ii) to incorporate extractive features with abstractive summarization to take advantage of both types of ATS, and iii) to create novel evaluation metrics based on references other than ground-truth summaries. All this work has one optimal goal which is to enhance the quality of abstractive summarization. The following sections describe these three areas in detail.

4.1. RL to solve deep sequence-to-sequence problems

A flurry of sequence-to-sequence-based research has been directed to generate an acceptable summary that the reader can rely on, providing the information that stands out most in the article. Unfortunately, even the most significant work in the field has led to several shortcomings: as the source articles get longer, deep sequence-to-sequence models become incapable of dealing with long-term dependencies, and then generate low-quality and incoherent summaries. Another problem is that their generated summaries contain few novel words and phrases other than those included in the source article, i.e., low novelty levels. Moreover, inconsistency problems of the exposure-bias [138] and different training/testing evaluation metrics (loss/evaluation mismatch), lead to incoherent and low-quality summaries. As discussed in section 8.1, the last two problems resulted from deep sequence-to-sequence models because they have only one objective, which is to maximize the likelihood of predicting summaries. Finally, deep sequence-to-sequence models are unable to generalize to datasets other than those they are already trained on, which leads to the lack of generalization problem.

RL systems have efficiently addressed most of these problems using policy-learning [96]. However, RL approaches must be combined with TL approaches to solve the lack of generalization problem.

Specifically, the quality of abstractive ATS can be measured by its shortness, the importance and relevance of its information to the source document, and its validity in terms of semantic and syntax [23]. To improve the performance and quality of sequence-to-sequence-based summaries, researchers worked to optimize objectives other than the maximum-likelihood using RL approaches [19,99,100,139,140]. Moreover, Paulus et al., [99] combine the RL loss from a self-critical policy gradient with cross-entropy in order to optimize the non-differentiable ROUGE as a reward while maintaining the resulting summary readability. The results show improvements in terms of ROUGE scores, readability, and quality. Furthermore, Li et al., [139], propose a joining training process for an actor-critic framework to improve the training process using the policy gradient method. The maximum likelihood evaluator is combined with a neural network-based binary classifier to make the results and the human-written summaries indistinguishable.

To solve the problems of exposure bias and loss/evaluation mismatch [138], Keneshloo et al., [140] suggested an RL framework with scheduled sampling to gradually remove the reliance of model training on the cross-entropy loss and increase the model's dependency on its own output. Furthermore, Keneshloo et al. [141] proposed to use either scheduled sampling [142] or Gibbs sampling [143]. Other RL-based attempts have been proposed to mitigate these gap problems [19,99,100,139,140]. In practice, however, RL-based approaches struggle to efficiently address these gap problems because RL-based training suffers from the noise gradient estimation problem [144] which makes its training suffer from instability and hyperparameter sensitive problems. [133]

The ROUGE metric's non-differentiability problem is solved by RL policy improvement [23] by optimizing its scores directly as rewards.

To address the problem of generating incoherent summaries when dealing with larger text, i.e., generating low-quality summaries when encoding at document-level, Paulus et al., [99] use an intra-attention mechanism to avoid repetition. Moreover, Celikyilmaz et al., [100], first divide long text paragraphs between multiple communicating encoders and then produce the summary by a unified decoder that is conditioned on all encoders' hidden states. As a result, this model, which is trained end-to-end using both maximum likelihood estimation and RL objectives, produces more efficient representations and significantly improves ROUGE results. Furthermore, Chen and Bansal [20] proposed a model consisting of an extractor to extract the most salient sentences, and an abstractor to rewrite these sentences from scratch using novel words, as described in Section 4.2. This model enhances the quality of the summaries with faster training and decoding. Also, it can be used to summarize longer text with better performance.

The problem of low levels of novelty has been considered after noticing the high copy rates in early work of deep sequence-to-sequence models. This problem is a very challenging task for sequence-to-sequence models [18]. Early sequence-to-sequence attention-based models [16,28,114] have produced abstractive summaries with very high copy rates even in short summaries with 92%, 74%, and 75% copy rates, respectively, using Gigword dataset. Next, for longer text and summaries, and after the pioneering work of pointer-generator architectures emerged [16,18] with their ability to extract some words from the input text, the copy-rate of full sentences surpasses 30% [106]. This high-level of extractiveness in the abstractive summaries encourages researchers to work hardly to solve this substantial problem, which negatively affects the quality of the results. Many attempts have been made to alleviate this high word overlapping behaviour, including [20,23,106,111,145] who have efficiently addressed this problem using RL approaches using various techniques. Chen and Bansal [20] define novelty score as the ratio of novel words in the summary that do not appear in the input text. As mentioned previously, their mixed model, extractive and abstractive agents, increases the levels of novelty in the resulting summaries. Cibils et al., [111] presented a model based on DBS and a diversity factor in calculating neural network loss to improve the diversity of generated summaries, which in turn improves their novelty. Also, they introduced a new metric to calculate the percentage of copied text in the produced summary. Another significant work in this field has been done by [23] who have developed a novelty metric to encourage the model to generate novel words, which is described in Section 4.3. Furthermore, Boutkan

et al., [106] achieved comparative novelty results by using the multi-head attention mechanism [4], pointer dropout [146], and two new loss functions, to avoid overfitting and encourage more novelty while obtaining comparative ROUGE scores. Moreover, Chakraborty et al., [145] strengthen the pointer-generator network of [18] by adding an OOV penalty to improve the novelty of the generated summaries. However, the novelty score is inversely proportional to ROUGE scores. That is, as the novelty score increases, the ROUGE scores decrease [145]. Therefore, finding a compromising measurement is still an open research question. Table 3 compares novelty scores for the previously discussed models applied on the non-anonymized version of the CNN/DM dataset.

However, as presented in Table 3, there is still a huge gap between the novelty of human summaries and automated summaries, leaving plenty of room for exploration in the research space.

In the same context, Song et al., [147] has recently done a significant work in this field by controlling the level of novelty of the resulting summaries by framing the task as language modeling using different mechanisms to generate summary hypotheses. Their experiments were performed only on Gigaword and Newsroom datasets, yielding SotA ROUGE results on the Gigaword corpus, as shown in Section 7.2.1.

Finally, RL can be combined with TL approaches to solve the problem of lack of generalization. This problem occurs when a model that is trained on a large enough corpus fails to generate summaries of other smaller data. Keneshloo et al., [140] presented a solution to this problem using RL by fine-tuning a pre-trained model on one dataset and testing it on different datasets. Furthermore, Chen et al., [20] generalized their work to the DUC-2002 dataset using RL and performed well. Table 4 illustrates the problems of sequence-to-sequence models and their solutions proposed by RL-based research.

Generally, although RL approaches enhanced the results and solved some problems of DL-based models, they are not sufficient to solve other challenges and obstacles that face abstractive ATS research. For example, efficiently handling long documents remains an obstacle, resulting in low-quality and low-novelty summaries. Compared to PTLMs, which are discussed in Section 5, deep sequence-to-sequence models are trained on relatively much smaller datasets, resulting in poor semantic and contextual features of word embeddings. Moreover, deep sequence-to-sequence models are unable to take advantage of parallelization due to their sequential nature obstacle, resulting in low training speeds. Solutions to the above problems are offered by TL approaches, Section 5.

4.2. RL to combine extractive and abstractive summaries

In the previous work of deep sequence-to-sequence models, as discussed in section 3.1.3, [14–18] were the first to incorporate extractive objectives with abstractive summarization. This was done by copying some words from the source document into the generated summary to solve OOV and inaccurate factual problems. Afterward, researchers use RL approaches to deal with these problems more efficiently and generate better summaries with respect to quality and performance by fusing extractive and abstractive models [19–23,64]. Inspired by human behaviour in summarizing documents, these models first extract the most salient sentences from the entry document, then abstract them, using two networks: extractor and abstractor networks. For each extracted sentence, it may be unmodified, paraphrased, or excluded in the last generated summary [148].

Liu et al., [19] suggest an adversarial framework that trains abstractive and extractive models at the same time using RL policy gradient to optimize the abstractive model for a highly-rewarded summary, resulting in a more coherent summary. Chen and Bansal [20] propose a fast summarization model that first extracts salient sentences and then rewrites them. Their model uses RL-based sentence-level policy gradient to bridge the non-differentiable computation between extractive and abstractive networks hierarchically while maintaining language fluency. Hsu et al., [21] introduce an inconsistency loss function to penalize the inconsistency between two levels of attention, sentence-level and word-level, to train their combined extractive-abstractive model. Gehrmann et al., [22] propose a two-step abstractive summarization model biased by an efficient and important content selector that works as bottom-up attention to determine which words in the source text should be included in the summary. Krys'cin'ski et al., [23] use a contextual network for extracting and compacting the source document and a language model for generating a concise paraphrase. However, like the pointer-generator mechanism, the problem of low levels of novelty still exists in these models. Moreover, there is a mismatch problem in these models between the training objective and the evaluation metric. The former uses the sentence-level ROUGE score as an RL reward for the sentence selection action, which doesn't guarantee the optimal summary-level ROUGE scores that is used by the latter. For this, Bae et al., [149] proposed a novel training method using the summary-level ROUGE scores as an RL objective instead of sentence-level to globally improve sentence selection and directly increase ROUGE scores.

In the same context, by integrating PTLMs with RL approaches, Bae et al., [149] and Wang et al., [64] have proposed models to enhance abstractive ATS outcomes and solve RL-based problems taking advantage of PTLMs' universal representations and parallelization. Both models are described in Section 5.2.4.

Table 3
Models' n-gram novelty.

Research	Model Name	1-gram	2-gram	3-gram	4-gram
See et al. [18]	Pointer-gen + coverage	0.07	2.24	6.03	9.72
Pasunuru and Bansal [87]	Entail (RL)	-	2.63	6.56	10.26
Liu et al. [19]	SumGAN	0.22	3.15	7.68	11.84
Boutkan et al. [106]	WPLoss-1atthead	1.44	8.86	17.96	25.50
Chen and Bansal [20]	rnn-ext+abs+RL+rerank	0.3	10.0	21.7	31.6
Krys'cin'ski et al. [23]	ML+RL ROUGE+Novel, with LM	3.25	17.21	30.46	39.47
Reference Summary [23]		13.55	49.97	70.32	80.02

Table 4
Examples of deep sequence-to-sequence problems that are solved by RL approaches

Problem	RL Solution proposed by
Low quality summaries for long documents	[20,99,100]
Low Novelty	[20,23,106,111,145]
Exposure-bias	[19,99,100,139,140]
Loss/Evaluation mismatch	[99,100,140]
ROUGE metric's non-differentiability	[23,99]
Lack of Generalization	[20,140]

4.3. RL to create novel metrics and rewards

Most RL-based sequence-to-sequence abstractive ATS models use ROUGE [5] for rewarding during training. However, ROUGE metric has three main limitations: its bias towards lexical similarity, its low consideration for fluency and readability of the generated abstractive summaries [150], and its hard prerequisite of using ground-truth summaries to produce the scores. Moreover, high ROUGE scores summaries usually obtain low human appealing [137]. Therefore, researchers developed new metrics to enhance novelty [23], factual consistency [151], and quality based on question-answering [108,150] and human ratings [137] using RL rewarding approaches without requiring ground-truth summaries.

Novelty (a.k.a. abstractiveness) metrics identify the percentage of novelty in the resulting summary. Several researchers have designed their models to generate hybrid summaries containing both abstractive and extractive features. The goals are either to avoid infactual details/UNK words by copying named-entity and rare words [16,18], or to enhance the results by taking advantage of both extractive simplicity and informative and abstractive readability using RL [20]. As a result, these models suffered from high copy rates, and their generated summaries elicit a higher sense of being more extractive than abstractive. Therefore, measuring novelty and pushing the results towards a higher level of abstraction will help improve the quality of results even further.

Recently, the authors of [23] proposed a novelty metric to encourage the generation of novel words as a reward. They defined their metric as the number of unique novel n-grams generated within the resulting summary, normalized by the length ratio of the generated and ground-truth summaries to avoid a bias towards generating short summaries, as follows:

$$Novelty = \frac{\text{unique}(\text{SGEN},\ \text{n})\ --\ \text{unique}(\text{XINPUT},\text{n})}{\text{unique}(\text{SGEN},\ \text{n})} * \frac{||SGEN||}{||SREF||} \tag{10}$$

where *SGEN* is the generated summary, *XINPUT* is the input text, *SREF* is the reference summary, and *n* is n-grams.

Factual consistency metric [151] has been developed to measure whether the generated summary contains only the information contained in the input document. In addition, the metric identifies the conflict between the input and output texts.

To enhance the quality of summaries, researchers work with two strategies, question-answering and human rating. Question-answering-based quality metrics were developed by [108] and [150]. APES [108] measures the summary quality by its ability to answer a set of questions about the essential entities in the article. Scialom et al., [150] proposed QA-based unsupervised metrics for reinforced summarization models that don't require reference summaries without using ROUGE as a reward.

On the other hand, a human-rating-based quality metric is developed by [137] to get rid of relying on reference summaries required by ROUGE metric, which had not been approved by experts evaluation. This metric has learned a new reward function based on 2,500 human ratings. Examples of other potential rewards include maintaining saliency, directed logical entailment, and non-redundancy [87].

5. Abstractive ATS using TL approaches

The idea of TL is inspired by the ability of humans to transfer their knowledge gained from one task to solve other related tasks that they have not faced before. In the machine learning field, the knowledge learned from one domain/task can be transferred to another related domain/task [152].

NLP research fields, including abstractive ATS, have flourished following the developments of Transformers [4] and various PTLMs [6,7,10–13], described in the following sections. Specifically, PTLMs are trained on abundant unlabelled data benefitting from the parallelization ability of the Transformer as their fundamental architecture to speed up the training process. This results in learning universal language representations that can be utilized by, i.e., transferred to, various downstream tasks, including abstractive ATS, to increase the quality of their results and the speed of training.

5.1. Transformer

The Transformer architecture [4] is the backbone of the latest PTLMs, including BERT [6], BART [10], PEGASUS [7] and many others. This architecture relies entirely on attention mechanisms rather than recurrences to draw global dependencies between inputs and outputs. Therefore, it overcomes the problems of RNN, LSTM, and GRU of long-term dependencies and sequential nature, using a self-attention mechanism to model similarities between words regardless of their positions in parallel. Specifically, The Transformer consists of six encoders in a hierarchical structure and the same number and structure of decoders. Each encoder includes a multi-head

self-attention layer and a fully connected feed-forward neural network layer. On the other hand, decoders have an additional layer between these two layers: the encoder-decoder attention layer, to focus on essential words in the input sequence. To calculate attention, three matrices are used, the query, key, and value matrices, each of which consists of n vectors for a document containing n tokens, which leads to quadratic memory complexity.

This architecture outperforms sequence-to-sequence models which were designed either by RNNs or CNNs architectures and showed SotA results in various NLP tasks with less training time. Fig. 5 depicts the internal structure of the Transformer as described in [4].

However, the main limitations of the self-attention mechanism used by the standard Transformer can be summed up as the quadratic memory and computational complexities, the large number of operations required when dealing with long sequences, and the fixed-length context prerequisite to learn long-term dependencies.

To improve Transformer efficiency and solve its problems, various attempts have been suggested such as locality-sensitive hashing based attention, Reformer [153], which reduces memory complexity and speeds up training with a O(n log n) complexity. Also, performing sparsity, Sparse Transformer [154], can train deeper networks with fewer operations with a O(n $\sqrt{n}$) complexity, saving O ($\sqrt{n}$) over the self-attention mechanism. Moreover, using autoregressive-based architectures, Transformer-XL [155], enables models to understand the context and learn dependency beyond a fixed length limitation.

In particular, for abstractive ATS, **BigBird** [29] has been essentially proposed to improve handling long documents using the sparse attention mechanism which reduces dependency to linear, i.e., O(n). BigBird is applied on the abstractive ATS task to summarize long documents and achieved SotA results as discussed in Section 5.2.4 and shown in Section 7.2.2. In addition, HEPOS (Head-wise Positional Strides) encoder-decoder attention [156] was recently proposed to effectively identify the salient content and capture the global context of a long input text with lower complexities and costs. Compared with self-attention encoders, HEPOS-based models can handle double the words and generate more informative and faithful summaries with higher ROUGE scores, as shown in Section 7.2.2.

However, more details of the various efficient Transformers and self-attentions are presented in [157] and [156], as it is beyond the scope of this article.

5.2. *Pre-trained language models PTLMs*

The breakthrough of pre-trained language models revolutionized the field of NLP research. With their powerful capabilities and resources, large organizations and educational institutes design and train mega models with a tremendous number of parameters on large-scale general-purpose datasets (160GB or more). Training PTLMs results in learning universal language representations that can then be reused by downstream tasks, even with small amounts of datasets. This provides downstream tasks with a strong initialization, avoiding training them from scratch, resulting in significant improvements in training speed and quality of their results.

Basically, PTLMs are differentiated based on their architectures and pre-training tasks. Knowing these details is crucial to decide whether a particular PTLM is suitable for a specific task. For example, BERT is better suited to understanding tasks than generating tasks. Hence, PTLMs must be chosen wisely based on these details when adapting them to downstream tasks.

Most pre-training tasks are designed for generation tasks, understanding tasks, or specific tasks. The most commonly used pre-training tasks include language modelling (such as ELMO, GPT, UniLM), masked language modelling, i.e., cloze task [158] (such as BERT, MASS), denoising autoencoder (such as BART) , permuted language modelling (such as XLNet), and others designed for specific-tasks such as gap sentence generation [7] and future n-gram prediction for abstractive ATS [159].

For PTLM architectures, ELMo uses LSTM while the rest of PTLMs use Transformers as their backbone architecture. Some of them pre-train the encoder only, such as BERT, others pre-train the decoder only, such as GPT, while others pre-train both the encoder and the decoder together using the encoder and decoder parts of Transformer, such as Ernie-Gen, BART, and T5. In general, for TG tasks, the framework usually consists of a bidirectional encoder and a unidirectional decoder [160].

There are two main strategies for downstream tasks to apply pre-trained language representations: feature-based and fine-tuning [6]. In the feature-based strategy, the fixed features are extracted from the pre-trained model, the task-specific architecture is required, and the pre-trained representations are used as additional features. On the other hand, in the fine-tuning approach, a simple classification layer is often added to the pre-trained model. The downstream task is trained by jointly fine-tuning all pre-trained parameters and introducing minimal task-specific parameters. Recently, fine-tuning has become the traditional adaptation strategy of most PTLMs. However, fine-tuned parameters are not the same for different downstream tasks. In modern PTLMs, the output vectors of their neural encoders represent the word semantics depending on its context. These vectors are called contextual word embeddings [161].

For abstractive ATS, several PTLMs are fine-tuned to enhance the quality of the generated summaries, such as BART [10], T5 [8], MASS [162], UniLM [163], and Switch [9] as described in Section 5.2.2. Moreover, as shown in Section 5.2.3, other PTLMs are pre-trained in tasks specifically designed for the abstractive ATS task, such as the gap sentence generation task used in PEGASUS [7], and the future n-gram prediction task used in ProphetNet [159]. Furthermore, researchers boost the performance of the abstractive ATS task by combining multiple PTLMs for encoders and decoders using various methods. Other researchers provide general and extensible frameworks for abstractive ATS such as BERTSUMABS, BERTSUMEXTABS [164], GSum [122], Refactor [132], and SimCLS [133]. All this work is described in Section 5.2.3 and Section 5.2.4

Currently, one popular research direction is to modify and/or adapt Transformers and PTLMs to different understanding and generation tasks, as shown in Section 5.2.4. Hence, first, the details of some chosen PTLMs are discussed, and then the different ways of adapting these PTLMs to different tasks are described, with an emphasis on the abstractive ATS task.

