Advancements in Text Summarization Using LSTM and Transformer-Based Models: A Comparative Review

M. Rukhsah
PG Scholar, Department of MTI PJJ, Universitas Amikom Yogyakarta
Sleman, Yogyakarta, Indonesia.

Article Info

Article History:

Received Sep 21, 2025
Revised Oct 19, 2025
Accepted Nov 23, 2025

Keywords:

Long Short-Term Memory (LSTM)
Transformer Architecture
Deep Learning
Natural Language Processing (NLP)
Sequence-to-Sequence Models
Pretrained Language Models

ABSTRACT

Text summarization has proved to be a vital part of NLP as it has turned long texts short and yet with the primary components. This survey investigates the development of text summarization algorithms, putting a particular focus on the Long Short-Term Memory (LSTM) and Transformer-based summarization. Initially, LSTM models had significantly advanced the way neural text summarization was conducted by carefully managing sequential connections through encoder-decoder architectures and, on numerous occasions, through attention and pointer-generator networks. But why they suffer in distant connections and fail to employ parallel computing is the question that gave rise to Transformer-based models, which introduce self-attention and increase the summarization performance. Recently, systems based on BERTSUM, PEGASUS, BART and T5 have set new scores in both summarization tasks. This paper critically compares these architectures in terms of their model, the complexity of the training algorithm involved, the metrics to be used in evaluating and the areas of application they operate in. Besides that, the paper also highlights highly relevant dataset and trends in progress such as the absorption of ready-learned language models, adapting to new domains and encountering new evaluation challenges. The necessity to cope with the concerns like consistency, summarizing low-resource texts and scalability guides the choice of the future research direction.

Corresponding Author:

M. Rukhsah,
PG Scholar, Department of MTI PJJ,
Universitas Amikom Yogyakarta, Indonesia.
E-mail: ucousman@gmail.com

1. INTRODUCTION

Text summarization aims to apply NLP in generating short and informative summaries out of a written text that is significantly longer [1]. Overall, text summarization can be divided into abstractive or extractive summarization. Extractive only picks and joins primary sentences of the source, whereas abstractive generates new sentences in a simpler language that conveys the main ideas. Extractive methods are easier to develop, whereas abstractive methods are similar to the ways people summarize, but more challenging, language- and meaning-wise [3].

Text summarization is gaining importance in news collection, briefing legal filings, compression of medical reports, analysis of customer surveys and summarization of research articles. Now that we have access to so much information, summarizing systems have become highly useful in making things more accessible, enhancing decision-making and saving us time.

In the past, early summarization has been achieved by using techniques like rules, statistics like TF-IDF and LSA and graph-like techniques like TextRank. But they never managed to cover complicated relationships between meanings and grammar of language.

A significant shift in the field was introduced by deep learning. They were improved with the help of models based on RNN and primarily on LSTM because of their capability to discover the pattern in sequence and perform abstractive translations with the help of the encoder-decoder approach. The use of LSTM-based architectures such as Seq2Seq with attention and pointer-generator networks became the standard architecture of early neural summarizers. Their advancements are curtailed by their problems in managing dependencies with large scope and parallelization problems.

Due to these drawbacks, the seminal paper of [2]. "Attention Is All You Need" presented Transformer-based models that initiated a new summit in the research of summarization. Eliminate recurrence and increased ability to be parallelized, scaled up and retain context using self-attention in Transformers. Abstractive summarization has largely improved owing to new records by BERTSUM, BART, PEGASUS and T5.

In this paper, the difference between LSTM and Transformer-based models in text summarization will be examined. We discuss the transformations of summarization techniques with the shift to LSTM based models and now with the Transformer models, we discuss what they are constructed of, and how they compare to each other by model complexity, performance, speed and actual usage. Further, the article discusses how research is currently utilising large language models (LLMs), factual validity review and summarisation with minimal resources. In our last consideration, we examine the issues that still remain outstanding, and ways in which one may make summarization systems more accurate, quick and reliable.

