Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

A Modified Preprocessing Approach for Abstractive Text Summarization Using Transformers

Mohmmadali Saiyyad, Nitin Patil

R. C. Patel Institute of Technology, Shirpur, India
Email: sayyad1188@gmail.com

The exponential availability and use of textual resources on the digital media has made extraction of knowledge challenging. Users often looked for topic summaries from many sources in order to meet their information needs. Abstractive summarization of a single document is a way for Natural Language Processing (NLP) which seeks to produce a succinct synopsis of the original text from a unique document. It is not easy to find a method for extracting text from an input document and turning it into a natural language summary that highlights the important ideas. In this paper we have introduced our modified pre-processing technique called as Advanced Segmentation Method (ASM) with Transformer plus PGN (pointer-generator network) model. Transformers that have already been trained are being added to traditional machine learning models. Transformer-based language models that are self-supervised trained are receiving a lot of attention when they are optimized for tasks related to text summarization and processing the text. In this paper we have used Hunter Prey optimizer, Sail Fish Optimizer and Hunter Sailfish optimizer. We have done comparison with RNN, SeqtoSeq Attention, Bi-LSTM models.

Keywords: Abstractive text summarization, Transformer, hunter prey optimizer, RNN, LSTM, Bi-LSTM, SeqtoSeq plus Attention, Sail-fish optimizer.

1. Introduction

A wide range of devices may now access the internet, making it available to the general public. It could be challenging to pinpoint the precise information that is required and useful because there is a lot of unstructured material on the internet. When searching through a large amount of information, information overhead is a fairly typical issue and usually takes a lot of time. On the other hand, computerized text summary is offering a useful hand in obtaining the key idea in a shorter amount of time (Rahman et al.2019). Algorithms for summarizing fall into two main categories: extractive and abstractive. The words, phrases, or sentences from the original material are usually assembled using extractive summarization techniques, which usually choose one lengthy sentence at a time. In the past, extractive summarization techniques accounted for the majority of automatic summarizing research. Because abstractive

summarizing approaches are more in line with human thought processes, the target summary may include terms or phrases that are absent from the original text. It is commonly acknowledged that abstractive summarizing techniques consistently outperform extractive techniques, mostly due to their superior capacity to hold the essential key aspects of the original text during the condensing process (Wang et al. 2021).

In this manuscript we have discussed our Transform based preprocessing approach. The manuscript is elaborated as follows: Section 2 provides the previous research work done by various researchers related work for abstractive summarization. Section 3 describes preprocessing techniques. In the Section no 4 we describe Transformer architecture. In the Section 5 we describes Performance Matrices. Section 6 deals with experimental outcomes and elaboration of outcomes and Section 7 paper conclusion.

2. RELATED WORK

Abstractive summarizing approaches are difficult since the summary use fresh words or phrases that, like human beings, capture the essence of the entire text. We have discussed the research work done using various deep learning techniques.

(Zeng et al. 2024) provide a VIP-token centric compression (VCC) strategy that reduce the sentence sequence in a certain way, depending on how the sequence affects the approximation of the VIP-tokens representation.

(Ritika et al. 2024) focuses on developing an automated text summarizing framework that uses machine learning methodology to generate a summary from text input.to analyse the result. CNN dailymail dataset were incorporated. Two transformer-based language models, T5 and BART were used to check the performance in comparative manner. BART has a 3.02% higher ROUGE-1 Score than T5.

(Ahuir et al. 2024) proposed the technique for summarization of emotional data, Xsum and Cnn/Dailymail, as well as uses BART, Pegasus, and T5, to examine the summaries generated from the above models. The technique works in such a effective manner that original emotions are also reflected in the summary with retention of 75%.

To address the issue of inadequate summaries for long videos and inadequate summaries for short video, (Argade et al. 2024) introduced the Multimodal Abstractive Summarization with the help of BART in collaboration with attention mechanism. A bidirectional GRU work at the embedding layer for data encoding, while audio and video data encoding is handled by LSTM.