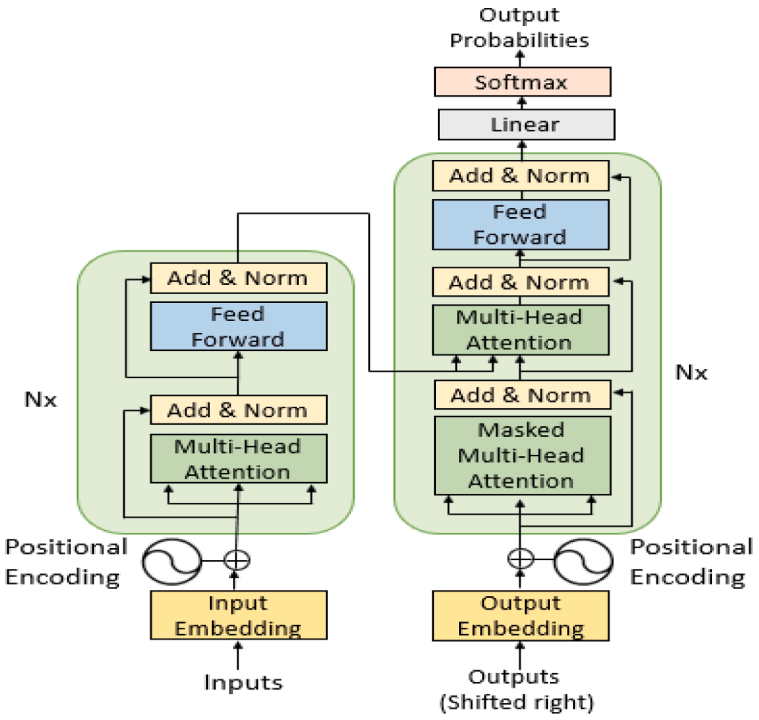

Fig. 5. The Transformer architecture.

5.2.1. Popular PTLMs for comprehension and generation tasks

ELMo [165] is a deep contextualized word representation that models complex word usage characteristics and how these uses differ across linguistic contexts. Specifically, this PTLM consists of a two-LSTM-layers encoder and is pre-trained with a Bidirectional language model (BiLM) task using forward and backward language models. ELMo generates contextual representations to downstream tasks by shallowly concatenating the extracted context-sensitive features of each token from both sides, i.e., left-to-right and right-to-left, from two independently trained unidirectional LSTMs shown in Fig. 6. However, ELMo cannot be pre-trained to learn interactions between these features. Moreover, ELMo can be applied to downstream tasks using only the feature-based strategy, where its actual parameters are not adjusted, i.e., fixed.

The Bidirectional Encoder Representations from Transformers (**BERT** [6]), shown in Fig. 7, is a Transformer-based, large-scale, self-supervised, encoder-only, bidirectional multi-layer PTLM. BERT is trained for language understanding tasks using the masked language modelling (the cloze task) and next-sentence prediction objectives. As a result, an efficient embedding that utilizes the semantic text features is generated by combining representations of words and sentences into a single very large Transformer [164]. Also, BERT can efficiently identify relevant sentences by pre-training the model in a binary classification task to predict which sentences will follow one another.

Compared to ELMo, BERT is Transformer-based while ELMo is LSTM-base architecture. Unlike ELMo, BERT can pre-train the interactions between left and right context tokens features. Also, BERT is effective for both fine-tuning and feature-based strategies. However, BERT has limitations in masked position dependencies disregarding and pre-train finetune discrepancy. these problems are solved by other recent PTLMs such as XLNet [166] and Ernie-Gen [160] models. Moreover, because predictions in BERT are not made auto-regressively, BERT is better suited to Natural Language Understanding (NLU) tasks than TG tasks.

Open AI released three generations of their model, GPT, GPT-2, and GPT-3. **GPT** (Generative Pre-Training) [11] is a task-agnostic, decoder-only, left-to-right Transformer-based model that predicts sequence words one by one and learns general language representations, which can then be transferred to multiple downstream tasks with a little adaptation. The GPT architecture is shown in Fig. 8. The GPT model is pre-trained on NLP generative tasks on various unlabelled datasets, to improve language understanding.

Compared to BERT, both are Transformer-based architectures, but GPT uses only the decoder part of the Transformer, while BERT uses the encoder part. Moreover, GPT uses a left-to-right Transformer that can only model the context from the left side, whereas BERT uses a bidirectional Transformer to model the context based on both sides.

Next, GPT-2 [12] extends GPT with some modifications to learn various NLU and TG tasks in zero-shot transfer without any finetuning to a specific task dataset. However, repetition, logic conflicts, and lack of long-range coherence are the main shortcomings of GPT-2 in generating tasks [167].

The autoregressive-based GPT-3 [13] is then proposed, including 175B parameters. This model is evaluated with various settings including zero-shot, one-shot, and few-shot learnings without weight updating, and with and without language prompting, achieving competitive results with other fine-tuning approaches. However, GPT-3 still suffers from several weaknesses. For example, it still

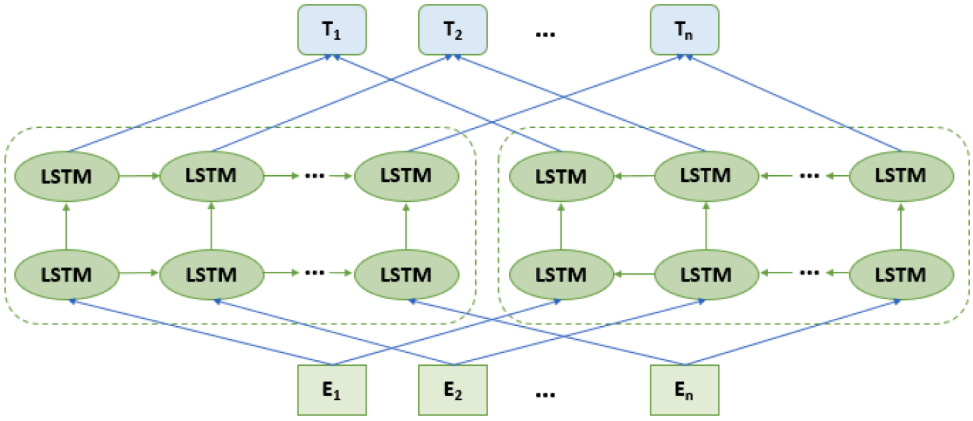

Fig. 6. ELMo [6].

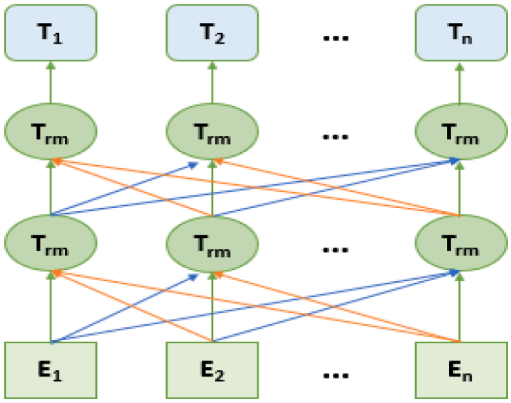

Fig. 7. BERT [6].

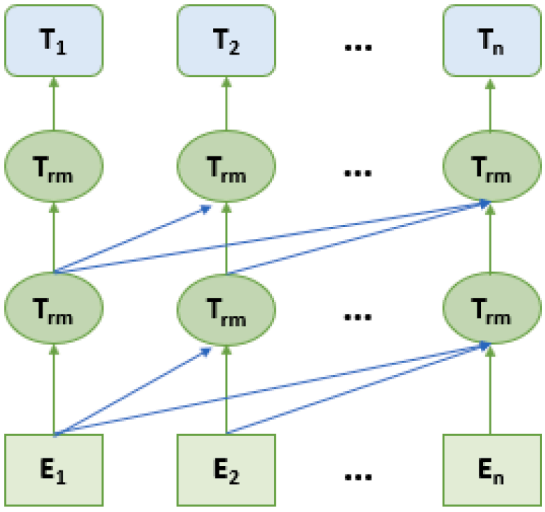

Fig. 8. GPT [6].

struggles with several inference and reading comprehension tasks. Also, it has structural and algorithmic limitations. Poor pre-training sample efficiency is another limitation. Other shortcomings include the possibility of generating text with trivial errors, high cost inferencing, and the uncertainty about whether the model learns new tasks based on pre-trained learning or from scratch.

5.2.2. PTLMs fine-tuned to Abstractive ATS

The Unified Language Model, **UniLM** [163], is designed with a shared multi-layer Transformer network and parameters to be jointly pre-trained on a big dataset using three different types of unsupervised language modelling objectives: unidirectional, bidirectional, and sequence-to-sequence. Its task is to predict a masked word based on its context, i.e., cloze task. This multi-objective pre-training allows UniLM to be fine-tuned not only for NLU tasks, like BERT, but also for TG tasks such as MT and ATS.

MAsked Sequence to Sequence, **MASS** [162], is a Transformer-based sequence-to-sequence pre-trained model designed for encoder-decoder based language generation tasks such as MT and ATS. Inspired by BERT [6], using randomly masked fragments passed into the encoder, the decoder predicts the missing part. MASS is fine-tuned to three tasks: MT, ATS, and conversational response generation. It improved the results significantly especially in the MT task.

Experimentally, both UniLM and MASS models performed well in abstractive ATS achieving reasonable ROUGE scores.

BART [10], is a denoising auto-encoder PTLM which is particularly effective when fine-tuned to TG tasks and works well for comprehension tasks. Specifically, BART is pretrained to convert a corrupted text to its original script by optimizing the cross-entropy reconstruction loss between the decoder output and the original text. This is done by corrupting a text with an arbitrary noising function, masking out a random subset of words, and then learning how to reconstruct the original text. Bart, which is described in Fig. 9, is a generalized version of BERT (i.e., bidirectional encoder and masked language modelling), GPT [11] (i.e., left-to-right decoder), XLNet [166] (i.e., Permute language model), UniLM [163] (i.e., multitask masked language model), and MASS [162] (i.e., masked sequence to sequence). As a result, BART performs well on multiple discriminative tasks like GLUE [168] and SQuAD [169], and generation tasks like abstractive dialogue, question-answering, translation, and ATS. As for the latter, it gains comparative scores on the XSum and CNN/DM datasets, as shown in Section 7.2.1.

Compared to UniLM, both can be used for both discriminative and generative tasks, but UniLM predictions are conditionally independent, while in BART they are autoregressive [10]. Compared to MASS, both are very similar, but BART is more effective on discriminative tasks.

The second version of UniLM, **UniLMv2** [170], is proposed for two complementary modeling tasks: autoencoding and partially autoregressive, to jointly learn bidirectional and sequence-to-sequence language modeling, respectively. Moreover, a pseudo-masked language model training procedure is introduced, making the computations more efficient. Based on conventional masks, inter-relations between corrupted tokens are learned via autoencoding, which provides global masking information to partially autoregressive modeling. This makes position embeddings accessible to partially autoregressive modeling, making it able to learn intra-relations between masked spans. For abstractive ATS, as shown in Section 7.2.1, UniLMv2 boosted the performance of UniLM on both CNN/DM and XSum corpora.

The Text-To-Text Transfer Transformer model (**T5**) [8] is a unified framework that treats with every language problem as a text-to-text problem with the flexibility in applying the same model settings, including its objective, training, and decoding processes directly to various NLP tasks.

Compared to BERT, T5 utilizes the standard encoder-decoder architecture for both classification and generative tasks with the same BERT's objective of masked language modelling to achieve better results. Instead, BERT is developed to produce a single prediction for each input token or sequence, which is more suitable for classification tasks or span prediction tasks than generative tasks. As a result, T5 can be successfully applied to a number of NLP tasks including generative, classification, and regression tasks. For abstractive ATS, T5 achieves comparative results on the CNN/DM dataset, as shown in Section 7.2.1. Fig. 10 depicts the T5 model showing some of its capabilities.

Ernie-Gen [160] is an enhanced multi-flow sequence-to-sequence pre-training and fine-tuning framework proposed for TG tasks focusing on addressing the exposure-bias problem more efficiently when fine-tuning PTLMs to downstream tasks. To this end, both the infilling generation mechanism and the noise-aware generation method have been incorporated to bridge the discrepancy between training and inference. Moreover, instead of word-to-word prediction, Ernie-Gen is trained using a span-by-span generation task to produce human-like summaries. As a result, Ernie-Gen achieve comparable results on abstractive ATS, as shown in section 7.2.1.

Recently, Fedus et al., [9] from Google Brain, introduced two large Switch Transformer models, **Switch-XXL**, and **Switch-C** with 395B and 1.6T parameters, respectively. Different from the standard Transformer encoder, the dense feed forward network layer is replaced by a sparse switch feed forward network layer. Although the proposed models are much larger than the previous ones, they successfully solved Transformers and other PTLMs problems by reducing computation and communication costs, increasing

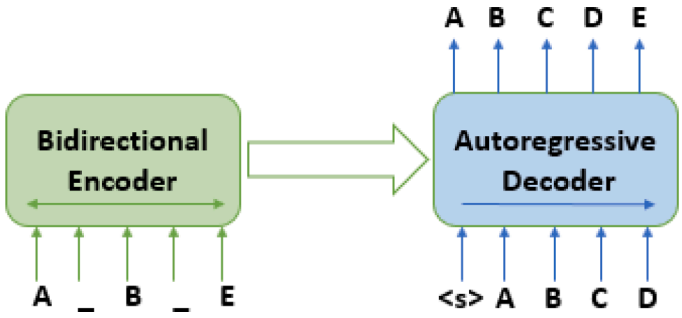

Fig. 9. BART [10].

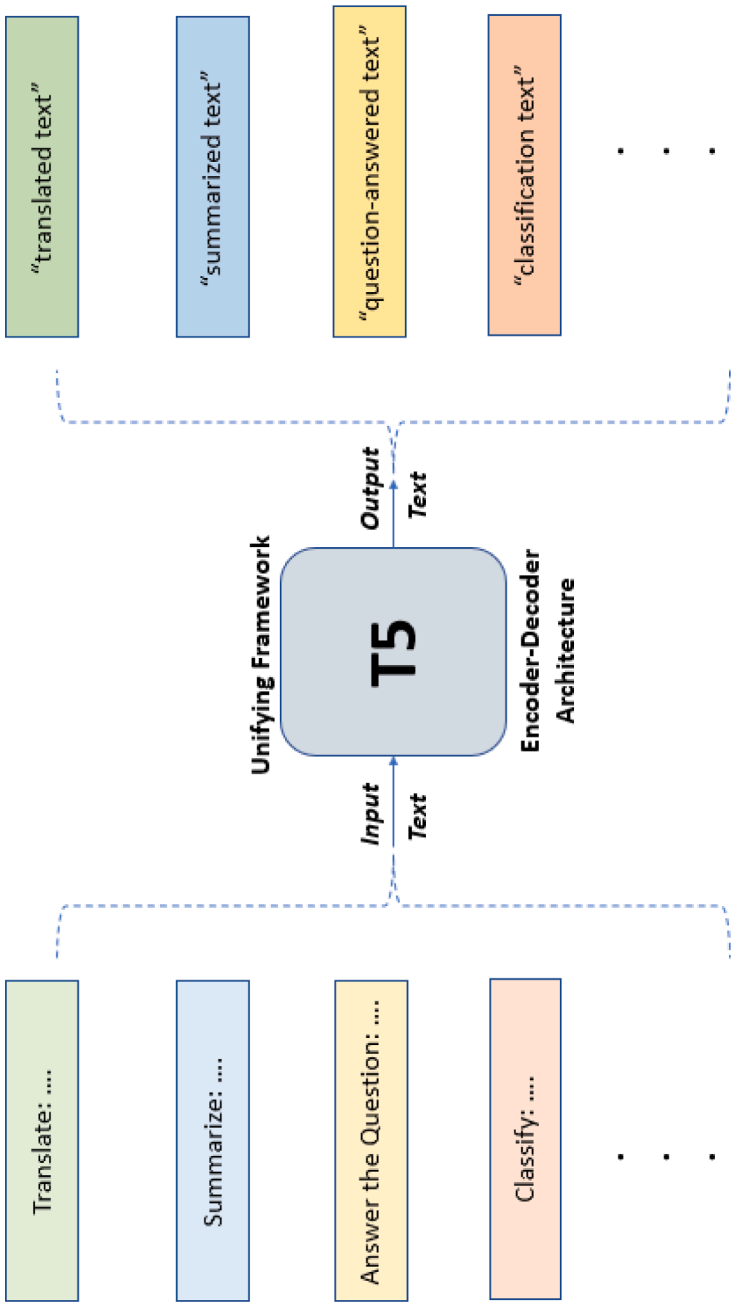

Fig. 10. T5 Model.

pre-training speed, and mitigating training instability. The researchers fine-tuned their proposed model to the abstractive ATS task, resulting in a ROUGE-2 score of 19.6 on the CNN/DM dataset, and 22.3 on the XSum corpus using Switch-Base and Switch-Large model, respectively.

5.2.3. Abstractive ATS PTLMs

Pegasus [7] is a Transformer-based encoder-decoder sequence-to-sequence model that is pre-trained on huge unsupervised text corpora with a new self-supervised objective, gap sentences generation, which is essentially designed for the abstractive ATS task. This model's main idea is to remove/mask important sentences from the input document to be generated as one output sequence from the remaining sentences. The advantage of Pegasus is that it masks multiple completed sentences instead of smaller continuous text spans. In addition, it chooses the sentences based on their importance, not randomly like other models. Moreover, Pegasus can work at the subword level instead of word level as an effective alternative strategy to the copy mechanism. Specifically, Pegasus applies two learning objectives at the same time, the masked language model and the gap sentence generation. For example, as shown in Fig. 11, out of the three sentences, one of them is masked with [MASK1] and used as the target generation text (i.e., gap sentences generation). Whereas the other two sentences are kept in the input with some tokens that are randomly masked using [MASK2] (i.e., masked language model). As a result, Pegasus was evaluated on 12 different abstractive ATS corpora from different domains. It achieves comparative results on both short-document and long-document datasets, as shown in Section 7.2.1 and Section 7.2.2, respectively.

ProphetNet [159], which is based on Transformer encoder-decoder architecture, introduces a novel self-supervised learning objective by predicting multiple future tokens based on previous context tokens. The future n-gram prediction objective utilizes useful information to guide the model to predict multiple future tokens simultaneously while preventing overfitting on local correlations. In addition, ProphetNet modifies the internal structure of Transformer [4] using the n-stream self-attention mechanism and the mask based autoencoder denoising task for sequence-to-sequence pre-training. ProphetNet was evaluated on CNN/DM and Gigaword datasets resulting in significant enhancements in ROUGE results compared to other premium PTLMs including T5, BART, and PEGASUS, as shown in section 7.2.1.