2. BACKGROUND

Summarization of text which is a significant task in NLP, involves extracting key ideas of a source document and reducing its volume, while still preserving its sense. Since early beginnings of using simple techniques depending on word frequency, summarization is currently done using neural networks which are capable of producing summaries that are readable by humans. Extractive summarization aims at selecting and keeping the significant sentences; abstractive summarization has the propensity to rephrase the information to make it similar to the summary produced by humans.

2.1 Fundamentals of Text Summarization

At its core, summarization hinges on several foundational concepts:

- **Relevance Detection**: Identifying the most informative or representative parts of the input.
- **Compression**: Reducing the length while preserving critical information.
- **Paraphrasing**: Particularly for abstractive models, rewording ideas without loss of meaning.
- **Context Awareness**: Understanding document structure, narrative flow, and discourse relationships.

These processes demand not just syntactic parsing but deep semantic representation—capturing not only what is said but *what it means*, and how it connects to the broader context. This makes summarization a highly complex task, more than mere sentence ranking or truncation.

2.2 Key Challenges in Text Summarization

Despite tremendous progress, text summarization remains riddled with critical challenges:

Semantic Understanding

People who abstractly summarize, resolve to make concrete decisions and think, ought to concern themselves with problems that are in legal or medical domains [8]. Indicatively, the relationship between variables is still a pending matter in addition to deduction of unstated facts.

Context Retention

Themes can be shared or connected among the different paragraphs of a long document. Many models struggle with long-range connections (as occur in legal cases or scientific papers) since they are designed using shallow representations which tend to forget valuable bridges [9].

Coherence and Fluency

Created summaries must be sensible and readable. The summaries that are generated by certain models may be grammatically correct but nonsense, primarily when they are presented with something they have not been trained on as input [10].

Factual Consistency

Errors occur in the form of abstractive summarizers, which are large-model-based, generating material that was not even in the original text [11].

Domain Adaptation

And when you apply the models trained with news data to technical, biomedical or legal writing, they will not summarize well.

Evaluation Bottlenecks

ROUGE or BLEU-based measure the similarity of words between the generated summary and the target document, which is not optimal in the evaluation of the quality of abstractive summaries. The inventions in ensuring evaluation techniques are more precise and conscious to the surroundings are proceeding [12].

3. LSTM-BASED ARCHITECTURES

The success of neural sequence modeling Early breakthroughs in neural sequence modeling came through the application of Long Short-Term Memory (LSTM) networks to text summarization. Since LSTM is a particular variant of recurrent neural network (RNN), it was constructed to address the issue of vanishing gradient in ordinary RNNs that allows it to engage in learning distant data points. Due to this fact, LSTM is applied rather frequently to the tasks of machine translation, answering questions and selection of the most significant sentences of a text. Important LSTM design considering the task of summarization is examined in the following sections in detail.

3.1 Standard LSTM

A vanilla LSTM model has memory cells that have internal gates, which are input gate, forget gate and output gate, which control the information flow through the network. Through these gates, the LSTM can decide what information to keep and what to discard as it processes data sets that are sequentially ordered. Normally, the ordinary LSTM is used in an encoder-decoder framework in performing text summarization. The encoder reviews the input partition by partition and encodes its meaning in a fixed vector that represents the meaning of the entire text. Then the decoder, with the help of the context, generates a summary word by word. Although this method significantly enhanced the compression of long or complicated text into a summary, it had a problem in recording all the details as the fixed-length context restricted the data volume that could be sent by the encoder to the decoder. Figure 1 represents the LSTM-based architectures for text summarization.