The processing of texts produced by humans has advanced significantly thanks to large language models (LLMs). One significant issue that still needs to be addressed is how to retain context when reading lengthy texts or several documents. The way that LLMs currently handle context retention is frequently ineffective in terms of time and storage. (Gunjan et al. 2024) design a two-step process for evolving summary and designing the answer for question method. The path allows for LLM is not inundated with redundant or unrelated material, giving a significant amount of time and resources. This method makes it easier to create concise responses and summaries, which improves the LLM's overall effectiveness.

(Sun et al. 2024) suggested an enhanced encoder-decoder model with multiobjective reinforcement learning and a tree based attention mechanism. Semantic analysis of the given input document can be achieved using the multi-head attention encoding. Out of vocabulary issued is resolved by taking pointer generator network into consideration. Throughout the training process, multi objective reinforcement learning archives semantic consistency, and improved readability.

(Saeed et al. 2024) implemented A specific module created to improve source text for PTMs (pre-trained models) is introduced in the suggested solution, known as medical text simplification and summarizing (MedTSS). Without requiring further training, MedTSS mitigates entity hallucination difficulties, resolves token limit issues, and reinforces several concepts. Additionally, the module analyses language to make generated summaries easier to understand; this feature is especially useful for difficult medical research publications.

(Hangbo et al. 2023) present a method to improve the BERT model so that it can infer the sequence to sequence work in linear fashion. Through the use of well-crafted self-attention masks and a unified modelling approach, In the elaborative method to provide the summary there is no extra decoder drafting is required.

(Diego et al. 2023) addressed the question of "hallucination" of abstractive text summary. The sequence encoding and sequence decoding method is used to produce the summary. Summary generation is the final task of sequence decoding component. whereas the encoder is in charge of extracting the overall meaning from the raw text. Legal professionals rely on the summaries to find precedents that support their arguments, therefore this problem—known as "hallucination"—represents a significant problem in legal writings. Legal documents are often very lengthy and might not be fed to the encoder completely, which is another cause for concern. In order to address these problems, a new approach, LegalSumm, multiple summaries are generated and they are tested over the model summary.

(Tong et al. 2023) suggested approach uses structured semantics and information from extracted knowledge graphs as a summarization guide. Encoders that can handle both textual and graphical inputs are Bi-functional transformer models, which are designed to improve BART, one of the most advanced Large model. To achieve the peak performance, decoding is carried out using this richer encoding. Wiki-Sum dataset is used and it is tested over various models to analyse the performance.

(Choi et al. 2023) proposed source code SUMmarization using an extraction and abstraction combined retrieval-augmented adaptive transformer. The proposed technique utilises an extractive methodology to improve the frequency of significant keywords by using a recovered summary of a comparable code. It also provides abstractive summarization of input text, accounting for the method. The augmented representation of code and generated code is used to propagate the hybrid code. the self-attention network performs encoding using both sequence and structural manner.

(Mishra et al. 2023) introduced Source code SUMmarization using an extraction and abstraction combined retrieval-augmented adaptive transformer. The proposed architecture uses an extractive mechanism to improve the frequency of significant keywords by using a recovered summary of a comparable code. Summarization is done using abstractive way,

which is responsible for information in sequence and proper structure.

(Cao et al. 2023) generates a model medical data summarization using a Multi-modal Memory Transformer Network (MMTN). Medical images are correlated in this model to improve the precision. The model uses word prediction technique to condense it with vision processing to generate the medical reports.

(Zheng et al. 2023) presents architecture of a transformer-based Chinese generative communication-model that realizes one-way language generation by using basic architecture of a non-complete mask and more than one-layer decoding model of transformer. With the use of relative positional coding, the suggested approach addresses the issue of absolute position. coding's inferiority in terms of long-distance information, it is more convenient to make a model which create dialogue in single way. The main part of the position embedding layer is replaced with relative position information in the transformer module, which also modifies the self-attention calculation formula.