BERTSUM [164] adjusts BERT for both extractive and abstractive types of ATS. BERTSUM uses the pre-trained BERT-based encoder at document-level to efficiently represent sentences semantically. Also, a randomly initialized Transformer decoder is utilized to solve the problem of large text input. Furthermore, as discussed in Section 4.2 of combining extractive and abstractive objectives, a dual-objective encoder is used to optimize document-level summarization performance with minimum requirements without using the copy mechanism or RL approaches. Specifically, for each input token, BERT manipulates three kinds of embeddings: token, segment, and position embeddings. These embeddings are summed into a single input vector which is fed to the bidirectional Transformer layers to generate contextual vectors representing that token. As mentioned earlier, BERT is not suitable for fine-tuning as an encoder for TG tasks including the ATS task. To this end, BERTSUM extends BERT by modifying the three essential types of embeddings used in BERT to be suitable for the summarization task, as described in Table 5.

As a result, two general frameworks for abstractive ATS, BERTSUMABS, and BERTSUMEXTABS, have been proposed. The former uses the pretrained BERTSUM encoder, and an untrained 6-layered Transformer decoder initialized randomly, using two separated optimizers to overcome the problem of fine-tuning instability. The latter is fine-tuned twice on the extractive ATS task first, then on the abstractive ATS task.

STEP [171] is pre-trained to reinstate the original text from an artificially constructed input using three pre-training objectives related to abstractive ATS: sentence reordering, next sentence generation, and masked document generation. The distinct contribution of this work is that it achieves comparable results through training on much less data (19GB) compared to other related PTLMs such as Pegasus and ProphetNet.

For **extractive ATS**, MatchSum [172] is proposed based on BERTSUM [164] as a matching-based summarization framework that matches the input document and the proposed summaries in the semantic space. In addition, HIerachical BERT (HIBERT) [173] and DISCOBERT [174] are proposed for document encoding. These models are pre-trained to predict sentences (HIBERT) and sub-sentential discourse units (DISCOBERT) instead of words, at document-level to solve BERT's problems of redundancy, lack of

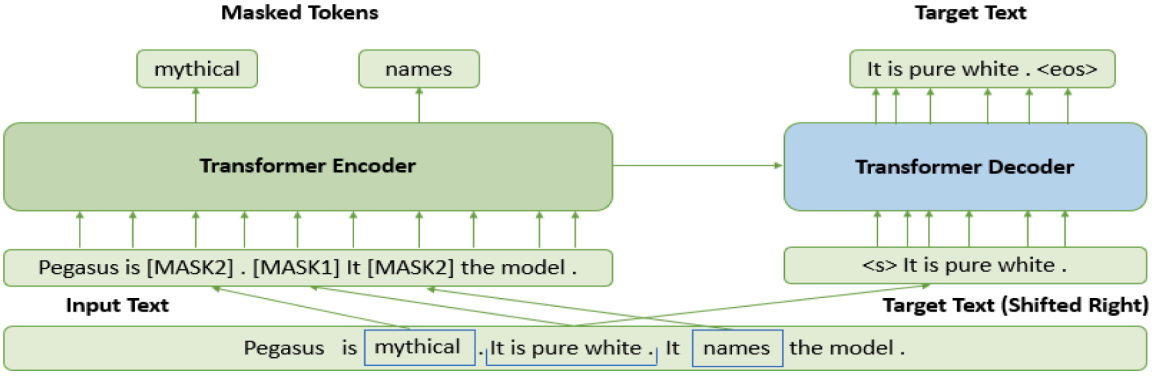

Fig. 11. Pegasus [7].

Table 5
Original BERT vs BERTSUM.

Embedding	Function	Summarization requirements	BERT	BERTSUM
Token embeddings	To identify the token meaning	Requires sentence-level representations	Word-level embedding	Sentence-level embedding by adding [CLS] tokens at the beginning of each sentence.
Segment embeddings	To distinguish sentences	Requires multi-sentence encoding and manipulating	Represents only sentence-pair inputs	Uses interval segment embeddings to discriminate multiple sentences.
Position embeddings	To indicate the token position	Requires flexibility	Maximum length of 512	More than 512 with random initialization.

information, and long-range dependencies. Furthermore, Zhong et al., [175] show how extractive ATS models can make use of different architectures, transferable knowledge, and learning schemas to enhance their results. To the best of our knowledge, for extractive ATS models, MatchSum [172] achieves the SotA Rouge-1/Rouge-2/Rouge-L scores with 44.41/20.86/40.55, respectively.

Table 6 compares chosen PTLMs relevant to the abstractive ATS task.

5.2.4. Utilizing PTLMs to abstractive ATS

Modern models compete to leverage the advantages of Transformers and PTLMs to several understanding and generative tasks, including the abstractive ATS task. These PTLM-based models exploit the rich semantic and contextual features of global language representations to improve the quality and accuracy of the resulting summaries in informative, readability, and faithfulness terms. Recently, adapting the Transformer and PTLMs and/or altering their internal structures for specific tasks, including ATS, has become a research trend due to the observed improvements in results. This section discusses some of recent chosen work in this context. For example, for language understanding tasks, Lei et al., [82] combine a simple recurrent unit with the Transformer, while Huang et al., [83] combine the Transformer with Bidirectional LSTM. Both Lei et al., [82] and Huang et al., [83] outperformed the basic Transformer and BERT models in terms of speed and accuracy, respectively, on various NLP comprehension tasks. Furthermore, Wang et al., [84] modify the Transformer architecture by adding an LSTM before or after all Transformer blocks to capture sequential concepts more efficiently for various NLP tasks. Also, they add a masked self-attention instead of self-attention heads to avoid leftward information flow. Moreover, they add a linear output layer during fine-tuning GPT [11], GPT-2 [12] and BERT [6] models. As a result, their modified architecture achieves better results and captures stronger word-level context for language models.

Table 6
A comparison of abstractive ATS-related PTLMs.

PTLM	Year	Transformer part used	Pre-training Task	Parameters	Pre-training corpus (size)
ELMo [165]	2018	"LSTM Encoder"	Bidirectional Language modeling	94M	One-Billion-Word [176] (30M sentences)
GPT [11]	2018	Decoder	Language modeling	110M/**117M**	BookCorpus (16GB) [177]
BERT [6]	2019	Encoder	Masked language modeling + Next sentence prediction	110M/**340M**	English Wikipedia (16GB) + BooksCorpus [177] (3.3G words)
T5 [8]	2019	Encoder-Decoder	Masked sequence-to-sequence language modeling	60M/220M/ ``770M/2.8B/ **11B**	C4 (750GB) [8]
MASS [162]	2019	Encoder-Decoder	Masked sequence-to-sequence language modeling	-	WMT monolingual corpus* (190M/62M/ 270M sentences)
UniLM [163]	2019	Encoder	Multi-task sequence-to-sequence masked language modeling + Next sentence prediction	340M	English Wikipedia + BookCorpus (16GB) [177]
BART [10]	2019	Encoder-Decoder	Denoising auto-encoder	406M	BookCorpus [177] + English Wikipedia + CC-News [178]+ OpenWebText** + STORIES [179] (160GB)
PEGASUS [7]	2020	Encoder-Decoder	Gap sentence generation + Masked language modeling	568M	C4 (750GB) [8] / HugeNews (1.5B articles) (3.8TB)
ProphetNet [159]	2020	Encoder-Decoder	Future n-gram prediction	-	BookCorpus [177] + English Wikipedia + CC-News [178]+ OpenWebText** + STORIES [179](160GB)
STEP [171]	2020	Encoder-Decoder	sentence reordering + next sentence generation + masked document generation	585M	GIGA-CM (6.5M documents) (19GB)
Ernie-Gen [160]	2020	Encoder-Decoder	Span-by-span generation	110M/ **340M**	English Wikipedia + BookCorpus (16GB&430GB) [177]
UNILMv2 [170]	2020	Encoder-Decoder	Sequence-to-sequence masked language modeling + Bidirectional language modeling + Pseudo-masked language model	110M	BookCorpus [177] + English Wikipedia + CC-News [178]+ OpenWebText ** + STORIES [179] (160GB)
Switch-C [9]	2021	"Switch Transformer Encoder"	Masked language modeling	7.4B/26.3B/ 395B/**1.6T**	Improved C4 (180B tokens)

*http://www.statmt.org/wmt16/translation-task.html
**https://skylion007.github.io/OpenWebTextCorpus/

Moreover, for multi-document ATS, Kieuvongngam et al., [43] fine-tune the GPT-2 model via text-to-text, multi-loss training strategy to summarize COVID-19 research articles. Also, they fin-tune BERT for their extractive ATS model.

For long-document abstractive ATS, the BigBird-PEGASUS model [29] uses a sparse attention encoder, which is described in Section 5.1. This model beats the Transformer-based Pegasus in summarizing long documents, as shown in Section 7.2.2. Moreover, Gidiotis and Tsoumakas [75] use the divide-and-conquer approach by dividing the long input document and its summary into smaller parts and then allowing the model to learn to summarize each part of the document in isolation. Then these partial summaries are merged to form the final long summary. The researchers of [75] have demonstrated that this approach enhances summarization performance and reduces computational complexity when used with different summarization models based on LSTM, RUM, and Transformer.

This field of research has recently been enriched with more attempts. Xiao and Carenini [180] study repetition reduction in long documents achieving comparative results. Also, Pilault et al., [65] utilize the mixed extractive-abstractive approach using a single PTLM, GPT-2 [12], to generate the final abstractive summary. ROUGE results of these models are presented in Section 7.2.2. More recently, Huang et al. [156], discussed in Section 5.1, add HEPOS to the Locality-Sensitive Hashing (LSH) attention encoder. Their model, LSH+HEPOS, achieves SotA results on arXiv and PubMed datasets outperforming the sparse-attention encoder BigBird-PEGASUS, the Transformer-based PEGASUS, and the divide-and-conquer based Dancer-PEGASUS models, as shown in Section 7.2.2.

For short-document abstractive ATS, many ideas have been proposed to promote the performance by adapting and combining Transformers and PTLMs in different ways.

By integrating extractive and abstractive summaries to enhance the quality of the results, Bae et al., [149], Wang et al. [64], and Liu and Lapata [164] suggested their models. Bae et al., [149] leverage BERT representations in the extractive network only, using [20]'s RL-based abstractor. Wang et al. [64] exploit BERT word embeddings in two sub-models, extractor and abstractor, train them together and then bridge their results using RL approaches. The extractive sub-model extracts the most salient statements from a document, then the abstractive sub-model rewrites these key sentences in a readable and concise form. The proposed model yielded comparative results on the CNN/DM dataset, as shown in Section 7.2.1, which demonstrates the relevance of their idea. Fig. 12 demonstrates their proposed model.

Additionally, other ideas and techniques have been proposed to leverage Transformers and PTLMs, such as dual-encoding [85], dual-decoding [181], decoder-only network [182] , various training strategies [183], and feature-based adaptation [184].

Recently, by utilizing the knowledge-enhanced mechanism described in Section 3.1.5, [121,122,127,128] use an additional encoder to incorporate extra useful external information into their models to optimize the accuracy, faithfulness, fluency, and quality of the abstractive ATS outcomes. As a result, these models have shown significant improvements in performance and ROUGE scores, as shown in Section 7.2.1. SemSUM [128] incorporates semantic dependency graphs to guide the generation process to produce a summary that is more semantically relevant to the contents of the input text. Zhu et al., [127] incorporate factual relations via knowledge graphs to enhance the factual consistency of abstractive ATS. Furthermore, Saito et al., [121] incorporate saliency by investigating nine combinations of PTLMs and saliency models, then propose a new combination model. This proposed model, CIT, incorporates salient tokens as extra inputs to their model, which guides the summary generation and ensures covering all key information. Experiments of applying these combinations to the abstractive ATS task show significant performance enhancements, as shown in Section 7.2.1. Moreover, Dou et al., [122] use various kinds of external knowledge as an input to guide and control the output

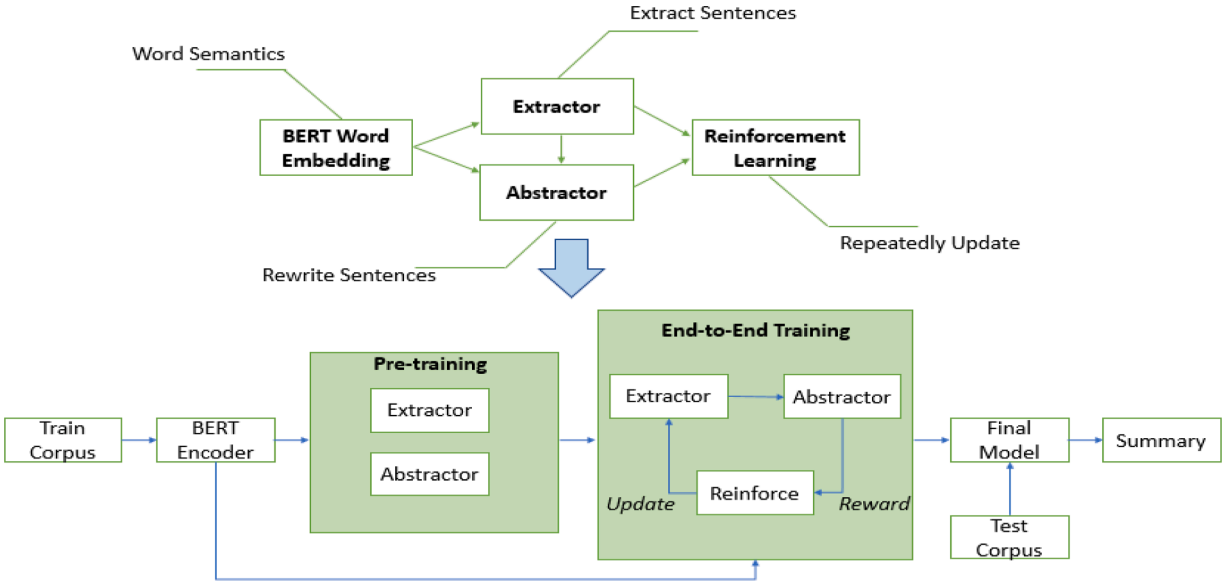

Fig. 12. Extractive-abstractive combination with BERT and RL [64].

and increase faithfulness. As a result, they build a general framework, GSum, by adopting a Transformer instantiated with BERT or BART for the encoder and decoder components. Their model uses two encoders to encode the input document and the guidance signal, which are both attended to by a decoder when generating outputs. To predict guidance signals, Oracle extractions are used at the training time, and MatchSum model [172] is used at the test time.

More recently, utilizing the two-stage learning strategy, discussed in Section 3.1.5, in abstractive ATS has pushed the SotA performance to a higher level by combining the outputs of multiple SotA models. This ensemble approach is utilized by [132,133] demonstrating that abstractive ATS models can generate better summaries than their original outputs. [132] solve different gap problems of stacking and reranking methods by introducing a two-stage learning general framework, Refactor, where the combined models share the same parameters. Specifically, Refactor is first trained to identify candidate summaries from document sentences (pre-trained Refactor), and then trained to identify candidate summaries from different base model outputs (fine-tuned Refactor). As a result, Refactor utilizes the complementarity of top-performing abstractive ATS models (such as BART, PEGASUS, and GSUM) boosting their performance on CNN/DM and XSum datasets. Another significant work in this field has recently been accomplished by [133] who present SimCLS, a two-stage learning contrastive framework. SimCLS uses BART (for CNN/DM corpus) and Pegasus (for XSUM corpus) as the first learning stage to generate candidate summaries. Then, as a second learning stage, RoBERTa is used as an evaluation model and trained with contrastive learning to predict the performance of candidate summaries based on the source document. As a result, SimCLS boosts BART performance by a 2.51 ROUGE-1 score dominating all SotA models with respect to ROUGE metric. The full results are demonstrated in Section 7.2.1. Table 7 compares the architecture details of the aforementioned work of adapting Transformers and PTLMs.

6. Datasets

6.1. Abstractive ATS datasets

Various annotated datasets are designed for abstractive ATS targeting different tasks. Some are proposed for training models on short input documents to generate short summaries, while others are proposed for long documents and summaries. Moreover, these corpora are gathered from various domains, such as news articles, scholarly articles, stories., etc. Most datasets are written in English. However, there are several datasets written in other languages. In general, the majority of these datasets are news articles in the English language. Details of the dataset in the abstractive ATS field are described in Table 10.

Firstly, for **short documents**, DUC[2], Gigaword [28,185,186], CNN/DM [16,187], and XSum [188] are the commonly used abstractive ATS benchmarks. **Document Understanding Conference** (DUC) consists of several publications (DUC-2001 - DUC-2007). For example, DUC-2004 corpus contains 500 articles, each with 4 reference summaries, compiled from the New York Times Associated Press Wire. This dataset targets two tasks given a short article. The first task is to generate very short summaries (<=75 Bytes). The second task is to generate short summaries (<=665 Bytes). However, this dataset is rather small and thus unsuitable for training the data-hungry DL neural models that require a wealth of data to learn. Instead, it is mostly used to evaluate and test models to assess their generalizability and robustness capabilities.

Gigaword corpus [185,186] is filtered by Rush et al., [28] to contain 4M examples of article-title pairs. The task of this dataset is to generate a headline given the first sentence of an article. This dataset is collected from seven sources of news article publishers, which are Agence France-Presse-English Service (afp_eng), Associated Press Worldstream-English Service (apw_eng), Central News Agency of Taiwan-English Service (cna_eng), Los Angeles Times/Washington Post Newswire Service (ltw_eng), Washington Post/Bloomberg Newswire Service (wpb_eng), New York Times Newswire Service (nyt_eng), and Xinhua News Agency-English Service (xin_eng). However, since most of the modern abstractive ATS research work focuses on summarizing multiple sentences, the Gigaword corpus is rarely used in training modern models.

Later after that, the QA task dataset **CNN/DM** [187], was modified by Nallapati et al., [16] for the abstractive ATS task. This dataset then became the most widely used in abstractive ATS research for empirical comparisons. The CNN/DM dataset has been released in two versions: anonymized and non-anonymized. In the anonymized version, named entities, such as the names of people, sites, countries, etc., are replaced by unique identifiers. Whereas in the non-anonymized version, the original text is preserved without modifications. Nallapati et al., [16] have used the anonymized version in their experiments, while See et al., [18] favoured using the original non-anonymized version. Most of the following researchers followed See et al.'s selection and used the original non-anonymized version of the CNN/DM dataset. This annotated dataset contains 287,227 training pairs, 13,368 validation pairs, and 11,490 test pairs. Each of these pairs consisted of an English news article from either CNN (93K) or The Daily Mail (220K), accompanied by a multi-sentence summary as bullet points concatenated to form regular sentences. Overall, the average document length in this corpus is around 30 sentences, with an average of almost 700 tokens per sentence. While the average length of summaries is 3-4 sentences, with an average of approximately 50 tokens each.

Recently, **XSum** [188] was created, providing more abstractive examples to enhance the novelty levels of abstractive ATS models, as described in Section 6.2. This corpus is divided into 90% for training pairs, and 5% for each validation and test pairs, collected from BBC articles from various fields. In this dataset, the input is a single article that contains around 20 sentences of 400 tokens. And the output is a single short sentence of around 23 tokens summary that answers the question "What is the article about?".