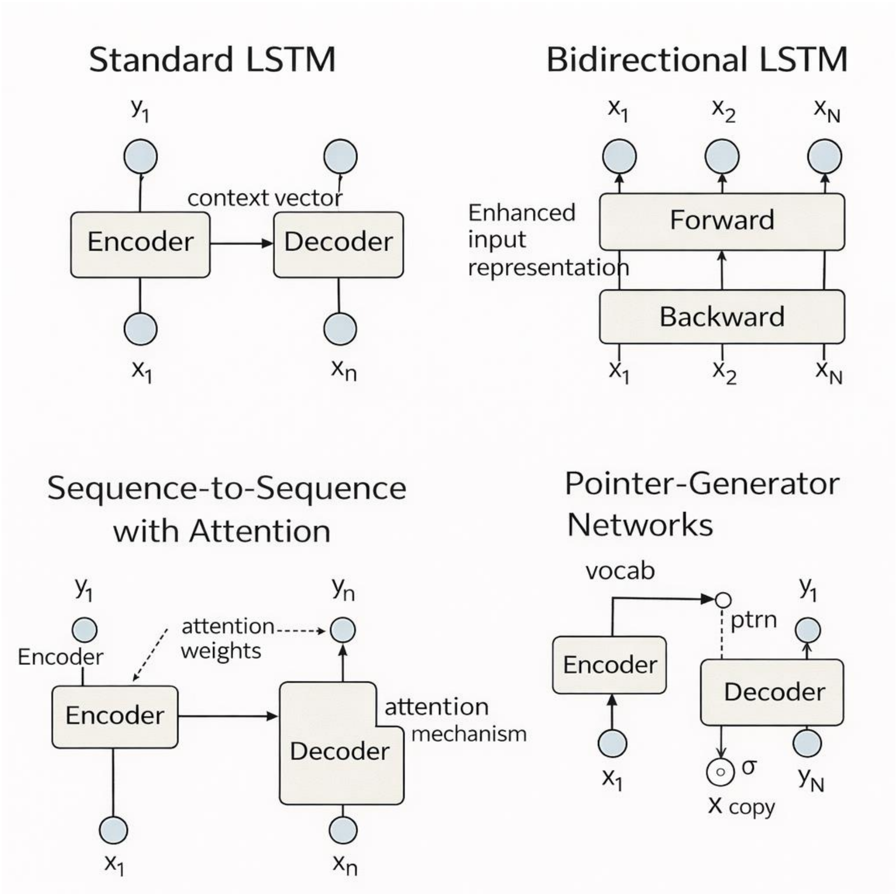

Figure 1. LSTM-Based Architectures for Text Summarization

3.2 Bidirectional LSTM (BiLSTM)

In order to address the problems with standard LSTM models, BiLSTM networks examine and compute the input sequence forward and backward: two directions in total. The ability to see surrounding words on either side assists the model to figure out the significant associations between language structures and semantics. In listening to summarization, BiLSTM enhances the richness of the representation resulting to structured and contextualized summaries. However, the inefficiency of its computation still exists due to the necessity in processing data in sequential order in BiLSTM.

3.3 Sequence-to-Sequence with Attention

Machine translation first used the Sequence-to-Sequence method and it rapidly became the default for LSTM in text summarization. In this design, you will find:

- The source text is made into a hidden representation by the encoder LSTM.
- Once the input is processed, the decoder LSTM picks out the next word for the summary.

To address the problems of fixed-length vectors, the attention mechanism of Bahdanau and Luong was introduced, allowing the decoder to attend to specific sections of the source text at different time steps. This mechanism allows summaries generated by the model to be more accurate and easy to read by assigning special importance to the words relevant in the sentence.

3.4 Pointer-Generator Networks

A significant advancement of LSTM-based summarization is Pointer-Generator Networks as it integrates copying and generation. Depending on a hybrid system, the model will be able to:

- Select words among a pool of words.

To insert the copy of the word that is in the source, you can move the pointer that is connected to it. It is obligatory to use this ability because it preserves the important meaning of original text in most of the situations when it is difficult to find OOV or technical terms. Normally, it has a system to monitor the level of focus on each section of the document and edit out the excessively repetitive portions of the summary.