(Laskar et al. 2022) suggested method by using the BERTSUM model, a straightforward transformer-based summarizing architecture that uses the Transformer decoder as the decoder and the BERT model as the encoder.

(A. almori et al. 2022) discussed datasets, methodologies, architectures, difficulties, proposed techniques, methodology introduced, result analysis, research trends, and performance analysis of deep learning models.

(Jonathan et al. 2020) carry out a brief extraction step, which is the important aspect to tone the transformer language model on raw data before generating a summary. Additionally, demonstrate that the proposed method achieves higher ROUGE scores while producing more abstractive summaries than previous work that uses a copy strategy. Also present in-depth comparisons with robust baseline methodologies, previous state-of-the-art work, and many iterations of methodology, such as those that just employ transformers.

(Gunel et al. 2020) offer exploratory encoder-decoder model that leverages TranformerXL concepts to encode longer term dependency and successfully embed semantic element information from the Wikidata information nodes in the calculation of attention. It also demonstrates improvements in performance with respect to transformer using the same resources.

(Dimitrii et al. 2020) proposed two new abstractive text summarization models that incorporate convolutional self-attention mechanism over lower layers and conditioning on a pre-trained language model which offers a window-based encoding technique that addresses BERT's input constraints and enables processing of larger input texts.

(Ekaterina et al. 2020) provides a new approach of hybrid Sequence-to-sequence with Transfer Learning. Unified Transformer techniques have been used to study the text summarization problem. The Seq2Seq model comprises an embedding layer, with three LSTM layers network for input and one LSTM layer for the output network. The output layer employs the SoftMax activation algorithm, and the custom attention layer was also employed to retain the extended sequences. The dimensions of the embedding layers are 200 units, and the hidden layers are 256 units in size. In addition, each hidden layer employs a drop-out value of 0.4 to lessen overfitting of the model and enhance performance.

(Su et al. 2020) suggested method which can accomplish variable-length abstractive summarization and fluency at the same time. A segmentation based design and a multistage transformer based architecture builds the suggested abstractive summarization model. Initially, input corpus is divided into segments by segmentation based design using combination of Bi-LSTM and BERT model. BERT is used to extract important sentence form the segment array (BERTSUM). The extracted array of data is used to give training to the model. A segmentation based design and the transformer based architecture are trained in turn until the point of convergence is achieved.

(Zhang et al. 2020) suggested using effective self-supervised training method based on transformer model which works on enormous text datasets. Similar to an extractive summary, PEGASUS groups together the high relativity sentences from an input document and by calculating co-relation gives an output stream out of the remaining sentences. PEGASUS model is examined on the basis of its generated summary of news, science, stories, instructions, emails and patents.

3. PREPROCESSING

In case of text data, preprocessing is very important step. It will help to shape the data in such a way that model can easily train over it. The preprocessing of data effectively allows to fine tune the model. The results are improved if the input data is clean and non-biased.

We have introduced a modified segmentation and tokenization process to get the input text in syntactic form. After preprocessing the text document is passed to the transformer architecture for further processing. The detail description is given below [Anandarajan et al. 2019]. A keyword is just a word with certain semantics. Phrases can build a genuine sentence and have richer meanings than keywords, therefore it would be advantageous to extract phrases from sentences. However, the drawback of nearly all of these techniques is the partial extraction of terms. We employ a more sophisticated phrase extraction technique called Advanced Segmentation Method (ASM), which is based on these phrase extraction techniques. More phrases can be extracted using this method than with other phrase extraction techniques currently in use.