[2] http://duc.nist.gov/

Table 7
A comparison among various PTLM-based abstractive ATS models.

Research	Year	Model name	Main idea	Architecture(s) used	PTLM(s)/Framework(s) used
Egonmwan and Chali [85]	2019	TRANS-ext + filter +abs	Dual-Encoding	Transformer, GRU	-
Zhang et al., [181]	2019	Two-Stage + RL	Dual-Decoding	Transformer	BERT
Khandelwal et al., [182]	2019	Transformer LM	Decoder-Only	Transformer	GPT-2
Hoang et al., [183]	2019	Transformer-SM	Various training strategies	Transformer	GPT
Edunov et al., [184]	2019	SRC-ELMO+SHDEMB	Feature-based adaptation	Transformer, LSTM	BERT, GPT, ELMo
Wang et al. [64]	2019	(m7) BEAR (large + WordPiece)	Ext+AbstExt+AbstExt+Abst	Transformer, GRU	BERT
Bae et al., [149]	2019	BERT-ext + abs + RL + rerank		Transformer, LSTM	BERT
Liu and Lapata [164]	2020	BERTSUMEXTABS		Transformer	BERT
Dou et al., [122]	2020	GSum (BART+MatchSum)	Generation-guide mechanism	Transformer	BART, BERTSUMABS
Saito et al., [121]	2020	CIT	Dual-EncodingGeneration-guide mechanism	Transformer	BART, RoBERTa
Zhu et al., [127]	2020	Corrected by FC	Dual-EncodingGeneration-guide mechanism	Transformer	BART, RoBERTa, UniLM
Liu et al [132]	2021	Refactor	Dual-EncodingGeneration-guide mechanism Dual-Encoding Two-stage learning	Transformer	GSum, Pegasus
Liu and Liu [133]	2021	SimCLS		Transformer	BART, Pegasus, GSum, ProphetNet

Table 8 compares CNN/DM and XSum in detail. Most of these details are collected from [16,29,98,164,183,188]. Density and compression ratio measurements are described in Section 6.2.

Secondly, for **long documents**, several datasets are collected from different fields such as scientific papers **(arXiv** [97] and **PubMed** [97]), patent documents **(BIGPATENT** [98]), congressional bills (**BillSum** [189]), and accountability reports (**GOVREPORT** [156]). Currently, **arXiv, PubMed**, and **BIGPATENT** are the most widely used long-document abstractive ATS benchmarks. The task of these datasets is to generate an abstract from a paper body (or patent description). In these datasets, the average document length is ranging from 3k-5k tokens, and their accompanied summaries lengths are averaged around 200 tokens. Table 9 compares these datasets in detail and shows some important statistics. Density and compression ratio metrics are defined in Section 6.2.

More recently, GOVREPORT [156] and BillSum [189] datasets were published with a more even spread of important information throughout the document. GOVREPORT includes significantly longer documents (9.4K tokens) and summaries (550 tokens) collected from the U.S. Government Accountability Office and the Congressional Research Service. On the other hand, BillSum dataset [189] is collected from US Congressional and California state bills including an average of 1.8K-word input documents and 200-word summaries.

However, both GOVREPORT and GOVREPORT can be used for evaluation purposes as the number of their examples is relatively small (19.4K and 23.5K documents, respectively) and insufficient for training DL-based models.

Additionally, **Text Analysis Conference** (TAC)[3] has released several benchmarks for different ATS types such as update ATS (TAC2008), opinion ATS (TAC2008 and TAC2009), guided ATS (TAC2010 and TAC2011), and multi-lingual multi-document ATS (TAC2011). The latest TAC edition for ATS, TAC2014[4], focuses on summarizing scientific articles by incorporating details from other citing papers to improve the quality and informativeness of the generated summaries. Specifically, it covers 20 topics in the biomedical domain, each consisting of a reference paper and several papers that cite it. Each reference paper is accompanied by four scientific summaries written by experts in the biomedical field. The length of these summaries is less than 250 words. However, the TAC2014 dataset is relatively small and not suitable for training deep neural abstractive ATS models.

Thirdly, for **other domains**, the **WikiHow** dataset [190] consists of instructional steps from WikiHow.com accompanied by summaries, and **Reddit TIFU** [191] contains stories examples along with their summaries.

Finally, for **other languages, LCSTS** [112] contains 2.4M Chinese article-summary pairs, compiled from China-based microblogs, Sina Weibo, Verified Chinese media and organizations Weibo.com. LCSTS is the largest non-English dataset. Moreover, **MLSUM** [192] contains over 1.5M examples of multi-lingual articles collected from five sources of five different languages: Le Monde (French), Suddeutsche Zeitung (German), El Pais (Spanish), Moskovskij Komsomolets (Russian), and Internet Haber (Turkish). The task is to generate a multi-sentence abstractive summary given an article written in the same language.

Table 10 outlines the specifications for the most widely used ATS datasets.

[3] https://tac.nist.gov/
[4] https://tac.nist.gov/2014/

Table 8
CNN/DM and XSum corpora statistics.

		CNN/DM	XSum
Split Size	**Train**	287,227	204,045
	Validation	13,368	11,332
	Test	11,490	11,334
	Total	312,085	226,711
Average (Sentences)	**Input**	30.71	19.77
	Summary	3.78	1.00
Average (Tokens)	**Input**	685.17	431.07
	Summary	51.99	23.26
Median (Tokens)	**Input**	777	359
	Summary	59	25
90 Percentiles (Tokens)	**Input**	1439	920
	Summary	93	32
Novelty %	**1-gram**	12.70	34.88
	2-gram	46.29	78.78
	3-gram	65.04	92.03
	4-gram	75.56	96.80
Compression Ratio		13.0	18.8
Density		3.8	1.2

6.2. Novelty of abstractive ATS datasets

Grusky et al., [193] measures the novelty of datasets using coverage and density metrics. Coverage metric calculates the percentage of words in the generated summary that are extracted from the original document. While density metric computes the average length of the extractive fragment to which each word in the summary belongs, i.e., extracted spans:

$$Coverage(A,S) = \frac{1}{|s|} \sum_{f \in F\ (A,S)} |f| \tag{11}$$

$$Density(A,S) = \frac{1}{|s|} \sum_{f \in F\ (A,S)} |f|^2 \tag{12}$$

where A is the original document, S is the generated summary, and $f \in F(A, S)$ are all extracted fragments.

Both measurements could be translated in terms of novelty as follows: Higher coverage and density scores mean more extractiveness. Whereas lower coverage and density scores indicate more new words and smaller extracted text spans, respectively, which means higher novelty levels.

Another measurement suggested by [193] is the compression ratio between the original document and the generated summary. This score estimates the length of the generated summary compared to the original text. The compression ration can be calculated as follows:

$$Compression(A,S) = \frac{|A|}{|S|} \tag{13}$$

A higher compression ratio means more challenges with the long-text problem, described in Section 8.1. These metrics can be used to compare datasets for their level of novelty and how long their summaries relate to the original text.

As can be seen in Fig. 13 and Table 8, although the CNN/DM corpus [16,187] is considered as an abstractive ATS dataset, it has high degrees of density and coverage and low ratios of novel n-grams, which means that its extractiveness level is high and thereby it is skewed towards extractive summaries [193]. In contrast, summaries of XSum [188] and Reddit TIFU [191] corpora are more abstractive as they have lower density and coverage scores and higher ratios of novel n-grams [7]. Moreover, Fig. 14 shows that XSum has the highest levels of novelty among all other datasets.

For long document ATS datasets, Bigpatent [98] encompasses the best summaries in terms of novelty with the lowest density scores, as shown in **Table 9**. However, the two scientific papers corpora, arXiv and PubMed [97] have a tendency toward extractive approaches. As shown in **Fig. 15,** both datasets have high percentages of n-grams copied from the source article to the target summary.

7. Evaluation metrics and comparisons

7.1. Abstractive ATS evaluation metrics

Because measuring the correctness of summaries is difficult, most abstractive ATS research uses ROUGE [5] as the standard evaluation metric. ROUGE, stands for Recall-Oriented Understudy for Gisting Evaluation, includes a set of scores for evaluating abstractive ATS and MT tasks. ROUGE metric is designed to measure the lexical similarity, i.e., the overlap, of n-grams between the resulting summary and a reference, usually a human-written summary. Specifically, there are three main measurements for evaluating

Table 9
arXiv, PubMed, and Bigpatent corpora comparison.

Dataset	Examples Split				Input Statistics			Summary Statistics			Compression Ratio	Density
	Train	Valid.	Test	Total	Tokens Average	Tokens Median	Tokens 90 percentiles	Tokens Average	Tokens Median	Tokens 90 percentiles		
arXiv	203,037	6436	6440	215K	4938	6151	14405	220	171	352	39.8	3.8
PubMed	119,924	6633	6658	133k	3016	2715	6101	203	212	318	16.2	5.8
Bigpatent	1,207,222	670,68	67072	1.3M	3573	3082	7693	116.5	123	197	36.4	2.4

Table 10
Specifications of commonly used ATS datasets.

Dataset	Date Created	Author(s)	# of Pair-Examples	Source / Period	Task	Input	Output
News Articles (Short Documents)							
Document Understanding Conference DUC	2001-2007	http://duc.nist.gov/	500 articles (DUC-2004)	New York Times Associated Press Wire (DUC-2004)	Sentence-level Summarization	Single article	Task 1: Very-Short summaries Task 2: Short summaries
Gigaword	2015	Graff et al., [185], Napoles et al., [186], and Rush et al., [28]	4M	Seven sources: 1994-2010	Abstractive Head-Line Generation	Article's first sentence	Sentence Paraphrases
CNN/DailyMail	2015	Hermann et al., [187], and Nallapati et al. [16]	312K	CNN newspaper (93K) The Daily Mail newspaper (219K) 2007-2015	Abstractive multi-sentence summarization	Single article	Multi-sentence summaries.
XSum	2018	Narayan et al. [188]	227k	BBC articles 2010-2017	Abstractive single-sentence Summarization	Single article	Single-sentence summaries
Scientific Articles (Long Documents)							
arXiv	2018	Cohan et al. [97]	215k	arXiv.org	Long document ATS	Paper body	Abstract
PubMed	2018	Cohan et al. [97]	133k	PubMed.com	Long document ATS	Paper body	Abstract
BIGPATENT	2019	Sharma et al. [98]	1.3M	Google Patents Public Datasets using BigQuery, after 1971 across nine different technologies	Long document ATS	Patent description	Abstract
Other Domains							
WikiHow	2018	Koupaee& Wang [190]	200K	Dataset of instructions from WikiHow.com	Abstractive multi-sentence summarization	Article of multiple instructions	Multi-sentence summaries
Reddit TIFU	2019	Kim et al. [191]	120K	Informal stories from Reddit.com 2013-2018	Abstractive summarization	Online post	Long or short summary sentence
Other Languages							
LCSTS	2015	Hu et al. [112]	2.4M	Weibo.com	Abstractive Chinese Short summarization	Single Article	Short Summary
MLSUM	2020	Scialom et al. [192]	1.5M+	Multilingual dataset Five sources for five different languages 2010-2019	Abstractive Summarization	Article written in one language	Multi-sentence summary generated in the same language

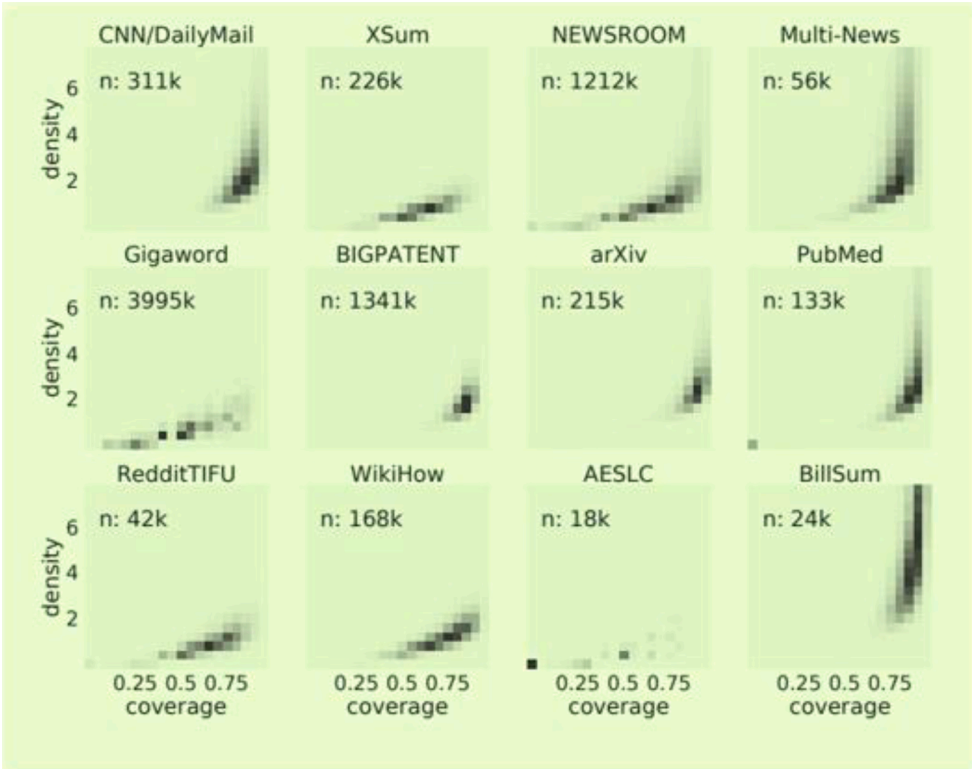

Fig. 13. Coverage and density comparison of various datasets. Darker blocks indicate higher percentages, and n is the number of examples in the dataset [7].

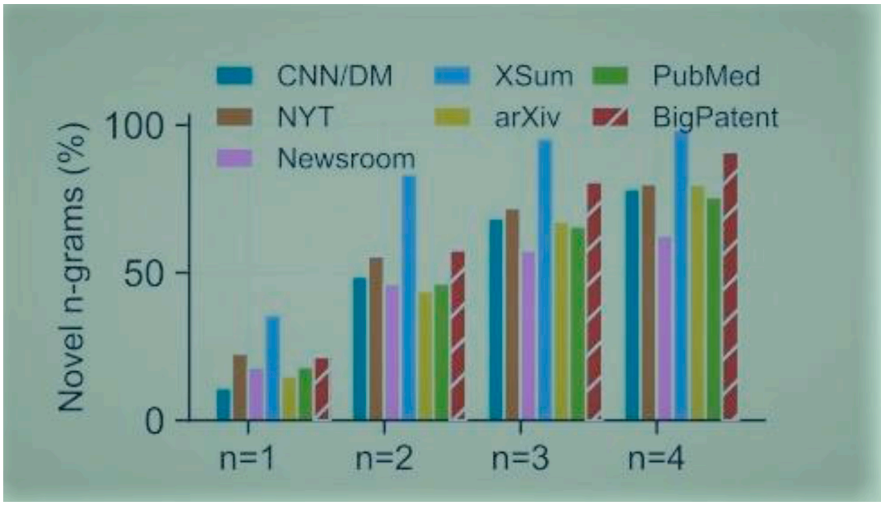

Fig. 14. Novelty levels of short-document abstractive ATS datasets [98].

word overlap, namely, Recall, Precision, and F1.

Recall measures how much the generated summary captures the reference summary, and it is calculated as follows:

$$recall = \frac{ow}{rw} \tag{14}$$

where *ow* is the number of overlapping words, and *rw* is the word count for the reference summary. However, the problem of this measurement is that it prefers longer summaries.

Precision solves the previous problem by measuring how much of the generated summary is relevant, i.e., the suitability of the

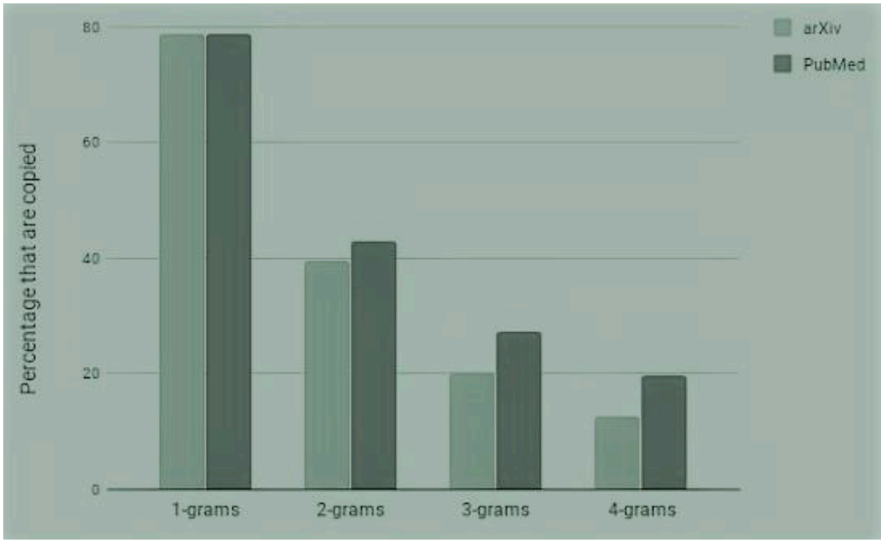

Fig. 15. The percentage of n-grams copied from the source document to the target summary of arXiv and PubMed corpora [75].

generated summary, and it is calculated as follows:

$$precision = \frac{ow}{gw} \tag{15}$$

where *gw* is the number of generated summary words. In contrast to the recall score, the precision score favours shorter summaries.

F1 balances both recall and precision grades by computing their harmonic mean as follows:

$$F1 = \frac{2 * recall * precision}{recall + precision} \tag{16}$$

ROUGE-F1 is the most popular standard used to measure three scores of ROUGE, namely, ROUGE-1, ROUGE-2, and ROUGE-L. ROUGE-1 measures the overlap of unigrams, i.e., every single word, between the resulting and reference summaries. Whereas ROUGE-2 measures the overlap of bigrams, i.e., every two consecutive words, between the resulting and ground-truth summaries. Finally, ROUGE-L measures the longest common sequence between the resulting and reference summaries.

Although ROUGE is the most widely used abstractive ATS metric, it has several limitations. ROUGE, which is used in the evaluation phase, is inconsistent with the cross-entropy loss function used during the training phase. Also, it is non-differentiable. These two problems of ROUGE were solved efficiently using RL-based approaches [99, 23], as discussed in Section 4.1.

Furthermore, higher ROUGE scores do not actually guarantee higher quality and higher readability [99], but on the contrary, they often receive lower human ratings [137]. This is because ROUGE is unable to capture similarities at such high levels as semantic similarities but only focuses on local similarities.