3.5 Advantages and Limitations

The advantages and limitations of the key aspects are depicted in Table 1.

Table 1. Advantages and Limitations of the Aspect

Aspect	**Advantages**	**Limitations**
Sequential Dependency	Effectively models temporal and syntactic structures in language	Struggles with long-range dependencies, especially in long documents
Contextual Understanding	Enhanced with BiLSTM and attention for richer semantic understanding	Fixed context vectors in vanilla models limit expressiveness
Vocabulary Handling	Pointer-generator enables handling of rare and OOV words	Generation quality suffers in domains requiring broader context
Repetition Control	Coverage mechanism helps mitigate redundancy	May still produce disfluent or redundant outputs in complex documents
Efficiency	Suitable for medium-length text summarization tasks	Inefficient due to sequential processing; longer training and inference times

4. TRANSFORMER-BASED ARCHITECTURES

Transformer architectures easily capture global dependencies and that is why it is the primary structure of modern NLP. In the landmark publication “Attention Is All You Need” by [2] the Transformer architecture was introduced, which used multi-head self-attention and position-wise feed-forward layers to solve the recurrence problem in RNNs. Due to these developments, models were able to process input simultaneously, read each input token and achieve the best performance in various language-related tasks, such as in text summarization.

Here we examine several state-of-the-art Transformer-based models on summarization, investigating what advantages they offer, how they are preproduced and whether they are applicable to both extractive and abstractive types of summarization.

4.1 Vanilla Transformer

The original Transformer architecture was based on an encoder-decoder approach wherein attention and feed-forward layers are stacked vertically. The Transformer is trained by encoding input text into context embeddings and then decoder generates the output by attending to the previous outputs and encoder outputs. The Transformer was trained on particular datasets in the first place to be created. Neither does it employ large sets of data, usually unlabelled, and therefore cannot be applied to summarization with further training. Text summarization architecture is shown in Figure 2.

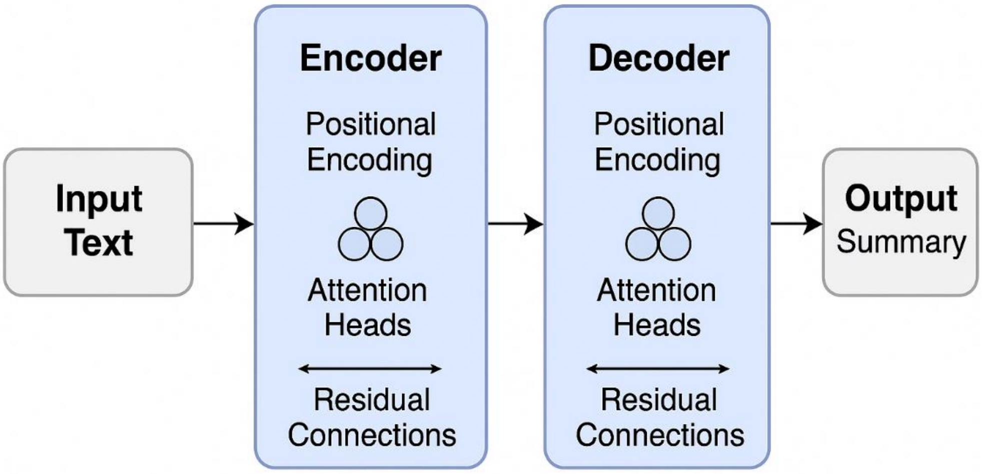

Figure 2. Transformer Architecture for Text Summarization

4.2 BERTSUM

[4] Fine-tuned the BERT encoder with a classifier on the top to generate BERTSUM. Special [CLS] tokens are inserted at the start of every sentence and the representations of these tokens are further utilised to determine the importance of each sentence.

BERTSUM applies the abilities of BERT to process language and examine in both directions to identify important sentences. It sets new state-of-the-art results on the CNN/DailyMail datasets than the current extractive systems. Despite that, it is still limited in its ability to delete sentences but not create new phrases accordingly it is not formulated to work with abstractive summarization.