ASM can assist us in extracting additional phrases, however it will also extract some incorrect and repetitive words. Therefore, after acquiring phrases in the phrase acquisition process, we must remove some of these incorrect or repetitive phrases.

$$\boldsymbol{S_{i,j}^{p}}\& = \boldsymbol{cos}\left(\vec{\boldsymbol{w}}_{\boldsymbol{p}} \odot \vec{\boldsymbol{v}}_{\boldsymbol{i}}, \vec{\boldsymbol{w}}_{\boldsymbol{p}} \odot \vec{\boldsymbol{v}}_{\boldsymbol{j}}\right) \quad (1)$$

Where $\odot$ indicates component multiplication and $\vec{\boldsymbol{w}}_{\boldsymbol{p}}$ is a effective vector of weight. $\boldsymbol{S_{i,j}^{p}}$.combination of co-sine similarity, $\vec{\boldsymbol{v}}_{\boldsymbol{i}}$ effective vector for learning to emphasize various node embedding dimensions (Chen et al. 2020)

$$\vec{\boldsymbol{c}}^{\boldsymbol{t}} = \sum_{\boldsymbol{t'=0}}^{\boldsymbol{t-1}} \quad \vec{\boldsymbol{a}}^{\boldsymbol{t'}}. \quad (2)$$

A coverage vector $\vec{\boldsymbol{c}}^{\boldsymbol{t}}$ with every step t is ths sum of all attention vectors $\vec{\boldsymbol{a}}^{\boldsymbol{t'}}$ (Tu et al. 2016)

$$\boldsymbol{f_{filter}} * \boldsymbol{x_i} = \boldsymbol{U f(\Lambda) U^T x_i} \quad (3)$$

Where U is effective matrix of Eigen vector. $\boldsymbol{x_i}$ is the signal. filter is in Fourier domain (Kipf et al. 2016)

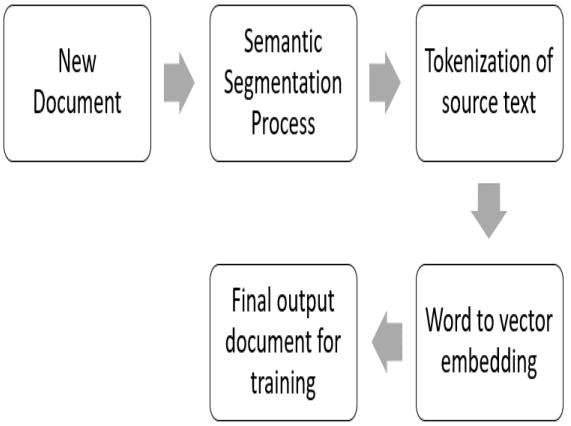

Figure 1. Modified Preprocessing Technique

Phases: Preprocessing Algorithm
Input: W_V: Word Vector
C_M: Cosine Matrix
S_C: Sentence Count
V_A: Selected Sentences Vector
Output: S_G: Group of Sentences
Initialize i=0
Initialize W_Y|
While Wv >0
Calculate C_M using Equation (1)
Calculate S_C using Equation (2)
Calculate V_A using Equation (3)
i=i-1
Selection: S_G Where $S_G \in Setof V_A$

Figure 2. Algorithm for Semantic Segmentation Process

4. Transformer architecture

A unique network design called Transformer was proposed by Google researchers (Vaswani et al. 2017) Fig. 3 shows the transformer model which can show connection between input and output. Fig. 3 shows Encoder-decoder module, connected layers, self-attention mechanism and multi-head attention.

Encoder is created using 6 layer placed on one another. The design of layer is placed in such

a way that every layer is connected to next layer to make feed forward network and creates multi-head self-attention mechanism. After layer normalization, a residual connection is used around each of the two sub-layers. Residual connection is established after normalization of each sub layer.

The decoder model also has six sub layer. First two layers in addition with the third layer uses attention mechanism to give the output. Normalization process is performed on each sub layer to so that residual network can be created. Modified self- attention is introduced to stop attention at succeeding position.

Attention is a Key-value related query and results are the key component of attention module. The results are calculated on the basis of sum of weights. The weights are calculated on the basis of query function. The most important aspect of multi-head attention is that it enables the model to share data from many representations. Figure 3 is the Layer representation of transformer model.