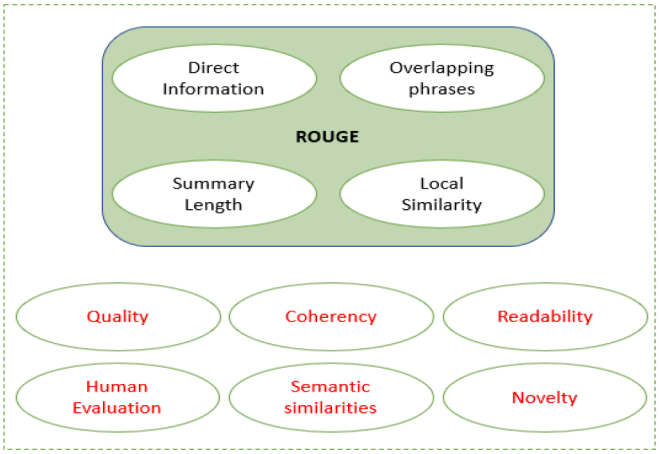

Fig. 16. Focused and unfocused aspects of ROUGE metric.

Also, ROUGE metric fails to capture the novelty nature in high-quality summaries [23]. That is because summaries with high levels of novelty are most likely to include synonyms other than expressions included in the ground-truth summaries, resulting in reduced overlap scores and thereby lower ROUGE scores. Fig. 16 depicts the aspects on which ROUGE metric is focused.

For other metrics, Meteor [194] and BLEU [195] are two well-known metrics that are mainly designed to measure MT results. However, these two metrics can also measure ATS results as well [18,113].

Recently, more robust semantic-based metrics have been proposed for abstractive ATS that correlate better with human evaluations, such as VERT [196], MoverScore [197], and BERTScore [198]. VERT compares model-generated and ground-truth summaries by combining document-level similarity and word-level dissimilarity scores to measure the semantic similarity of the resulting summary. VERT has recently used by Boutkan et al., [106] in evaluating their results. In addition, MoverScore combines contextualized embeddings with Earth Mover's Distance [199] that measures semantic distance between texts. More recently, BERTScore uses contextual embeddings to compute semantic similarities. MoverScore and BERTScore are starting to gain popularity in evaluating abstractive ATS research models [128,133].

Table 11
ROUGE results of the best short-document abstractive ATS models on CNN/DM, XSum, and Gigaword datasets grouped by the model architecture. R is short for ROUGE. '* ' indicates the model is trained and evaluated on the anonymized CNN/DM dataset. BOLD indicates the top performing models in each category. Underlined indicates the overall top performing models.

Research	Year	Model name	CNN/DM				XSum	Gigaword
			R-1	R-2	R-L	R-AVG	R-1/R2-/R-L	R-1/R2-/R-L
Best Deep Sequence-to-Sequence Models								
Nallapati et al.* [16]	2016	words-lvt2k-temp-att	35.46	13.30	32.65	27.14	-	35.30/16.64/32.62
See et al. [18]	2017	PG + coverage	39.53	17.28	36.38	31.06	28.10/8.02/21.72 [188]	-
Paulus et al.* [99]	2017	ML, with intra-attention	38.30	14.81	35.49	29.53	-	-
Narayan et al. [188]	2018	T-CONVS2S	-	-	-	-	**31.89/11.54/25.75**	-
Al-Sabahi et al. [70]	2018	Bidir_Rev_Cov	**42.60**	18.80	38.50	33.30	-	-
Zhang et al. [77]	2019	CNN-2sent-hieco-RBM	42.04	**19.77**	**39.42**	**33.74**	-	**37.95/18.64/35.11**
Best RL-Based Models								
Paulus et al. * [99]	2017	RL, with intra-attention	41.16	15.75	**39.08**	32.00	-	-
Chen and Bansal * [20]	2018	rnn-ext + abs + RL + rerank	40.88	17.8	38.54	32.41	-	-
Gehrmann et al. [22]	2018	Bottom-Up	41.22	18.68	38.34	32.75	-	-
Celikyilmaz et al. [100]	2018	DCA	**41.69**	**19.47**	37.92	**33.03**	-	-
Best Fine-Tuned PTLMs								
Lewis et al. [10]	2019	BART	44.16	21.28	40.90	35.45	45.14/22.27/37.25	-
Raffel et al. [8]	2019	T5-11B	43.52	**21.55**	40.69	35.25	-	-
Dong et al. [163]	2019	UniLM	43.33	20.21	40.51	34.68	42.14/19.53/34.13	38.45/19.45/35.75
Zhang et al. [7]	2020	PEGASUS (HugeNews)	44.17	21.47	41.11	35.58	**47.21/24.56/39.25**	39.12/19.86/36.24
		PEGASUS (C4)	43.90	21.20	40.76	35.29	45.20/22.06/36.99	38.75/19.96/36.14
Yan et al. [159]	2020	ProphetNet	44.20	21.17	41.30	35.56	-	**39.51/20.42/36.69**
Zou et al. [171]	2020	STEP	44.03	21.13	41.20	35.45	43.02/20.11/35.34	-
Xiao et al. [160]	2020	ERNIE-GEN (16GB)	44.02	21.17	41.26	35.48	-	39.25/20.25/36.53
		ERNIE-GEN (430GB)	**44.31**	21.35	**41.60**	**35.75**	-	-
Bao et al. [170]	2020	UNILMv2-Base-relative position bias	43.45	20.71	40.49	34.88	(Base) 44.00/21.11/36.08	-
Zhu et al. [127]	2021	UNILM Corrected by FC	42.75	20.07	39.83	34.22	42.18/ 19.53/34.15	-
Best PTLM-Based Models								
Bae et al. [149]	2019	BERT-ext + abs + RL + rerank	41.90	19.08	39.64	33.54	-	-
Song et al. [147]	2020	Beam+BPNorm	-	-	-	-	-	**39.19/20.38/36.69**
		Beam+SBWR	-	-	-	-	-	**39.08/20.47/36.68**
Dou et al. [122]	2020	GSum (BART+MatchSum)	45.94	22.32	42.48	36.91	45.40/21.89/36.67	-
Saito et al. [121]	2020	CIT	45.74	**22.50**	42.44	36.89	**45.42/22.13/36.92**	-
Liu et al., [132]	2021	Refactor-BART-rerank-FT	45.15	21.70	42.00	36.28		
		Refactor-GSUM-rerank-FT	46.18	22.36	42.91	37.15		
		Refactor-Pegasus-rerank-PT	-	-	-	-	47.45/24.55/39.41	
Liu and Liu [133]	2021	SimCLS	**46.67**	22.15	**43.54**	**37.45**	**47.61/24.57/ 39.44**	

Fig. 17. ROUGE results of best models on the CNN/DM dataset.

7.2. Comparisons of selected models

This section compares selected models that achieve the best results in both short and long document summarization. As mentioned earlier, ROUGE is currently the commonly used metric in Abstractive ATS research work. Therefore, all comparisons in this section will be based on ROUGE scores to ensure fairness.

For short-length document abstractive ATS research, the ROUGE results for modern neural models have improved over time. First, by using different DL-based techniques and architectures in sequence-to-sequence models, such as attention mechanism [28], copy mechanism [16], coverage mechanism [18], generation guide [49,63], CNN [77], and bidirectional encoder-decoder architecture [70], which are described in Section 3. Next, by incorporating RL approaches [20,22,99,100], which are discussed in Section 4. Then, the adaptation of PTLMs to abstractive ATS showed significant improvements in performance and ROUGE scores [7,8,10,159,163], which are described in Section 5.

Currently, an important research trend that yields to SotA performance is to find the most appropriate PTLMs and model architectures for abstractive ATS using different techniques, such as knowledge-enhanced and tow-stage learning, to efficiently adapt pre-trained global representations of these PTLMs for abstractive ATS to enhance the outcomes [121,122,132,133], as discussed in Section 8.3.

Recently, summarizing very long documents that contain thousands of words has attracted researchers due to its importance and usefulness to societies. This research trend particularly comes after incorporating huge contextual PTLMs and the availability of new long-document datasets shown in Section 6.1.

7.2.1. Short-document summarization models

Below is a comparison of the best performing short-document abstractive ATS models based on ROUGE metric [5], which is the most widely used abstractive ATS metric, on three popular datasets, CNN/DM, XSum, and Gigaword, grouped by model architecture: Deep sequence-to-sequence models, RL-based models, fine-tuned PTLMs, and PLTM-based models.

As mentioned in Table 11, SimCLS [133] delivers the best performance on the most widely used dataset, CNN/DM, boosting the performance of top-performing models in ROUGE-1 and ROUGE-L scores with SotA results of 46.67 and 43.54, respectively. While all other models in ROUGE-2 were dominated by the CIT model [121]. Also, SimCLS [133] outperforms all models on the XSUM corpus. Finally, for the Gigaword dataset, ProphetNet [159] and Beam+SBWR [147] are the top-performing models offering the best ROUGE scores.

Table 11 compares the top-performing short-document abstractive ATS models discussed throughout this paper on the CNN/DM, XSum, and Gigaword datasets with respect to the ROUGE scale. Fig. 17 shows the comparison charts for these models on the CNN/DM dataset.

7.2.2. Long-document summarization models

As discussed in Section 2.7, research on long-document abstractive ATS is scanty and still at the beginning compared to short-document abstractive ATS. Table 12 compares selected long-document abstractive ATS models that achieve the best ROUGE scores on three datasets: arXiv [97], PubMed [97], and BigPatent [98]. These corpora are described in detail in Section 6.1. Fig. 18 shows the comparison charts for these models on arXiv and PubMed datasets. As noted, BigBird-PEGASUS [29] and LSH+HEPOS [156] are the best performing models.

8. Discussions

8.1. Problems, solutions, and current challenges

During the past six years, research works to improve outcomes for abstractive ATS have faced many challenges. This section discusses the evolution of problems and their solutions as well as current challenges in abstractive ATS research in the three main models discussed in this article: **Deep neural sequence-to-sequence models, Deep RL-based models, and PTLMs.**

Table 12
ROUGE results of the best long-document abstractive ATS models. R is short for ROUGE. BOLD indicates the top performing models.

Research	Year	Model name	arXiv R-1/R-2/R-L	PubMed R-1/R-2/R-L	BigPatent R-1/R-2/R-L
Cohan et al. [97]	2018	Attention Seq2Seq	29.30/6.00/25.56	31.55/8.52/27.38	-
Cohan et al. [97]	2018	Pointer-Generator	32.06/9.04/25.16	35.86/10.22/29.69	-
Cohan et al. [97]	2018	Discourse-Aware	35.80/11.05/31.80	38.93/15.37/35.21	-
Pilault et al. [65]	2020	TLM-I+E (G,M)	41.62/14.69/38.03	42.13/16.27/39.21	38.65/12.31/34.09
Xiao and Carenini [180]	2020	ExtSum-LG++RdLoss	44.01/17.79/39.09	45.30/20.42/40.95	-
Xiao and Carenini [180]	2020	ExtSum-LG+MMR-Select+	43.87/17.50/38.97	45.39/20.37/40.99	-
Zhang et al. [7]	2020	PEGASUS	44.21/16.95/38.83	45.97/20.15/41.34	52.29/33.08/41.66
Gidiotis and Tsoumakas [75]	2020	DANCER PEGASUS	45.01/17.60/40.56	46.34/19.97 /42.42	-
Zaheer et al. [29]	2020	BigBird-PEGASUS	46.63/19.02/41.77	46.32/20.65/42.33	**60.64/42.46/50.01**
Huang et al. [156]	2021	LSH+HEPOS	**48.24/20.26/41.78**	**48.12/21.06/42.72**	

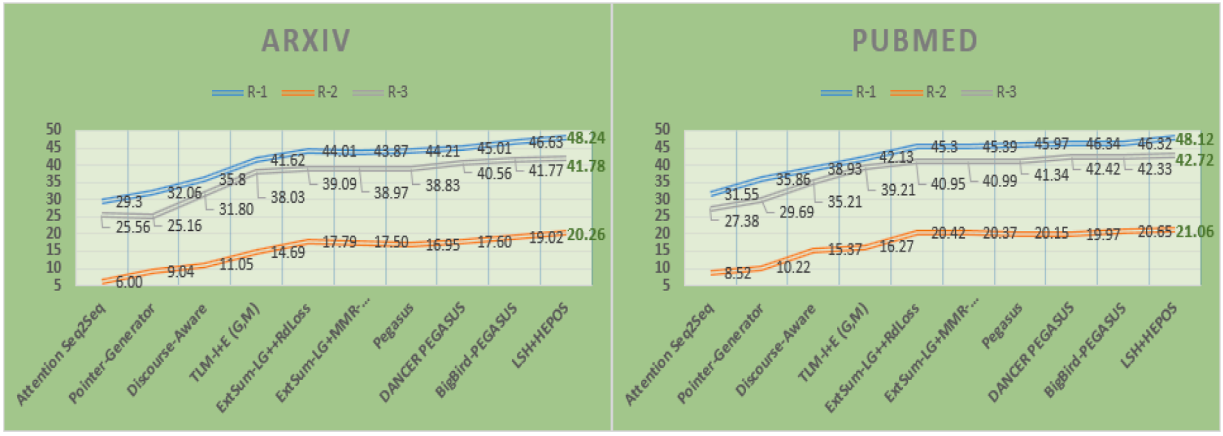

Fig. 18. ROUGE results of best models on arXiv and PubMed datasets.

Deep neural sequence-to-sequence models normally focus on summarizing short-length documents. Various approaches have been proposed to deal with several issues, such as long-term dependencies, vanishing and exploding gradients, inaccurate and fake detailing, OOV words, repetition, lack of faithfulness, as well as difficulty in controlling outputs. These problems occur especially when processing longer entries.

Many solutions have been offered based on deep sequence-to-sequence-based approaches. For the problem of long-term dependencies, several architectures and techniques have been proposed. Firstly, as discussed in Section 2.4, LSTM [71] and GRU [67] were found mainly to deal with such dependencies in the text and overcome the gradient-vanishing problem [72] using gating mechanisms. However, LSTM and GRU still struggle to handle such dependencies when the input document is lengthy. More recently, RUM [74] was discovered to better deal with the problem of long-term dependencies. Similarly, the attention mechanism is proposed to handle longer dependencies, as discussed in Section 3.1.1.

Next, the copy mechanism, as discussed in Section 3.1.3, was proposed to solve the problems of OOV words and inaccurate factual details by allowing the model to borrow some phrases from the original text to ensure generating accurate details. However, this mechanism leads to the problem of low novelty as the models tend to include a large number of phrases from the original text in the summary generated. Solutions to this problem are provided through RL- and TL-based approaches. These shall be discussed later in this section.

Then, the coverage mechanism, which is discussed in Section 3.1.4, was proposed to ensure that there is no repetition in the generated summary. It works by tracing the summary phrases, paying more attention to un-summarized parts, and less attention to the parts that have already been summarized.

Moreover, the knowledge-enhanced mechanism (Section 3.1.5) was proposed to increase the readability, informativeness, and faithfulness of the generated summaries. It integrates other useful information, i.e., training signals, entered to the model with the input sequence to control the outputs to improve the quality and accuracy of the results.

Although the previously discussed deep sequence-to-sequence based approaches have succeeded in solving some problems efficiently, there are other issues that require methods beyond DL such as RL- and TL-based approaches. Deep sequence-to-sequence models still produce low-quality and incoherent summaries, especially when the length of the input text is increased. Also, their summaries resulted in high copy rates, including many phrases from the source document leading to the problem of low novelty. Moreover, supervised deep sequence-to-sequence models are usually trained to expect the next word in the generated summary based only on the maximum likelihood objective function. This in turn leads to two inconsistency problems between the training and testing phases in terms of reference summaries and metrics used: the exposure-bias and the loss/evaluation mismatch problems. The exposure-bias problem [138] occurs when the model receives its inputs from different sources during the training and testing steps. In training, the decoder deals with the ground-truth summaries to be trained and generate results, while the model relies on its own output to produce the tokens during the testing phase. On the other hand, the problem of inconsistency between training and testing evaluation measurements occurs when the model uses dissimilar loss/evaluation metrics during training, i.e., cross-entropy loss, and testing, i.e., ROUGE. The consequences of these two problems may lead to error accumulation while generating the output at test time, resulting in inconsistent/low-quality summaries [19,139,140]. Another drawback of deep sequence-to-sequence models is that, during testing, the models are unable to generalize to datasets other than those they have already been trained on.

As described in detail in Section 4.1 and summarized in Table 4, solutions to the above problems are offered by incorporating RL approaches. However, deep RL-based models are still incapable to efficiently handle some of the issues. They still generate low-quality, low-novelty summaries, especially for very long documents with thousands of words such as scholarly articles. That is because these models train on limited amounts of annotated data resulting in poor semantic and contextual features of word embeddings. Furthermore, the sequential nature of these models prevents them from taking advantage of parallel processing during training and testing, which leads to low training speeds.

Utilizing Transformer-based PTLMs can efficiently solve the above-mentioned issues. Specifically, Transformers can efficiently

learn very long-range dependencies in very long documents using the self-attention mechanism which can parallelly model similarities between tokens regardless of their positions, as described in Section 2.7 and Section 5.1. Moreover, PTLMs can learn syntactic and semantic knowledge through training on unannotated huge data, then they can be fine-tuned to downstream tasks with annotated and small data. As a result, adapting PTLMs to abstractive ATS has greatly advanced the SotA by optimizing the quality, novelty, and fluency of summaries generated, even for very long documents, in a high-speed training time.

However, quadratic memory and computational complexities are the main drawback of the self-attention mechanism used by the Transformer. Furthermore, the large number of operations and the fixed-length context required when dealing with long sequences are other obstacles. Solutions to these problems are presented by various efficient Transformers (such as Reformer [153], Sparse Transformer [154], Transformer-XL [155]), and self-attentions (e.g. sparse, locality-sensitive hashing, HEPOS [156]), which are further discussed in [157] and [156].

Currently, finding the best PTLM(s) for abstractive ATS and adapting it efficiently remains an open research challenge to refine outcomes and approach human abstracts (Refer to Section 5.2.4).

Table 13 summarizes the aforementioned challenges and other problems discussed earlier throughout this article, together with their solutions. Fig. 19 summarizes the evolution of these problems and their solutions for various abstractive ATS models approaches.

8.2. Abstractive ATS and MT research

From the findings, we can conclude that the research work on MT has received the most attention from researchers, and it leads many NLP areas with respect to performance and quality of results. That is because the MT field received the largest share of datasets among all other NLP tasks [161]. Furthermore, the importance of MT in the economic and academic fields plays an essential role in persuading large businesses and academic institution laboratories to pay more attention to this field.

Both TG tasks, abstractive ATS and MT, are similar in terms of converting text from one format to another, and both require a high-level and deep understanding of the meanings of words and sentences to accomplish their tasks. Consequently, it is not surprising to notice that ATS research traces the MT's steps and applies its techniques and ideas, such as attention mechanism [66], coverage mechanism [118,119], generation guide mechanism [62], and others. Table 14 summarizes the techniques and mechanisms of ATS research work inspired by the MT field.

For that, tracking and utilizing the SotA techniques and advances in the field of MT will enrich the research work of abstractive ATS and enhance the quality of its outcomes.