4.3 PEGASUS

PEGASUS [5] introduces an alternate form of pretraining by bypassing Gap-Sentence a special kind of sentence generated by the model. During pretraining, the model does not choose words to mask, but it covers entire sentences and learns to reconstruct them, as would be done in making a summary.

Due to this, the system is not bad at summarizing despite the paucity of extra information in the scenario where resources are limited. Since PEGASUS is pretrained on what ought to be summary-worthy, it attains strong results on several news, scientific and medical summary datasets.

4.4 BART

The authors suggest an autoencoder named BART [6] based on Transformer architecture. The model concatenates a bidirectional encoder, such as BERT with an autoregressive decoder that is similar to GPT. BART starts learning by altering the original reference phrases (e.g. reorder them or remove some parts) and training to reproduce them.

The consequence of such a design is that BART is particularly well-adapted to abstractive summarization as it turns more robust and fluent. It is recommended that you will see it consistently among the top models in CNN/DailyMail, XSum and MultiNews, with clear summaries and on-topic results.

4.5 T5

[7] Proposed the Text-to-Text Transfer Transformer (T5) that casts all natural language processing (NLP) tasks including summarization, as a single problem of transforming one piece of text into another text. As an illustration, the process of summarization is presented as: summarize: [input text].

T5 was trained on the C4 dataset, in span-corruption, where sequences of words are replaced by sentinel tokens. Due to this framework, T5 can deal with various jobs with ease, which is represented in Figure 3. It adaptively attains high or comparable performance on summarization, translation and questions answering.

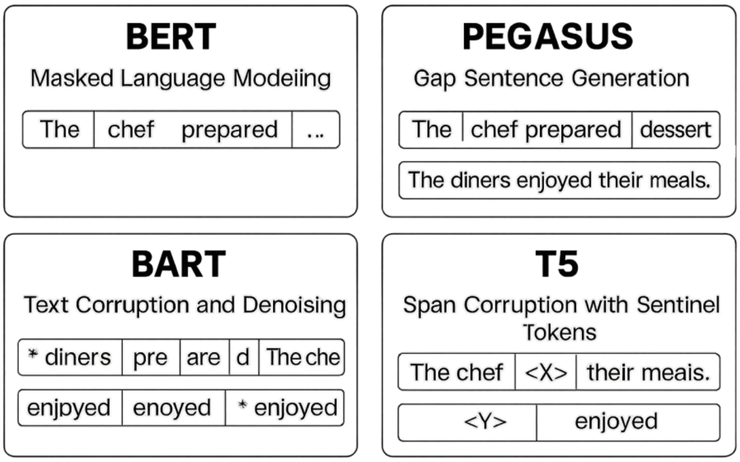

Figure 3. Pretraining Strategies of Transformer Models: BERT, PEGASUS, BART, and T5

4.6 Comparative Analysis

Table 2 denotes comparative analysis.

Table 2. Comparative Analysis

Aspect	**Strengths**	**Limitations**
Long-Range Dependency	Self-attention enables global context modeling	High memory usage for long input sequences
Parallelization	Fully parallelized training and	Transformer decoders remain

	inference	autoregressive during generation
Pretraining Effectiveness	Large corpora-based pretraining improves generalization	Domain adaptation needed for specialized text (e.g., biomedical, legal)
Abstractive Generation	Generates fluent, coherent, and novel summaries	Risk of hallucination and factual inconsistencies
Extractive Capabilities	BERTSUM effectively identifies salient sentences	Not applicable for generating new phrasing
Training Efficiency	Accelerated training due to parallelism	High compute requirements (e.g., GPUs/TPUs, large memory footprint)