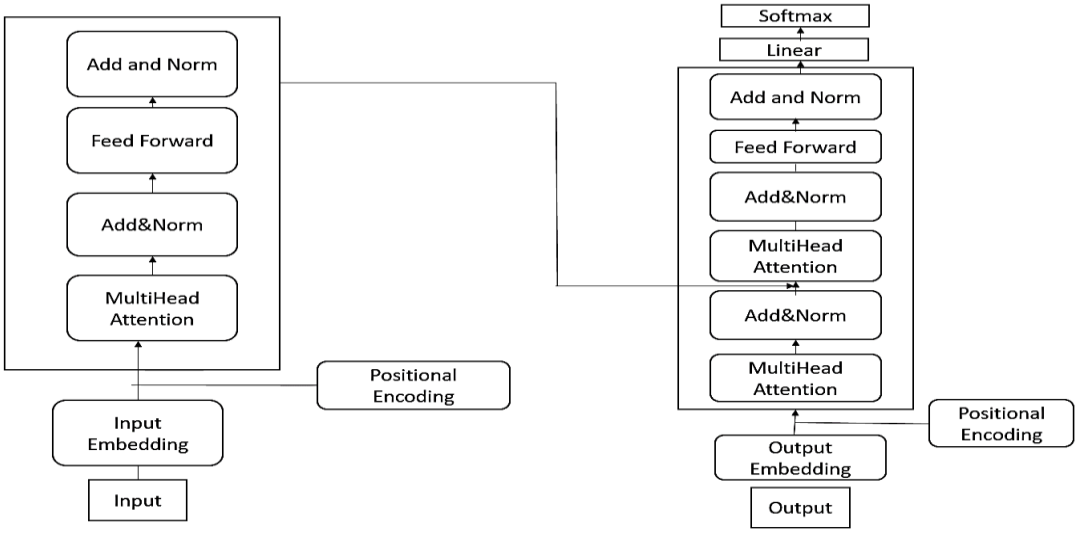

Figure 3. Layer representation of transformer model (Vaswani et al. 2017)

5. PERFORMANCE MATRICES

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is the matrices used for calculating performance of a model on the given dataset. First introduced in 2003, the ROUGE family of metrics is based on the similarity of n-grams. The formula for ROUGE score is as follows: (Steinberge et al. 2009)

$$R - N_r = \frac{\Sigma s_{\in ref} \Sigma gram_{N \in S}\ Count_m(gram_N)}{\Sigma s_{\in ref} \Sigma gram_{N \in S}\ Count(gram_N)} \qquad (4)$$

6. RESULTS AND DISCUSSION

We evaluate the models on a sufficiently large CNN/DM dataset. It contains over 200 million words. A few highlighted sections from each article add up to the overall summary. This dataset was produced in order to facilitate the creation of models capable of condensing lengthy text paragraphs into one or two sentences. The data is pre-processed using the improved algorithm implemented in python. In the Table 1 given below we have compared 4 models. Among the 4 models ASM plus Transformer plus PGN gives the best result. We have check the models with Hunter Prey Optimizer (Naruei et al. 2022) Sail Fish Optimizer (Shadravan et al.2019), and Hunter Sail Fish Optimizer (Aluri et al. 2023)

Table 1. Comparison of results with models on the basis of ROUGE Score (Hunter Prey optimizer)

Models	R1	R2	R3	RL
RNN	35.15	23.86	19.25	38.12
Bi-LSTM	38.12	25.23	22.56	40.56
Seq-Seq+Attention	40.19	28.16	24.78	41.63
ASM+Transformer+PGN	44.17	33.16	29.16	44.14