8.3. Future research directions

As the study concluded for future directions, the most significant and recent open research areas in the abstractive ATS field can be summarized as follows:

i **To find better PTLMs**: Leveraging PTLMs in abstractive ATS models shows significant enhancement in results, refer to Section 7.2. Many PLTMs have been developed with different architectures, features and objectives, refer to Section 5.2. Some of these PTLMs are more suitable for abstractive ATS than others. Many researchers have attempted to adapt PTLMs in various ways to abstractive ATS to enhance their results (refer to Table 7). Finding the best PTLM, or combination of complementary PTLMs, for the abstractive ATS task and efficiently adapting it (them) to enhance the generated abstractive summaries' quality is a research trend.

ii **More efficient Transformers**: Recently, more efficient Transformers and attentions have been introduced with better performance, less memory and computational complexity, and faster and more stable training to handle short and long documents more efficiently. Examples of such architectures include Transformer-XL [155], Sparse Transformer [154], Reformer [153], Switch Transformers [9], and many more. Utilizing these architectures will improve the results of different NLP tasks, including abstractive ATS, especially for long documents. Refer to Section 5.1

iii **Multi-reward**: The use of multi-reward RL, which can be user-defined and non-differentiable rewards, is highly suggested to improve various important aspects of abstractive ATS. Examples of these aspects include readability, coherency, syntax, non-redundancy, sentence ordering, conciseness, information diversity, information coverage, saliency, and entailment [87]. Refer to Section 4.

iv **Semantic-based metrics**: As discussed in Section 4.3 and Section 7.1, ROUGE metric [5], the most common evaluation metric used in abstractive ATS research, provides no insight into novelty, readability, global and semantics similarities. Also, Rouge's high scores summaries often achieve a low human rating [64,106], refer to Fig. 16. Therefore, more semantic-based evaluation metrics need to be developed to transcend the limitations of ROUGE, which relies more on syntactic terms. Recently, VERT, MoverScore, and BERTScore are recently developed as new semantic similarity metrics, but are still not as popular in research as ROUGE. Other attempts were made by [108,137,23,150,151], and [193]. Overall, the area of research for abstract ATS remains weak in finding robust semantic metrics that accurately characterize the competency aspect and correlate well with human assessments. Moreover, the novelty score is inversely proportional to that of ROUGE. Therefore, finding an optimal measurement is still an open research question.

v **Increase novelty levels**: Novelty is the main characteristic that distinguishes abstractive summaries from other genres. In early work, as discussed in Section 4.1 and Section 4.3, most abstractive ATS models suffered from very high copy rates, i.e., low levels of novelty, and thereby the resulting summaries were more likely to be extractive than abstractive. One of the main

Table 13
The evolution of abstractive ATS sequence-to-sequence models: problems and solutions.

Model/Architecture with Problem	Problem(s)	Proposed Solution	Justification for the Proposed Solution
Traditional deep neural networks	• Handling unstructured data • Handling variable-length inputs • Handling input sequence dependencies	RNN [69]	By its sequence nature and ability to feedback prior layers to maintain memory inputs and model problems in time, and its ability to capture unbounded (variable) context.
Vanilla RNN architecture	Vanishing Gradient Problem	• LSTM [71] • GRU [67]	By their ability to capture long-term dependencies
Vanilla RNN architecture	Exploding Gradient Problem	Gradient Norm Clipping strategy [73]	By rescaling their norm whenever it goes over a threshold.
Unidirectional RNN, LSTM, GRU	Limited awareness of the sentence context	Bidirectional RNN, LSTM, GRU [70]	By allowing each hidden state to be aware of the contextual information from both directions, i.e., past and future contexts.
RNN, LSTM, GRU	Sequential nature	CNN [76,77] Transformer [4]	By computing each element in the sequence in parallel during training and evaluation
Greedy search	Words disorder in sentences	Beam Search [107].	By their ability to guarantee the right order of the sentence words.
Beam search	Diversity problem	Diverse Beam Search (DBS) [109–111]	By producing diverse outputs by optimizing for a diversity-augmented objective.
Beam search	Local Optima	Generation guide mechanism (value network) [62]	By predicting long-term rewards to be received in the future.
Basic deep sequence-to sequence models	Long-term dependencies in long sequences	• Attention Mechanisms [3,4,106,200,16,28,66,99,101,103–105] • RUM [74]	• Attention: By allowing the encoder to focus only on the most important parts to save more memory and then work efficiently. • RUM: By handling memory copying and memory recall tasks better than LSTMs and GRUs.
Deep sequence-to-sequence models with Attention	• Inaccurate factual details • OOV words problem	Pointer Generator [14–18,116,117]	By its ability to use either synonyms or original words.
Deep sequence-to-sequence models with Attention and Pointer-Generator	Repeated statements	• Coverage Mechanism [18, 118–120] • RL [99]	By keep tracking of what has been generated in the summary
Deep sequence-to-sequence models with Attention, Pointer-Generator, and Coverage mechanisms	• Unfaithfulness (lack of key information) • Semantic irrelevancy • Fake information • Difficult to control the output • Difficult to control the summary length	Knowledge-enhanced (generation-guide) [49,62,63]	By incorporating useful information (training signals) with the input sequence to be entered into the model to guide the generation process, control the output, and improve the quality and accuracy of the results.
Sequence-to-Sequence models with various DL-based mechanisms	• Large text input • Low novelty • Exposure-bias • Loss/Evaluation mismatch • Lack of generalization	RL [19,20,23,99,100,106, 111,139,140,145]	Details in Table 4
Deep sequence-to-Sequence models with RL approaches	• Poor semantic and context features of embeddings • Sequential nature • Low-speed training • Low-quality summaries • Low-novelty summaries	• Transformers [4] • PTLMs (TL) [7,8,10,127,159,160,163, 164,170,171]	• Transformers can efficiently learn long-range dependencies using the self-attention mechanism which can parallelly model similarities between tokens regardless of their positions. • PTLMs can learn syntactic and semantic knowledge through training on unannotated huge data, then be fine-tuned to end-task annotated data.
Deep sequence-to-Sequence models with TL approaches	• Transformer's self-attention Problems: • Its quadratic memory and computational complexities. • The large number of operations it requires when dealing with long sequences	Efficient Transformers and self-attentions [153–156]	By using different types of attention and architectures that reduce memory complexity, speed up training, train deeper networks with fewer operations, and enable models to understand context beyond a fixed length limitation.

(*continued on next page*)

Table 13 (*continued*)

Model/Architecture with Problem	Problem(s)	Proposed Solution	Justification for the Proposed Solution
	• Its fixed-length context pre-requisite to learn long-term dependencies. • Finding the best PTLM and architecture for ATS • Efficiently Adapting pre-trained representations for ATS		

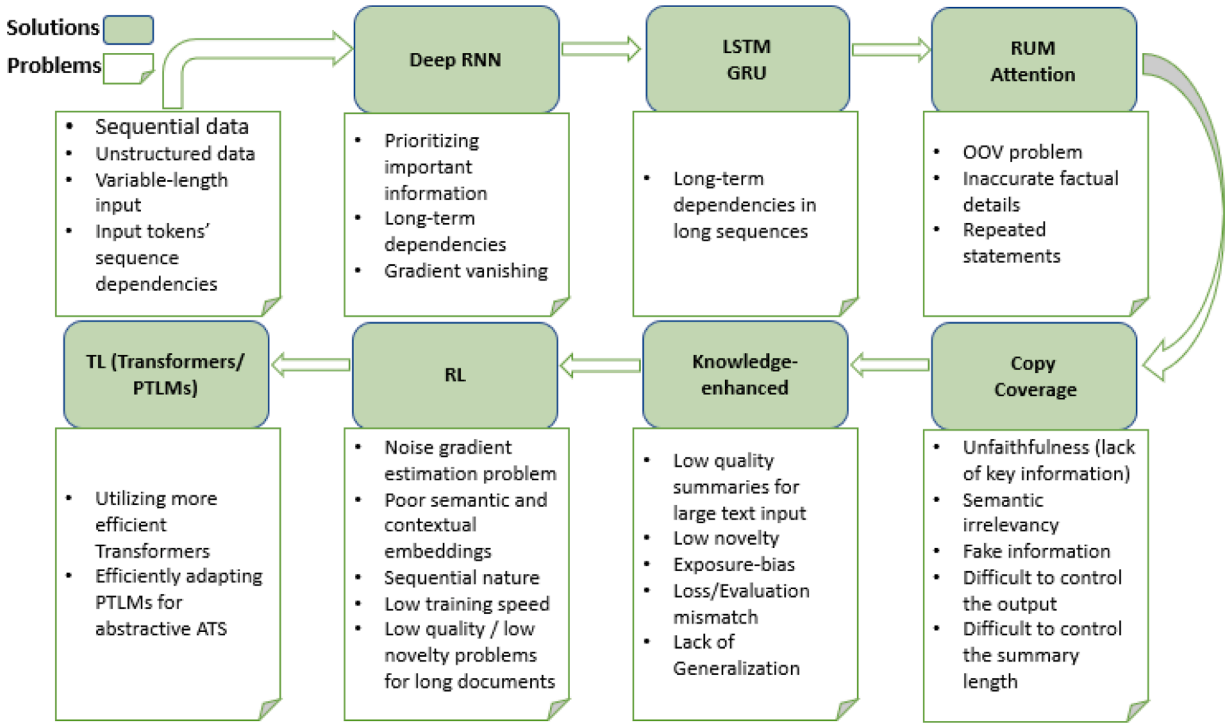

Fig. 19. Evolution of problems and solutions in sequence-sequence-based abstractive ATS models.

reasons for this problem, as discussed in Section 6.2, is the low novelty of datasets, such as CNN/DM [16,187]. Then, by taking advantage of the great enhancements of universal representations resulting from PTLMs and the development of novel datasets at more novelty levels, the novelty of abstractive ATS models has been greatly improved. However, the human levels of novelty in summarization are still a far cry from current abstractive ATS models. Thus, there is a need to focus more on enhancing the novelty levels of the resulting abstractive summaries and generate more datasets with high novelty and quality levels.

vi **Long-document ATS**: As discussed in Section 2.7, it is noticed that the research work on summarizing long-length documents is little compared to short-document summarization. That is because summarizing long documents requires more complex hardware and efficient approaches to efficiently learn very long-range dependencies. Recently, through major developments in massive PTLMs, and the availability of new long-document datasets, the focus has arisen on summarizing very long documents containing thousands of words as scientific papers. This area of research is still at the beginning at it requires further improvements.

vii **Track MT research**: As the study concluded, the tasks of MT and ATS are very similar, and the research work in the field of MT precedes the work on ATS research. As noticed in Table 14, abstractive ATS's most successful research work traces and reuses MT research techniques and mechanisms. Therefore, utilizing the MT research's SotA work will certainly improve abstractive ATS results even more.

viii **Extending best mechanisms to other datasets and types**: As shown in Table 1, there are many different types of ATS. Some are more interesting to researchers than others, such as extractive [27], abstractive, text-output, short- and single-document, generic-domain, and mono-lingual ATS [28]. Also, as discussed in Section 6, many other abstractive ATS datasets exist with different setups. Therefore, extending the SotA approaches and mechanisms that used by SotA abstractive ATS models, such as knowledge-mechanisms and two-stage learning, to other types and datasets of ATS is encouraged.

Table 14
Examples of chosen ATS research work inspired by MT research.

Research idea	Original MT Research	Abstractive ATS Research
Sequence-to-sequence model	Sutskever et al., 2014 [68]	Rush et al., 2015 [28]
Attention Mechanism	Bahdanau et al., 2015 [66]	Rush et al., 2015 [28] Nallapati et al., 2016 [16]
Pointer Network and Copy Mechanism	Luong et al., 2015 [117] (based on Vinyals et al., 2015 [116])	Rush et al., 2015, [28] Gu et al., 2016 [14] Nallapati et al., 2016 [16]
Large Vocabulary Trick (LVT)	Jean et al., 2015 [201]	Nallapati et al., 2016 [16]
Temporal Attention	Sankaran et al., 2016 [200]	Nallapati et al., 2016 [16]
Coverage Mechanism	Tu et al., 2016 [118] Mi et al., 2016 [119]	See et al., 2017 [18]
Bidirectional decoder	Doetsch et al., 2016 [202]	Al-Sabahi et al., 2018 [70]
Reversing the source sequence	Doetsch et al., 2016 [202]	Al-Sabahi et al., 2018 [70]
Convolutional sequence-to-sequence model	Gehring et al., 2017 [80]	Wang et al., 2018 [76]
RL mixed-objective learning function	Wu et al., 2016 [203]	Paulus et al., 2018 [99] Pasunuru and Bansal, 2018 [87] Gehrmann et al., 2018 [22]
Training a neural network to use another fixed network	Gu et al., 2017 [204] Gu et al., 2017 [205]	Chen and Bansal, 2018 [20]
Generation-guide mechanism	He et al., 2017, [62]	Li et al., 2018 [63]
Multi-head attention mechanism	Vaswani et al., 2017 [4]	Boutkan et al., 2019 [106]
Two-stage learning (stacking and reranking)	[130,131]	[132]

ix **Non-English ATS**: ATS research work in languages other than English is still very poor, as described in Section 6. One reason is the lack of available annotated datasets. Therefore, further attention to other languages is needed to enrich the research field of ATS. This can be done by developing new non-English datasets and extending SotA English abstractive ATS models' approaches to other languages. Accordingly, the researchers of [14,112,120,139] worked on Chinese abstractive ATS achieving acceptable results.

9. Conclusion

This study explored recent developments in abstractive ATS research work in various aspects such as types, datasets, techniques, architectures, challenges, solutions, contributions, evaluation metrics, research trends, and SotA models comparisons. This is achieved by focusing on deep neural sequence-to-sequence models, Reinforcement Learning (RL) approaches, and Transfer Learning (TL) approaches, to provide an overview to researchers who wish to explore this field of research. Deep sequence-to-sequence models implemented using encoder-decoder architectures have shown promising results. Significant improvements have been made using architectures based on reinforcement learning and transfer learning, particularly through the use of universal representations learned by various Transformer-based PTLMs. Currently, abstractive ATS research focuses on finding the most effective and appropriate PTLM (s) and how to efficiently adapt their pre-trained global representations to further improve the quality of summaries and bring them closer to human levels of summarization. Overall, for short document abstractive ATS models, SimCLS and CIT models dominate all other models. On the other hand, LSH+HEPOS and BigBird-PEGASUS are the top-performing models for long-document abstractive ATS.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1 Widyassari, A.P., Rustad, S., Shidik, G.F., Noersasongko, E., Syukur, A., Affandy, A., 2020. Review of automatic text summarization techniques & methods. J. King Saud Univ. - Comput. Inf. Sci. https://doi.org/10.1016/j.jksuci.2020.05.006.
2 Gambhir, M., Gupta, V., 2017. Recent automatic text summarization techniques: a survey. Artif. Intell. Rev. 47, 1–66.
3 Tan, J.; Wan, X.; Xiao, J. Abstractive document summarization with a graph-based attentional neural model. *ACL 2017 - 55th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap.*2017, 1, 1171–1181, doi:10.18653/v1/P17-1108.
4 Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., 2017. Polosukhin, I. Attention is all you need. Adv. Neural Inf. Process. Syst. 5999–6009. *2017-Decem*.
5 Lin, C.-Y., 2004. Rouge: A package for automatic evaluation of summaries. Text Summ. branches out 74–81. https://doi.org/10.1253/jcj.34.1213.
6 Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT*2019.
7 Zhang, J., Zhao, Y., Saleh, M., Liu, P.J., 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. Int. Conf. Mach. Learn. 11328–11339.
8 Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv Prepr. arXiv1910.10683*2019.
9 Fedus, W.; Zoph, B.; Shazeer, N. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *arXiv Prepr. arXiv2101.03961*2021.