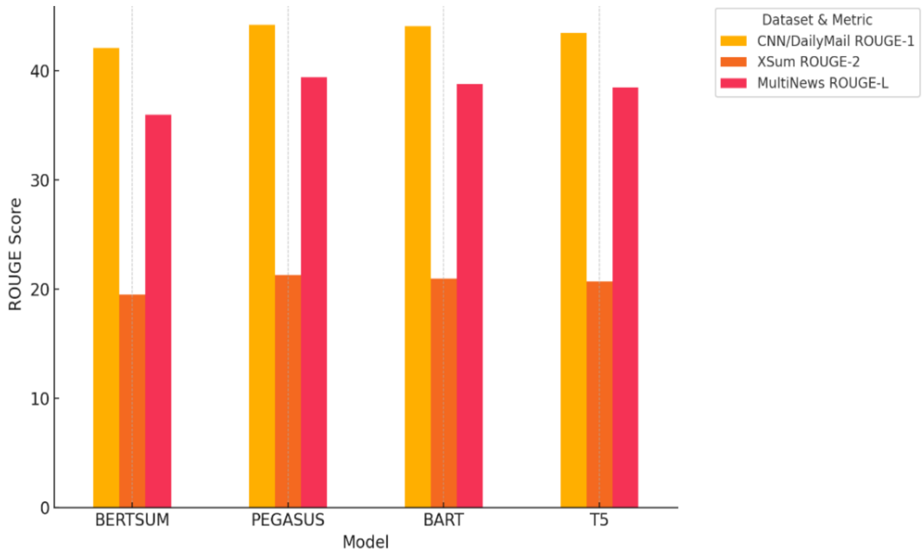

Figure 4. ROUGE Score Benchmark Comparison across Transformer Summarization Models

ROUGE Score Benchmark Comparison across Transformer Summarization Models is portrayed in Figure 4.

5. COMPARATIVE EVALUATION AND ANALYSIS

Transformer architectures have changed the text summarization field significantly. The section compares the architectures, their difficulty to train, the quality of their summaries, their suitability to extractive or abstractive summarization as well as their scalability along five key dimensions.

5.1 Model Architecture: Sequential vs. Parallel Processing

The nature of LSTM-based models is sequential, relying on processing one token after another, which decreases the level of parallelism and introduces latency in all processing operations. Despite the improvements (such as attention mechanisms or bidirectionality), it remains difficult to capture long-range dependencies with LSTM-based encoder-decoder models. Transformers do this with self-attention and can be parallelized completely. Consequently, the latency of processing all the tokens is achieved simultaneously, which allows fast training and inference. Besides that, Transformers allow to append global information via multi-head attention, which does not require recurrence and allows them to comprehend long sequences in a significantly improved way.

5.2 Training Time and Computational Complexity

Although LSTM models are fast to deploy and occupy less resources compared to other neural networks, they are slower to train because each step has to succeed the prior one. Even such transformer models, which are costly to train on due to their parallel structure, can be speeded up using GPU or TPU resources. The process of fine-tuning is more efficient when the size of the dataset is small since these models have been initially trained on huge text data. However, its primary limitation is memory consumption - self-attention grows huge as texts become longer and there is no way to address this problem unless Longformer-style optimizations or BigBird-style optimizations are applied.

5.3 Summarization Performance: ROUGE and BLEU Metrics

Based on the abstractive summarization reported by CNN/DailyMail, XSum and MultiNews, Transformers are obviously better than LSTMs. Such metrics like ROUGE and BLEU, verify whether the generated summary is analogous to that which was produced by a human being. PEGASUS and BART have surpassed ROUGE-1 of 44 due to their training on large pretraining datasets to generate digestible, diverse and stable chunks of information. But LSTM based models such as Seq2Seq with attention and pointer-generator networks show worst results (36-41 score) and in most cases, the generated summary either contains the same information repeated repeatedly or it is monotonous because of the limited ability to utilize context. BERTSUM, a Transformer designed to perform extractive summarization, scores higher in ROUGE-1 compared with numerous LSTM-based models, at approximately 42. By all means, Transformers outperform LSTMs at extraction and abstraction in summarization due to their superior designs and pre-training.