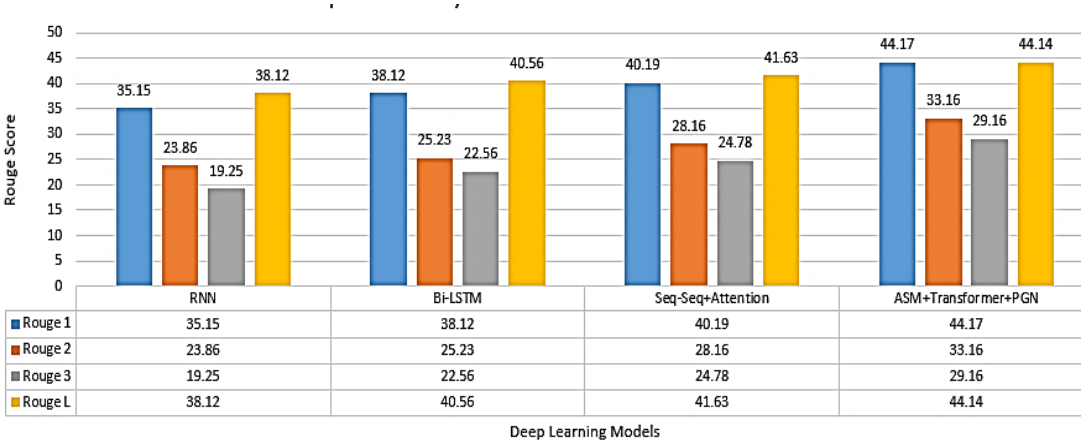

Figure 4: Analysis of performance on the basis of graphical statics of dataset (Hunter Prey optimizer).

Table 2. Comparison of results with models on the basis of ROUGE Score (Sail-Fish optimizer)

Models	R1	R2	R3	RL
RNN	36.15	24.48	20.21	39.22
Bi-LSTM	39.1	26.13	23.18	41.76
Seq-Seq+Attention	41.12	29.20	25.12	42.20
ASM+Transformer+PGN	45.17	34.20	30.22	44.80

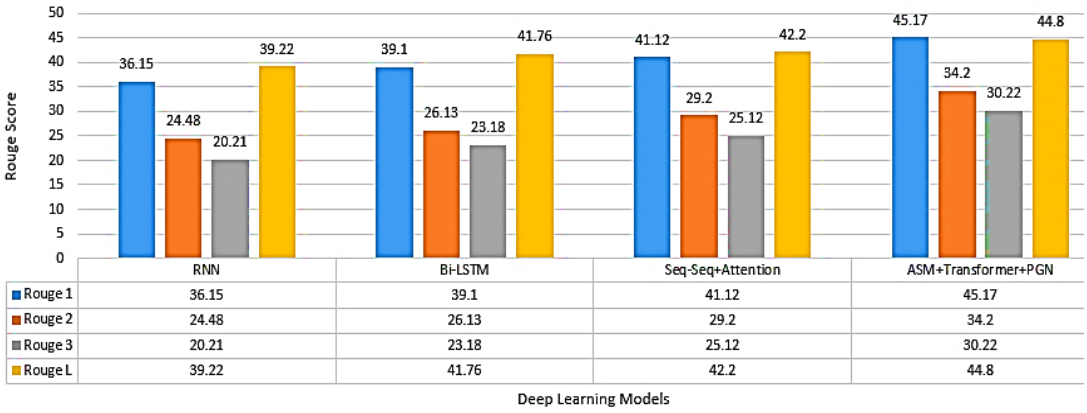

Figure 5: Analysis of performance on the basis of graphical statics of dataset (Sail Fish optimizer).

Table 3. Comparison of results with models on the basis of ROUGE Score (Hunter-Sail fish optimizer)

Models	R1	R2	R3	RL
RNN	36.50	24.90	20.80	39.75
Bi-LSTM	39.95	26.86	23.92	42.15
Seq-Seq+Attention	42.15	29.80	25.75	42.80
ASM+Transformer+PGN	46.20	35.40	31.23	45.40