10 Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv Prepr. arXiv1910.13461*.2019.
11 Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., 2018. Improving language understanding by generative pre-training. OpenAI 1–10.
12 Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., 2019. Language Models are Unsupervised Multitask Learners. OpenAI Blog 1, 9.
13 Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *arXiv Prepr. arXiv2005.14165*2020.
14 Gu, J.; Lu, Z.; Li, H.; Li, V.O.K. Incorporating copying mechanism in sequence-to-sequence learning. *54th Annu. Meet. Assoc. Comput. Linguist. ACL 2016 - Long Pap.*2016, 3, 1631–1640, doi:10.18653/v1/p16-1154.
15 Gulcehre, C., Ahn, S., Nallapati, R., Zhou, B., Bengio, Y., 2016. Pointing the unknown words. In: 54th Annu. Meet. Assoc. Comput. Linguist. ACL 2016 - Long Pap, 1, pp. 140–149. https://doi.org/10.18653/v1/p16-1014.
16 Nallapati, R., Zhou, B., dos Santos, C., Gulçehre, Ç., Xiang, B., 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In: CoNLL 2016 - 20th SIGNLL Conf. Comput. Nat. Lang. Learn. Proc, pp. 280–290. https://doi.org/10.18653/v1/k16-1028.
17 Zeng, W.; Luo, W.; Fidler, S.; Urtasun, R. Efficient Summarization with Read-Again and Copy Mechanism. *arXiv Prepr. arXiv1611.03382*.2016.
18 See, A.; Liu, P.J.; Manning, C.D. Get to the point: Summarization with pointer-generator networks. *ACL 2017 - 55th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap.*2017, 1, 1073–1083, doi:10.18653/v1/P17-1099.
19 Liu, L., Lu, Y., Yang, M., Qu, Q., Zhu, J., Li, H., 2018. Generative adversarial network for abstractive text summarization. In: 32nd AAAI Conf. Artif. Intell. AAAI 2018, pp. 8109–8110.
20 Chen, Y.C., Bansal, M., 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In: ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap., 1, pp. 675–686. https://doi.org/10.18653/v1/p18-1063.
21 Hsu, W.T., Lin, C.K., Lee, M.Y., Min, K., Tang, J., Sun, M., 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In: ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap., 1, pp. 132–141.
22 Gehrmann, S., Deng, Y., Rush, A.M., 2018. Bottom-up abstractive summarization. In: Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018, pp. 4098–4109. https://doi.org/10.18653/v1/d18-1443.
23 Kryściński, W., Paulus, R., Xiong, C., Socher, R., 2020. Improving abstraction in text summarization. In: Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018, pp. 1808–1817. https://doi.org/10.18653/v1/d18-1207.
24 Wang, L., Raghavan, H., Castelli, V., Florian, R., Cardie, C., 2013. A sentence compression based framework to query-focused multi-document summarization. In: ACL 2013 - 51st Annu. Meet. Assoc. Comput. Linguist. Proc. Conf., pp. 1384–1394. *1*.
25 Yao, J.ge, Wan, X., Xiao, J., 2017. Recent advances in document summarization. Knowl. Inf. Syst. 53, 297–336. https://doi.org/10.1007/s10115-017-1042-4.
26 Ibrahim Altmami, N., El Bachir Menai, M., 2020. Automatic summarization of scientific articles: A survey. J. King Saud Univ. - Comput. Inf. Sci. https://doi.org/10.1016/j.jksuci.2020.04.020.
27 Nallapati, R., Zhai, F., Zhou, B., 2017. SummaRuNNer : a recurrent neural network based sequence model for. In: Thirty-First AAAI Conf. Artif. Intell., pp. 3075–3081.
28 Rush, A.M., Chopra, S., Weston, J., 2015. A neural attention model for sentence summarization. In: Conf. Proc. - EMNLP 2015 Conf. Empir. Methods Nat. Lang. Process, pp. 379–389.
29 Zaheer, M.; Guruganesh, G.; Dubey, A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. Big Bird: Transformers for Longer Sequences. *arXiv Prepr. arXiv2007.14062*2020.
30 Olariu, A., 2014. Efficient Online Summarization of Microblogging Streams. In: Proc. 14th Conf. Eur. Chapter Assoc. Comput. Linguist..
31 Cohan, A., Goharian, N., 2015. Scientific article summarization using citation-context and article's discourse structure. In: Conf. Proc. - EMNLP 2015 Conf. Empir. Methods Nat. Lang. Process, pp. 390–400. https://doi.org/10.18653/v1/d15-1045.
32 Li, J., Li, S., 2013. A novel feature-based bayesian model for query focused multi-document summarization. Trans. Assoc. Comput. Linguist. 1, 89–98. https://doi.org/10.1162/tacl_a_00212.
33 Cao, Y., Wan, X., Yao, J., Yu, D., 2020. MultiSumm: towards a unified model for multi-lingual abstractive summarization. In: Proc. AAAI Conf. Artif. Intell., 34, pp. 11–18. https://doi.org/10.1609/aaai.v34i01.5328.
34 Zhu, J., Wang, Q., Wang, Y., Zhou, Y., Zhang, J., Wang, S., Zong, C., 2020. NCLS: neural cross-lingual summarization. In: EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf., pp. 3054–3064. https://doi.org/10.18653/v1/d19-1302.
35 Li, J., Li, H., Zong, C., 2019. Towards personalized review summarization via user-aware sequence network. In: Proc. AAAI Conf. Artif. Intell., 33, pp. 6690–6697. https://doi.org/10.1609/aaai.v33i01.33016690.
36 Rachman, G.H., Leylia Khodra, M., Widyantoro, D.H., 2019. Towards guided summarization of scientific articles: selection of important update sentences. In: ICECOS 2019 - 3rd Int. Conf. Electr. Eng. Comput. Sci. Proceeding, pp. 259–264. https://doi.org/10.1109/ICECOS47637.2019.8984495.
37 Putra, J.W.G., Khodra, M.L., 2017. Automatic title generation in scientific articles for authorship assistance: a summarization approach. J. ICT Res. Appl. 11, 253–267. https://doi.org/10.5614/itbj.ict.res.appl.2017.11.3.3.
38 Nikolov, N.I., Pfeiffer, M., Hahnloser, R.H.R., 2018. Data-driven Summarization of Scientific Articles. In: arXiv Prepr. arXiv1804.08875.
39 Kim, M.; Moirangthem, D.S.; Lee, M. Towards Abstraction from Extraction: Multiple Timescale Gated Recurrent Unit for Summarization. *arXiv Prepr. arXiv1607.00718*2016, doi:10.18653/v1/w16-1608.
40 Chen, J., Zhuge, H., 2019. Automatic generation of related work through summarizing citations. Concurr. Comput. 31. https://doi.org/10.1002/cpe.4261.
41 Collins, E., Augenstein, I., Riedel, S., 2017. A supervised approach to extractive summarisation of scientific papers. In: CoNLL 2017 - 21st Conf. Comput. Nat. Lang. Learn. Proc, pp. 195–205. https://doi.org/10.18653/v1/k17-1021.
42 Altmami, N.I., Menai, M.E.B., 2020. CAST: a cross-article structure theory for multi-article summarization. IEEE Access 8, 100194–100211. https://doi.org/10.1109/ACCESS.2020.2997881.
43 Kieuvongngam, V.; Tan, B.; Niu, Y. Automatic Text Summarization of COVID-19 Medical Research Articles using BERT and GPT-2. *arXiv Prepr. arXiv2006.01997*2020.
44 Sun, X., Zhuge, H., 2019. Automatic Generation of Survey Paper Based on Template Tree. In: 2019 15th Int. Conf. Semant. Knowl. Grids IEEE, pp. 89–96. https://doi.org/10.1109/SKG49510.2019.00023.
45 Winters, T., Mathewson, K.W., 2019. Automatically generating engaging presentation slide decks. In: Int. Conf. Comput. Intell. Music. Sound, Art Des. (Part EvoStar). Cham. Springer.
46 Zhang, Y.; Ding, D.Y.; Qian, T.; Manning, C.D.; Langlotz, C.P. Learning to Summarize Radiology Findings. *arXiv Prepr. arXiv1809.04698*2018.
47 Sotudeh Gharebagh, S.; Goharian, N.; Filice, R. Attend to medical ontologies: content selection for clinical abstractive summarization. *arXiv Prepr. arXiv2005.00163*2020, doi:10.18653/v1/2020.acl-main.172.
48 Lahiri, S., Mihalcea, R., Lai, P.H., 2017. Keyword extraction from emails. Nat. Lang. Eng. 23, 295–317. https://doi.org/10.1017/S1351324916000231.
49 Cao, Z., Wei, F., Li, W., Li, S., 2018. Faithful to the original: fact-aware neural abstractive summarization. In: 32nd AAAI Conf. Artif. Intell. AAAI 2018, pp. 4784–4791.
50 Ghosh, D., 2020. A Sentiment-Based Hotel Review. Emerg. Technol. Model. Graph. 39–44. *Springer, Singapore*.
51 Duan, Y., Jatowt, A., 2019. Across-time comparative summarization of news articles. In: WSDM 2019 - Proc. 12th ACM Int. Conf. Web Search Data Min, pp. 735–743. https://doi.org/10.1145/3289600.3291008.
52 Ghalandari, D.G.; Ifrim, G. Examining the State-of-the-Art in News Timeline Summarization. *arXiv Prepr. arXiv2005.10107*2020.
53 Ali, S.M., Noorian, Z., Bagheri, E., Ding, C., Al-Obeidat, F., 2020. Topic and sentiment aware microblog summarization for twitter. J. Intell. Inf. Syst. 54, 129–156. https://doi.org/10.1007/s10844-018-0521-8.
54 Zhu, C.; Xu, R.; Zeng, M.; Huang, X. End-to-End Abstractive Summarization for Meetings. *arXiv e-prints, arXiv-2004*.2020.
55 Suhara, Y.; Wang, X.; Angelidis, S.; Tan, W.-C. OpinionDigest: A Simple Framework for Opinion Summarization. *arXiv Prepr. arXiv2005.01901*2020.

56 Chowdhury, T., Chakraborty, T., 2019. CQASUMm: Building references for community question answering summarization corpora. In: ACM Int. Conf. Proceeding Ser., pp. 18–26. https://doi.org/10.1145/3297001.3297004.
57 Luo, W., Litman, D., 2015. Summarizing student responses to reflection prompts. In: Conf. Proc. - EMNLP 2015 Conf. Empir. Methods Nat. Lang. Process, pp. 1955–1960. https://doi.org/10.18653/v1/d15-1227.
58 Gorinski, P.J., Lapata, M., 2015. Movie script summarization as graph-based scene extraction. In: NAACL HLT 2015 - 2015 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Conf., pp. 1066–1076. https://doi.org/10.3115/v1/n15-1113.
59 Cheng, G., Xu, D., Qu, Y., 2015. Summarizing entity descriptions for effective and efficient human-centered entity linking. In: WWW 2015 - Proc. 24th Int. Conf. World Wide Web, pp. 184–194. https://doi.org/10.1145/2736277.2741094.
60 Gupta, V. DeepSumm – Deep Code Summaries using Neural Transformer Architecture. *arXiv Prepr. arXiv2004.00998*2020.
61 Genest, P.E., Lapalme, G., Yousfi-Monod, M., 2009. HexTac: the creation of a manual extractive run. In: Proc. Second Text Anal. Conf. TAC 2009, Gaithersburg, Maryland, USA.
62 He, D., Lu, H., Xia, Y., Qin, T., Wang, L., Liu, T.Y., 2017. Decoding with value networks for neural machine translation. In: Adv. Neural Inf. Process. Syst., pp. 178–187. *2017-Decem*.
63 Li, C., Xu, W., Li, S., Gao, S., 2018. Guiding generation for abstractive text summarization based on key information guide network. In: NAACL HLT 2018 - 2018 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., 2, pp. 55–60. https://doi.org/10.18653/v1/n18-2009.
64 Wang, Q., Liu, P., Zhu, Z., Yin, H., Zhang, Q., Zhang, L., 2019. A text abstraction summary model based on BERT word embedding and reinforcement learning. Appl. Sci. 9, 4701. https://doi.org/10.3390/app9214701.
65 Pilault, J., Li, R., Subramanian, S., Pal, C., 2020. On extractive and abstractive neural document summarization with transformer language models. In: Proc. 2020 Conf. Empir. Methods Nat. Lang. Process, pp. 9308–9319.
66 Bahdanau, D., Cho, K.H., Bengio, Y., 2015. Neural machine translation by jointly learning to align and translate. In: 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc, pp. 1–15.
67 Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf, pp. 1724–1734. https://doi.org/10.3115/v1/d14-1179.
68 Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to sequence learning with neural networks. In: Proc. 27th Int. Conf. Neural Inf. Process. Syst., 2, pp. 3104–3112.
69 Elman, J.L., 1990. Finding structure in time. Cogn. Sci. 14, 179–211. https://doi.org/10.1016/0364-0213(90)90002-E.
70 Al-Sabahi, K.; Zuping, Z.; Kang, Y. Bidirectional attentional encoder-decoder model and bidirectional beam search for abstractive summarization. *arXiv Prepr. arXiv1809.06662*2018.
71 Hochreiter, S., 1997. LSTM can solve hard long time lag problems. Adv. Neural Inf. Process. Syst. 473–479.
72 Chandar, S., Sankar, C., Vorontsov, E., Kahou, S.E., Bengio, Y., 2019. Towards non-saturating recurrent units for modelling long-term dependencies. In: Proc. AAAI Conf. Artif. Intell., 33, pp. 3280–3287. https://doi.org/10.1609/aaai.v33i01.33013280.
73 Pascanu, R., Mikolov, T., Bengio, Y., 2013. On the difficulty of training recurrent neural networks. In: Proceedings of the International conference on machine learning. PMLR, pp. 1310–1318.
74 Dangovski, R., Jing, L., Nakov, P., Tatalović, M., Soljačić, M., 2019. Rotational unit of memory: a novel representation unit for RNNs with scalable applications. Trans. Assoc. Comput. Linguist. 7, 121–138. https://doi.org/10.1162/tacl_a_00258.
75 Gidiotis, A., Tsoumakas, G., 2020. A divide-and-conquer approach to the summarization of long documents. In: IEEE/ACM Trans. Audio Speech Lang. Process., 28, pp. 3029–3040. https://doi.org/10.1109/TASLP.2020.3037401.
76 Wang, L., Yao, J., Tao, Y., Zhong, L., Liu, W., Du, Q., 2018. A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. In: IJCAI Int. Jt. Conf. Artif. Intell., pp. 4453–4460. *2018-July*.
77 Zhang, Y., Li, D., Wang, Y., Fang, Y., Xiao, W., 2019. Abstract text summarization with a convolutional seq2seq model. Appl. Sci. 9. https://doi.org/10.3390/app9081665.
78 Shi, T., Keneshloo, Y., Ramakrishnan, N., Reddy, C.K., 2021. Neural abstractive text summarization with sequence-to-sequence models. ACM Trans. Data Sci. 2, 1–37.
79 Lecun, Y., Bottou, L., Bengio, Y., Ha, P., 1998. Gradient-based learning applied to document recognition. In: Proc. IEEE, pp. 2278–2324. https://doi.org/10.1109/5.726791.
80 Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N., 2017. Convolutional sequence to sequence learning. In: 34th Int. Conf. Mach. Learn. ICML 2017, 3, pp. 2029–2042.
81 Dauphin, Y.N., Fan, A., Auli, M., Grangier, D., 2017. Language modeling with gated convolutional networks. In: 34th Int. Conf. Mach. Learn. ICML 2017, 2, pp. 1551–1559.
82 Lei, T., Zhang, Y., Wang, S.I., Dai, H., Artzi, Y., 2018. Simple recurrent units for highly parallelizable recurrence. In: Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018, pp. 4470–4481. https://doi.org/10.18653/v1/d18-1477.
83 Huang, Z.; Xu, P.; Liang, D.; Mishra, A.; Xiang, B. TRANS-BLSTM: Transformer with Bidirectional LSTM for Language Understanding. *arXiv Prepr. arXiv2003.07000*2020.
84 Wang, C.; Li, M.; Smola, A.J. Language Models with Transformers. *arXiv Prepr. arXiv1904.09408*2019.
85 Egonmwan, E., Chali, Y., 2019. Transformer-based model for single documents neural summarization. In: Proc. 3rd Work. Neural Gener. Transl., pp. 70–79. https://doi.org/10.18653/v1/d19-5607.
86 Yao, K., Zhang, L., Du, D., Luo, T., Tao, L., Wu, Y., 2018. Dual encoding for abstractive text summarization. IEEE Trans. Cybern. 1–12. https://doi.org/10.1109/TCYB.2018.2876317.
87 Pasunuru, R., Bansal, M., 2018. Multi-reward reinforced summarization with saliency and entailment. In: NAACL HLT 2018 - 2018 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., 2, pp. 646–653. https://doi.org/10.18653/v1/n18-2102.
88 Lunh, H.P., 1958. The automatic creation of literature abstracts. IBM J. Res. Dev. 2, 159–165.
89 Ko, Y., Seo, J., 2008. An effective sentence-extraction technique using contextual information and statistical approaches for text summarization. Pattern Recognit. Lett. 29, 1366–1371. https://doi.org/10.1016/j.patrec.2008.02.008.
90 Harabagiu, S., Lacatusu, F., 2005. Topic themes for multi-document summarization. In: SIGIR 2005 - Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr, pp. 202–209. https://doi.org/10.1145/1076034.1076071.
91 Parveen, D., Strube, M., 2015. Integrating importance, non-redundancy and coherence in graph-based extractive summarization. IJCAI Int. Jt. Conf. Artif. Intell. 1298–1304. *2015-Janua*.
92 Yoshida, Y., Suzuki, J., Hirao, T., Nagata, M., 2014. Dependency-based discourse parser for single-document summarization. In: EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf, pp. 1834–1839. https://doi.org/10.3115/v1/d14-1196.
93 Kupiec, J.; Pedersen, J.; Chen, F. A Trainable Document summarizer. *SIGIR '95 Proc. 18th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*1995, 1, 68–73, doi: 10.1145/215206.215333.
94 Ren, P., Wei, F., Chen, Z., Ma, J., Zhou, M., 2016. A redundancy-aware sentence regression framework for extractive summarization. In: COLING 2016 - 26th Int. Conf. Comput. Linguist. Proc. COLING 2016 Tech. Pap., pp. 33–43.
95 Zhang, Y., Xia, Y., Liu, Y., Wang, W., 2015. Clustering sentences with density peaks for multi-document summarization. In: NAACL HLT 2015 - 2015 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Conf., pp. 1262–1267. https://doi.org/10.3115/v1/n15-1136.
96 Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V., 2017. Self-critical sequence training for image captioning. In: Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, pp. 1179–1195. https://doi.org/10.1109/CVPR.2017.131. *2017-Janua*.

97 Cohan, A., Dernoncourt, F., Kim, D.S., Bui, T., Kim, S., Chang, W., Goharian, N., 2018. A discourse-aware attention model for abstractive summarization of long documents. In: NAACL HLT 2018 - 2018 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., 2, pp. 615–621. https://doi.org/10.18653/v1/n18-2097.

98 Sharma, E., Li, C., Wang, L.BigPatent, 2019. A large-scale dataset for abstractive and coherent summarization. In: ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf., pp. 2204–2213.

99 Paulus, R.; Xiong, C.; Socher, R. A deep reinforced model for abstractive summarization. *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*2017, 1–12.

100 Celikyilmaz, A., Bosselut, A., He, X., Choi, Y., 2018. Deep communicating agents for abstractive summarization. In: NAACL HLT 2018 - 2018 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., 1, pp. 1662–1675. https://doi.org/10.18653/v1/n18-1150.

101 Xu, K., Ba, J.L., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R.S., Bengio, Y., 2015. Show, attend and tell: neural image caption generation with visual attention. In: Int. Conf. Mach. Learn., pp. 2048–2057.

102 Shang, L., Lu, Z., Li, H., 2015. Neural responding machine for short-text conversation. In: ACL-IJCNLP 2015 - 53rd Annu. Meet. Assoc. Comput. Linguist. 7th Int. Jt. Conf. Nat. Lang. Process. Asian Fed. Nat. Lang. Process. Proc. Conf., 1, pp. 1577–1586. https://doi.org/10.3115/v1/p15-1152.

103 Luong, M.T., Pham, H., Manning, C.D., 2015. Effective approaches to attention-based neural machine translation. In: Conf. Proc. - EMNLP 2015 Conf. Empir. Methods Nat. Lang. Process, pp. 1412–1421. https://doi.org/10.18653/v1/d15-1166.

104 Li, J., Luong, M.T., Jurafsky, D., 2015. A hierarchical neural Autoencoder for paragraphs and documents. In: ACL-IJCNLP 2015 - 53rd Annu. Meet. Assoc. Comput. Linguist. 7th Int. Jt. Conf. Nat. Lang. Process. Asian Fed. Nat. Lang. Process. Proc. Conf., 1, pp. 1106–1115. https://doi.org/10.3115/v1/p15-1107.

105 Miao, Y., Blunsom, P., 2016. Language as a latent variable: discrete generative models for sentence compression. In: EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc, pp. 319–328. https://doi.org/10.18653/v1/d16-1031.

106 Boutkan, F.; Ranzijn, J.; Rau, D.; van der Wel, E. Point-less: More Abstractive Summarization with Pointer-Generator Networks. *arXiv Prepr. arXiv1905.019752*019.

107 Wiseman, S., Rush, A.M., 2016. Sequence-to-sequence learning as beam-search optimization. In: EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc, pp. 1296–1306. https://doi.org/10.18653/v1/d16-1137.

108 Eyal, M., Baumel, T., Elhadad, M., 2019. Question answering as an automatic evaluation metric for news article summarization. In: NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., 1, pp. 3938–3948. https://doi.org/10.18653/v1/n19-1395.

109 Vijayakumar, A.K.; Cogswell, M.; Selvaraju, R.R.; Sun, Q.; Lee, S.; Crandall, D.; Batra, D. Diverse beam search: decoding diverse solutions from neural sequence models. *arXiv Prepr. arXiv1610.02424*2016.

110 Han, X.W., Zheng, H.T., Chen, J.Y., Zhao, C.Z., 2019. Diverse decoding for abstractive document summarization. Appl. Sci. 9, 1–15. https://doi.org/10.3390/app9030386.

111 Cibils, A.; Musat, C.; Hossman, A.; Baeriswyl, M. Diverse beam search for increased novelty in abstractive summarization. *arXiv Prepr. arXiv1802.014572*018.

112 Hu, B., Chen, Q., Zhu, F., 2015. LCSTS: a large scale Chinese short text summarization dataset. In: Conf. Proc. - EMNLP 2015 Conf. Empir. Methods Nat. Lang. Process, pp. 1967–1972. https://doi.org/10.18653/v1/d15-1229.

113 Lopyrev, K. Generating News Headlines with Recurrent Neural Networks. *arXiv Prepr. arXiv1512.017122*015.

114 Chopra, S., Auli, M., Rush, A.M., 2016. Abstractive sentence summarization with attentive recurrent neural networks. In: 2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL HLT 2016 - Proc. Conf, pp. 93–98. https://doi.org/10.18653/v1/n16-1012.