5.4 Suitability for Extractive vs. Abstractive Summarization

The first to perform well on abstractive summarization were LSTM, because they are able to produce novel sentences. The pointer-generator networks were aimed to work with unknown language and generate summary writing like a human being. However, when it came to longer texts, LSTMs tended to struggle, as they were unable to handle a lot of context and their encoding was not enough to make the summaries as clear as they needed to be and full of the relevant information. Alternatively, the Transformer models perform well in extractive and abstractive summary preparation. Such models as BERTSUM are based on the sentence-level embedding technology when they are employed in extracting important sentences. When an abstractive summary is required, BART, PEGASUS and T5 models will give you fluent and accurate texts. Since Transformers are able to take care of each component of the input, the summaries they generate are more useful and comprehensible than those generated by the LSTM-based models.

5.5 Scalability and Long Document Summarization

The LSTM models do not suit large dataset. As documents increase in length, context fidelity is more difficult to achieve due to information issues in the encoder-decoder line. In contrast to them, Transformer models process more context information due to positional embeddings and self-attention. However, Transformers struggle with documents that are very large since the computation of attention is quadratic. By incorporating new models like Longformer, BigBird and LED that employ sparse attention, the issue of elements dropping out is avoided and they can readily take as input over 4,000 wordsworth of text. The characteristics of the LSTM-based models and transformer-based models are given in Table 3.

Table 3. Comparative Evaluation of LSTM-Based and Transformer-Based Summarization Models

Criterion	LSTM-Based Models	Transformer-Based Models
Architecture	Sequential (RNN-based)	Parallel (Self-attention based)
Training Time	Slow due to recurrence	Faster with GPU acceleration
Computational Complexity	Lower resource requirement	High memory usage and compute-intensive
Performance (ROUGE/BLEU)	Moderate (ROUGE-1 ~36–41)	High (ROUGE-1 ~42–44.5)
Extractive Capability	Weak, often requires additional heuristics	Strong with models like BERTSUM
Abstractive Capability	Limited by bottlenecks	Strong generative fluency with BART, PEGASUS, T5
Scalability to Long Inputs	Poor (sequence bottleneck)	Better (with sparse attention models)
Interpretability	More interpretable in attention variants	Ongoing work on explainable Transformers

6. LIMITATIONS AND RESEARCH GAPS

Despite the impressive outcomes, current models of text summarizing face a number of challenges. The models making such statements in the generated text that are not being reflected in the input is a big problem. Healthcare and law are the most dangerous fields when it comes to hallucination. Since large pretrained models are trained on big data which might contain social or gender biases, they may lead to summaries with such biases as well.

Another issue is that the state-of-the-art models are not compatible with languages that have fewer data mostly because they are predominantly coded in English. In addition to that, the majority of models experience issues when attempting to explain content that is not a single narrative, be it a conversation, a table or a multi-style document. More so, ROUGE and BLEU evaluate word matches and disregard the quality of the summed text sense which makes measuring quality of summaries a difficult task. They demonstrate the need of more sophisticated, fair and interactive summarization systems. Limitations of the existing system are illustrated in Table 4.

Table 4. Research Gaps and Limitations

Limitation	Description
Factual Inconsistency	Generated content may not align with input facts (hallucinations)
Pretraining Bias	Model inherits societal and cultural biases from training data
Language Generalization	Poor performance in multilingual and low-resource settings
Non-linear/Multimodal Summarization	Difficulty in summarizing dialogues, tables, charts, and hybrid formats
Evaluation Bottlenecks	Existing metrics do not reflect semantic fidelity or coherence adequately

7. FUTURE DIRECTIONS

The presence of issues in the current area has given rise to a number of promising methods. There are also crucial attempts to employ the advantages of both symbolic AI and neural networks through combining them. A combination of guided logic and data-based learning can be used to generate summaries that individuals can simply comprehend and believe. Additional knowledge sources (external sources of knowledge graphs or models with retrieval) can also be useful in factual consistency: they prevent the machine making errors.

Zero-shot and low resource summarization are also getting increased consideration. We can attempt to generate summaries in languages or regions that do not have a large collection of training data by transfer learning, cross-lingual learning or prompting large language models. An emphasis is also made on creating explainable summarization models that will assist a user in comprehending why specific content is selected. edge computing is still in its infancy, efforts have been made to summarize data faster and with less computation with Transformer architectures that promise to allow devices to create summaries when not connected to the internet, or when power is limited. The use of these approaches is significant in developing more robust, tolerant and realistic protocols of generating summaries.

8. CONCLUSION

Summarization of textual content progressed rapidly beyond the common methods into the realms of deeper learning and LSTM and Transformer networks have played a significant role in the change. Although LSTM-based models assisted in neural summarization due to their learning and attention mechanics that are sequential in nature, Transformer-based models can perform it more effectively, as they pay attention to global features, have more efficient processing of all inputs and generate improved result summaries. BART, PEGASUS and T5 are transformers that demonstrate state-leading results in both extractive and abstractive summarization and are outstanding on various data sets.

Nevertheless, there are still numerous significant challenges. Summarization systems cannot be used in crucial cases because of fact-checking mistakes, high bias, and issues with multilingual assistance and the lack of decent metrics. As further research is conducted the principal focus must be to make the models more dependent, equitable and available by integrating, elucidating and thrifting their systems. It will be beneficial to work on these issues through local learning, reducing resource requirements, and edge deployment to develop the finest and reliable text summarization systems.

REFERENCES

[1] Bahdanau, D., Cho, K., & Bengio, Y. (2014). *Neural machine translation by jointly learning to align and translate*. arXiv preprint arXiv:1409.0473. https://arxiv.org/abs/1409.0473

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... &Polosukhin, I. (2017). *Attention is all you need*. Advances in Neural Information Processing Systems (NeurIPS), 30. https://doi.org/10.48550/arXiv.1706.03762

[3] See, A., Liu, P. J., & Manning, C. D. (2017). *Get to the point: Summarization with pointer-generator networks*. Proceedings of the 55th Annual Meeting of the Association for

Computational Linguistics (ACL), 2, 1073–1083. https://doi.org/10.18653/v1/P17-1099

[4] Liu, Y., &Lapata, M. (2019). *Text summarization with pretrained encoders*. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP), 3730–3740. https://doi.org/10.18653/v1/D19-1387

[5] Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2020). *PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization*. International Conference on Machine Learning (ICML), 11328–11339. https://arxiv.org/abs/1912.08777

[6] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). *BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

[7] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research (JMLR), 21(140), 1–67. https://jmlr.org/papers/v21/20-074.html

[8] Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. Text summarization branches out: Proceedings of the ACL-04 workshop, 74–81. https://aclanthology.org/W04-1013/

[9] Zhong, M., Tang, D., Qin, B., & Liu, T. (2020). Extractive summarization as text matching. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), 6197–6208. https://doi.org/10.18653/v1/2020.acl-main.552

[10] Fabbri, A. R., Kryściński, W., McCann, B., Xiong, C., Socher, R., & Radev, D. R. (2021). SummEval: Re-evaluating summarization evaluation. Transactions of the Association for Computational Linguistics (TACL), 9, 391–409. https://doi.org/10.1162/tacl_a_00373

[11] Kavitha, M. (2025). Hybrid AI-mathematical modeling approach for predictive maintenance in rotating machinery systems. Journal of Applied Mathematical Models in Engineering, 1(1), 1–8.

[12] Kowalski, T., & Nowak, M. (2024). The Impact of Digital Transformation on Quality Assurance in Healthcare Systems. National Journal of Quality, Innovation, and Business Excellence, 1(2), 1-12.