115 Takase, S., Suzuki, J., Okazaki, N., Hirao, T., Nagata, M., 2016. Neural headline generation on abstract meaning representation. In: EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc, pp. 1054–1059. https://doi.org/10.18653/v1/d16-1112.

116 Vinyals, O.; Fortunato, M.; Jaitly, N. Pointer networks. *arXiv Prepr. arXiv1506.031342*015.

117 Luong, M.T.; Sutskever, I.; Le, Q. V.; Vinyals, O.; Zaremba, W. Addressing the rare word problem in neural machine translation. *ACL-IJCNLP 2015 - 53rd Annu. Meet. Assoc. Comput. Linguist. 7th Int. Jt. Conf. Nat. Lang. Process. Asian Fed. Nat. Lang. Process. Proc. Conf.*2015, 1, 11–19, doi:10.3115/v1/p15-1002.

118 Tu, Z., Lu, Z., Yang, L., Liu, X., Li, H., 2016. Modeling coverage for neural machine translation. In: 54th Annu. Meet. Assoc. Comput. Linguist. ACL 2016 - Long Pap., 1, pp. 76–85. https://doi.org/10.18653/v1/p16-1008.

119 Mi, H., Sankaran, B., Wang, Z., Ittycheriah, A., 2016. Coverage embedding models for neural machine translation. In: EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc, pp. 955–960. https://doi.org/10.18653/v1/d16-1096.

120 Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., 2016. Distraction-based neural networks for modeling documents. IJCAI Int. Jt. Conf. Artif. Intell. 2754–2760. *2016-Janua*.

121 Saito, I.; Nishida, K.; Nishida, K.; Tomita, J. Abstractive summarization with combination of pre-trained sequence-to-sequence and saliency models. *arXiv Prepr. arXiv2003.130282*020.

122 Dou, Z.Y.; Liu, P.; Hayashi, H.; Jiang, Z.; Neubig, G. GSum: a general framework for guided neural abstractive summarization. *arXiv Prepr. arXiv2010.080142*020.

123 Kikuchi, Y., Neubig, G., Sasano, R., Takamura, H., Okumura, M., 2016. Controlling output length in neural encoder-decoders. In: EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc, pp. 1328–1338. https://doi.org/10.18653/v1/d16-1140.

124 Liu, Y., Luo, Z., Zhu, K.Q., 2020. Controlling length in abstractive summarization using a convolutional neural network. In: Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018, pp. 4110–4119. https://doi.org/10.18653/v1/d18-1444.

125 Fan, A.; Grangier, D.; Auli, M. Controllable abstractive summarization. *arXiv Prepr. arXiv1711.052172*017, doi:10.18653/v1/w18-2706.

126 Liu, P.J.; Saleh, M.; Pot, E.; Goodrich, B.; Sepassi, R.; Kaiser, Ł.; Shazeer, N. Generating wikipedia by summarizing long sequences. *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*2018, 1–18.

127 Zhu, C.; Hinthorn, W.; Xu, R.; Zeng, Q.; Zeng, M.; Huang, X.; Jiang, M. Boosting Factual Correctness of Abstractive Summarization with Knowledge Graph. *arXiv Prepr. arXiv2003.086122*020.

128 Jin, H., Wang, T., Wan, X., 2020. SemSUM: semantic dependency guided neural abstractive summarization. In: Proc. AAAI Conf. Artif. Intell., 34, pp. 8026–8033. https://doi.org/10.1609/aaai.v34i05.6312.

129 Cao, Z., Li, W., Wei, F., Li, S., 2018. Retrieve, rerank and rewrite: soft template based neural summarization. In: Proc. Annu. Meet. Assoc. Comput. Linguist. (ACL)..

130 Zhou, L., Hu, W., Zhang, J., Zong, C., 2017. Neural system combination for machine translation. In: ACL 2017 - 55th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap., 2, pp. 378–384. https://doi.org/10.18653/v1/P17-2060.

131 Mizumoto, T., Matsumoto, Y., 2016. Discriminative reranking for grammatical error correction with statistical machine translation. In: 2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL HLT 2016 - Proc. Conf., pp. 1133–1138. https://doi.org/10.18653/v1/n16-1133.

132 Liu, Y.; Dou, Z.-Y.; Liu, P. RefSum: Refactoring Neural Summarization. 2021.

133 Liu, Y.; Liu, P. SimCLS: A Simple Framework for Contrastive Learning of Abstractive Summarization. 2021.

134 Zeiler, M.D. ADADELTA: AN ADAPTIVE LEARNING RATE METHOD. *arXiv Prepr. arXiv1212.5701*2012.

135 Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. 12, 2121–2159.

136 Mousavi, S.S., Schukat, M., Howley, E., 2016. Deep reinforcement learning: an overview. In: Proc. SAI Intell. Syst. Conf. Springer, Cham, pp. 426–440. https://doi.org/10.1007/978-3-319-56991-8_32.

137 Böhm, F., Gao, Y., Meyer, C.M., Shapira, O., Dagan, I., Gurevych, I., 2020. Better rewards yield better summaries: Learning to summarise without references. In: EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf., pp. 3110–3120. https://doi.org/10.18653/v1/d19-1307.

138 Ranzato, M., Chopra, S., Auli, M., Zaremba, W., 2016. Sequence level training with recurrent neural networks. In: 4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc., pp. 1–16.

139 Li, P.; Bing, L.; Lam, W. Actor-critic based training framework for abstractive summarization. *arXiv Prepr. arXiv1803.110702*018.

140 Keneshloo, Y., Ramakrishnan, N., Reddy, C.K., 2019. Deep transfer reinforcement learning for text summarization. In: SIAM Int. Conf. Data Mining, SDM 2019, pp. 675–683. https://doi.org/10.1137/1.9781611975673.76.

141 Keneshloo, Y., Shi, T., Ramakrishnan, N., Reddy, C.K., 2020. Deep reinforcement learning for sequence-to-sequence models. IEEE Trans. Neural Networks Learn. Syst. 31, 2469–2489. https://doi.org/10.1109/TNNLS.2019.2929141.
142 Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N., 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In: Adv. Neural Inf. Process. Syst., pp. 1171–1179. *2015-Janua*.
143 Su, J., Xu, J., Qiu, X., Huang, X., 2018. Incorporating discriminator in sentence generation: A Gibbs sampling method. In: 32nd AAAI Conf. Artif. Intell. AAAI 2018, pp. 5496–5503.
144 Greensmith, E., Bartlett, P.L., Baxter, J., 2004. Variance reduction techniques for gradient estimates in reinforcement learning. J. Mach. Learn. Res. 5, 1471–1530.
145 Chakraborty, S.; Li, X.; Chakraborty, S. A more abstractive summarization model. *arXiv Prepr. arXiv2002.109592*020.
146 Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15.1 15, 1929–1958.
147 Song, K., Wang, B., Feng, Z., Liu, R., Liu, F., 2020. Controlling the amount of verbatim copying in abstractive summarization. In: Proc. AAAI Conf. Artif. Intell., 34, pp. 8902–8909. https://doi.org/10.1609/aaai.v34i05.6420.
148 Moroshko, E.; Feigenblat, G.; Roitman, H.; Konopnicki, D. An editorial network for enhanced document summarization. *arXiv Prepr. arXiv1902.103602*019, doi: 10.18653/v1/d19-5407.
149 Bae, S.; Kim, T.; Kim, J.; Lee, S. goo Summary level training of sentence rewriting for abstractive summarization. *arXiv Prepr. arXiv1909.087522*019, doi: 10.18653/v1/d19-5402.
150 Scialom, T., Lamprier, S., Piwowarski, B., Staiano, J., 2020. Answers unite! Unsupervised metrics for reinforced summarization models. In: EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf., pp. 3246–3256. https://doi.org/10.18653/v1/d19-1320.
151 Kryściński, W.; McCann, B.; Xiong, C.; Socher, R. Evaluating the factual consistency of abstractive text summarization. *arXiv Prepr. arXiv1910.128402*019.
152 Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q., 2021. A comprehensive survey on transfer learning. In: Proc. IEEE, 109, pp. 43–76. https://doi.org/10.1109/JPROC.2020.3004555.
153 Kitaev, N.; Kaiser, Ł.; Levskaya, A. Reformer: The Efficient Transformer. *arXiv Prepr. arXiv2001.044512*020.
154 Child, R.; Gray, S.; Radford, A.; Sutskever, I. Generating Long Sequences with Sparse Transformers. *arXiv Prepr. arXiv1904.105092*019.
155 Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R., 2020. Transformer-XL: attentive language models beyond a fixed-length context. In: ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf., pp. 2978–2988. https://doi.org/10.18653/v1/p19-1285.
156 Huang, L., Cao, S., Parulian, N., Ji, H., Wang, L., 2021. Efficient attentions for long document summarization. In: arXiv Prepr. arXiv2104.02112.
157 Tay, Y.; Dehghani, M.; Bahri, D.; Metzler, D. Efficient Transformers: A Survey. *arXiv Prepr. arXiv2009.067322*020.
158 Taylor, W.L., 1953. Cloze procedure": a new tool for measuring readability. Journal. Q. 30, 415–433.
159 Yan, Y., Qi, W., Gong, Y., Liu, D., Duan, N., Chen, J., Zhang, R., Zhou, M., 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. In: arXiv Prepr. arXiv2001.04063. https://doi.org/10.18653/v1/2020.findings-emnlp.217.
160 Xiao, D., Zhang, H., Li, Y., Sun, Y., Tian, H., Wu, H., Wang, H.ERNIE-GEN, 2020. An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. In: IJCAI Int. Jt. Conf. Artif. Intell., pp. 3997–4003. https://doi.org/10.24963/ijcai.2020/553. *2021-Janua*.
161 Qiu, X.P., Sun, T.X., Xu, Y.G., Shao, Y.F., Dai, N., Huang, X.J., 2020. Pre-trained models for natural language processing: a survey. Sci. China Technol. Sci. 63, 1872–1897. https://doi.org/10.1007/s11431-020-1647-3.
162 Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y., 2019. MASS: masked sequence to sequence pre-training for language generation. In: 36th Int. Conf. Mach. Learn. ICML 2019, pp. 10384–10394. *2019-June*.
163 Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; Hon, H.-W. Unified Language Model Pre-training for Natural Language Understanding and Generation. *arXiv Prepr. arXiv1905.031972*019.
164 Liu, Y., Lapata, M., 2019. Text summarization with pretrained encoders. In: EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf, pp. 3730–3740. https://doi.org/10.18653/v1/d19-1387.
165 Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations. In: NAACL HLT 2018 - 2018 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., 1, pp. 2227–2237. https://doi.org/10.18653/v1/n18-1202.
166 Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.XLNet, 2019. Generalized autoregressive pretraining for language understanding. Adv. Neural Inf. Process. Syst. 32, 1–18.
167 Guan, J., Huang, F., Zhao, Z., Zhu, X., Huang, M., 2020. A knowledge-enhanced pretraining model for commonsense story generation. Trans. Assoc. Comput. Linguist. 8, 93–108. https://doi.org/10.1162/tacl_a_00302.
168 Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R., 2019. Glue: a multi-task benchmark and analysis platform for natural language understanding. In: 7th Int. Conf. Learn. Represent. ICLR 2019, pp. 1–20.
169 Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P., 2016. SQuad: 100,000+ questions for machine comprehension of text. In: EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc., pp. 2383–2392.
170 Bao, H., Dong, L., Wei, F., Wang, W., Yang, N., Liu, X., Wang, Y., Piao, S., Gao, J., Zhou, M., et al., 2020. Unilmv2: pseudo-masked language models for unified language model pre-Training. In: 37th Int. Conf. Mach. Learn. ICML 2020, pp. 619–629. *PartF16814*.
171 Zou, Y., Zhang, X., Lu, W., Wei, F., Zhou, M., 2020. Pre-training for abstractive document summarization by reinstating source text. In: Proc. 2020 Conf. Empir. Methods Nat. Lang. Process. (EMNLP), pp. 3646–3660. https://doi.org/10.18653/v1/2020.emnlp-main.297.
172 Zhong, M.; Liu, P.; Chen, Y.; Wang, D.; Qiu, X.; Huang, X. Extractive summarization as text matching. *arXiv Prepr. arXiv2004.087952*020, doi:10.18653/v1/2020.acl-main.552.
173 Zhang, X., Wei, F., Zhou, M.Hibert, 2020. Document level pre-training of hierarchical bidirectional transformers for document summarization. In: ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf., pp. 5059–5069. https://doi.org/10.18653/v1/p19-1499.
174 Xu, J., Gan, Z., Cheng, Y., Liu, J., 2020. Discourse-aware neural extractive text summarization. In: arXiv Prepr. arXiv1910.14142. https://doi.org/10.18653/v1/2020.acl-main.451.
175 Zhong, M., Liu, P., Wang, D., Qiu, X., Huang, X., 2020. Searching for effective neural extractive summarization: What works and what's next.. In: ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf., pp. 1049–1058. https://doi.org/10.18653/v1/p19-1100.
176 Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., Robinson, T., 2014. One billion word benchmark for measuring progress in statistical language modeling. In: Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, pp. 2635–2639.
177 Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., Fidler, S., 2015. Zhu_Aligning_Books_and_ICCV_2015_paper. In: Proc. ICCV, pp. 19–27.
178 Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv Prepr. arXiv1907.116922*019.
179 Trinh, T.H.; Le, Q. V. A simple method for commonsense reasoning. *arXiv Prepr. arXiv1806.028472*018.
180 Xiao, W.; Carenini, G. Systematically Exploring Redundancy Reduction in Summarizing Long Documents. *arXiv Prepr. arXiv2012.000522*020.
181 Zhang, H., Cai, J., Xu, J., Wang, J., 2019. Pretraining-based natural language generation for text summarization. In: CoNLL 2019 - 23rd Conf. Comput. Nat. Lang. Learn. Proc. Conf, pp. 789–797. https://doi.org/10.18653/v1/k19-1074.
182 Khandelwal, U.; Clark, K.; Jurafsky, D.; Kaiser, L. Sample efficient text summarization using a single pre-trained transformer. *arXiv Prepr. arXiv1905.088362*019.
183 Hoang, A.; Bosselut, A.; Celikyilmaz, A.; Choi, Y. Efficient Adaptation of Pretrained Transformers for Abstractive Summarization. *arXiv Prepr. arXiv1906.001382*019.
184 Edunov, S., Baevski, A., Auli, M., 2019. Pre-trained language model representations for language generation. In: NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., 1, pp. 4052–4059. https://doi.org/10.18653/v1/n19-1409.
185 Graff, D., Kong, J., Chen, K., Maeda, K., 2003. English gigaword. Linguist. Data Consortium, Philadelphia 4, 34.

186 Napoles, C., Gormley, M., Durme, B.Van Annotated Gigaword, 2012. In: Proc. Jt. Work. Autom. Knowl. Base Constr. Web-scale Knowl. Extr. (AKBC-WEKEX '12), pp. 95–100.
187 Hermann, K.M., Kočiský, T., Grefenstette, E., Espeholt, L., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P., 2015. Teaching machines to read and comprehend NIPS 2015. In: Proc. 28th Int. Conf. Neural Inf. Process. Syst, pp. 1693–1701.
188 Narayan, S., Cohen, S.B., Lapata, M., 2018. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In: Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018, pp. 1797–1807.
189 Kornilova, A.; Eidelman, V. BillSum: A Corpus for Automatic Summarization of US Legislation. *arXiv Prepr. arXiv1910.00523*2019, doi:10.18653/v1/d19-5406.
190 Koupaee, M.; Wang, W.Y. WikiHow: A Large Scale Text Summarization Dataset. *arXiv Prepr. arXiv1810.09305*2018.
191 Kim, B.; Kim, H.; Kim, G. Abstractive Summarization of Reddit Posts with Multi-level Memory Networks. *arXiv Prepr. arXiv1811.00783*2018.
192 Scialom, T.; Dray, P.-A.; Lamprier, S.; Piwowarski, B.; Staiano, J. MLSUM: The Multilingual Summarization Corpus. *arXiv Prepr. arXiv2004.14900*2020.
193 Grusky, M., Naaman, M., Artzi, Y., 2018. Newsroom: a dataset of 1.3 million summaries with diverse extractive strategies. In: NAACL HLT 2018 - 2018 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., 1, pp. 708–719. https://doi.org/10.18653/v1/n18-1065.
194 Denkowski, M., Lavie, A., 2014. Meteor universal: language specific translation evaluation for any target language. Proc. ninth Work. Stat. Mach. Transl. 376–380. https://doi.org/10.3115/v1/w14-3348.
195 Papineni, K., Roukos, S., Ward, T., Zhu, W.-J., 2002. BLEU: a method for automatic evaluation of machine translation. In: Proc. 40th Annu. Meet. Assoc. Comput. Linguist., pp. 311–318. https://doi.org/10.1002/andp.19223712302.
196 Krantz, J.; Kalita, J. Abstractive Summarization Using Attentive Neural Techniques. *arXiv Prepr. arXiv1810.08838*2018.
197 Zhao, W., Peyrard, M., Liu, F., Gao, Y., Meyer, C.M., Eger, S., 2020. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. In: EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf., pp. 563–578. https://doi.org/10.18653/v1/d19-1053.
198 Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating Text Generation with BERT. *arXiv Prepr. arXiv1904.09675*2019.
199 Rubner, Y., Tomasi, C., Guibas, L.J., 2000. The earth mover's distance as a metric for image retrieval. Int. J. Comput. Vis. 40, 99–121.
200 Sankaran, B.; Mi, H.; Al-Onaizan, Y.; Ittycheriah, A. Temporal Attention Model for Neural Machine Translation. *arXiv Prepr. arXiv1608.02927.*2016.
201 Jean, S., Cho, K., Memisevic, R., Bengio, Y., 2015. On using very large target vocabulary for neural machine translation. In: ACL-IJCNLP 2015 - 53rd Annu. Meet. Assoc. Comput. Linguist. 7th Int. Jt. Conf. Nat. Lang. Process. Asian Fed. Nat. Lang. Process. Proc. Conf., 1, pp. 1–10. https://doi.org/10.3115/v1/p15-1001.
202 Doetsch, P., Zeyer, A., Ney, H., 2016. Bidirectional decoder networks for attention-based end-to-end offline handwriting recognition. In: Proc. Int. Conf. Front. Handwrit. Recognition, ICFHR, 0, pp. 361–366. https://doi.org/10.1109/ICFHR.2016.0074.
203 Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv Prepr. arXiv1609.08144*2016.
204 Gu, J., Cho, K., Li, V.O.K., 2017. Trainable greedy decoding for neural machine translation. In: EMNLP 2017 - Conf. Empir. Methods Nat. Lang. Process. Proc, pp. 1968–1978. https://doi.org/10.18653/v1/d17-1210.
205 Gu, J., Neubig, G., Cho, K., Li, V.O.K., 2017. Learning to translate in real-time with neural machine translation. In: 15th Conf. Eur. Chapter Assoc. Comput. Linguist. EACL 2017 - Proc. Conf., 1, pp. 1053–1062. https://doi.org/10.18653/v1/e17-1099.