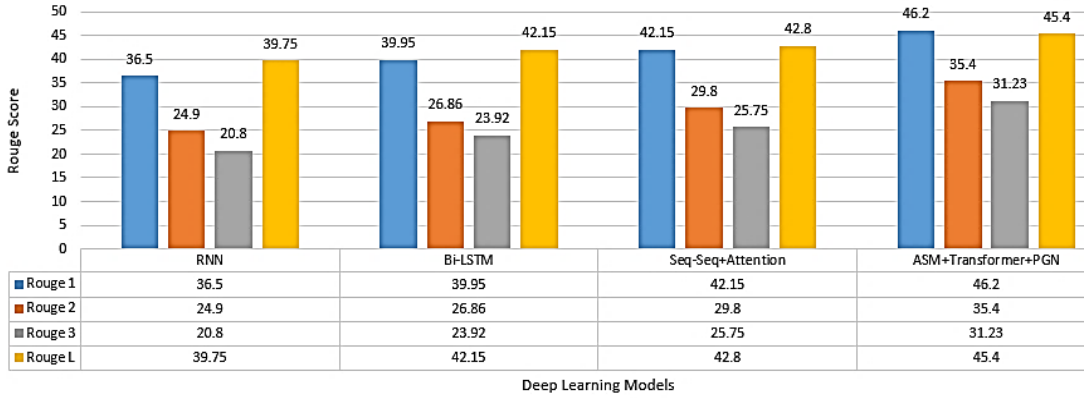

Figure 6: Analysis of performance on the basis of graphical statics of dataset (Hunter-Sail fish optimizer)

Original Article (Truncated):
Harry Potter star Daniel Radcliffe gains access to a reported £20 million ($41.1 million) fortune as he turns 18 on Monday, but he insists the money won't cast a spell on him. Daniel Radcliffe as Harry Potter in "Harry Potter and the Order of the Phoenix" To the disappointment of gossip columnists around the world, (...)
Reference Summary:
Harry Potter star Daniel Radcliffe gets £20M fortune as he turns 18 Monday. Young actor says he has no plans to fritter his cash away. Radcliffe's earnings from first five Potter films have been held in trust fund.
RNN Summary:
Daniel Radcliffe, the Harry Potter star, has been granted a reported £20 million fortune, but insists it won't affect him, despite being rumoured to have a spell.
Bi-LSTM Summary:
Daniel Radcliffe is not letting his newfound wealth affect him as he celebrates his 18th birthday. He remains humble despite his reported £20 million fortune.
Seq-Seq+Attention:
Daniel Radcliffe, known for his role as Harry Potter, has gained access to a £20 million fortune as he turns 18. He is determined not to let the money change him, much to the dismay of gossip columnists.
ASM+Transformer+PGN Summary:
Radcliffe stated that the money will not affect him and that he will continue to focus on his acting career. The young actor is famous and has earned of £20 million from work He Portrayed role as Harry Potter in the film series

Figure 7: Generated summaries with the comparative models

7. CONCLUSION

The Transformer model's structure has been improved, and this work presents an efficient application of the Advanced Segmentation Method (ASM), which is predicated on these phrase extraction strategies, with the intention of enhancing Automatic Text Summarization (ATS). Different type of corpus length is used to verify the results of the model experimented. The acquired experimental findings showed that the elaborated ASM plus Transformer plus PGN models outperformed several conventional and new deep learning model for text summarization.

The best-performing suggested model, "ASM plus Transformer plus PGN" have the ability to increased accuracy in such a way that the summarized text resemble more with the human generated summary. The model deals with lacking attention issue at the decoder level. In future work, the fastformer model can be experimented with the improved pre-processing phase.

References

1. Ahuir, V., González, J. Á., Hurtado, L. F., & Segarra, E. (2024). Abstractive Summarizers Become Emotional on News Summarization. Applied Sciences, 14(2), 713. doi: https://doi.org/10.3390/app14020713
2. Aksenov, D., Moreno-Schneider, J., Bourgonje, P., Schwarzenberg, R., Hennig, L., & Rehm, G. (2020). Abstractive text summarization based on language model conditioning and locality modeling. arXiv preprint arXiv:2003.13027. doi: https://doi.org/10.48550/arXiv.2003.13027
3. Alomari, A., Idris, N., Sabri, A. Q. M., & Alsmadi, I. (2022). Deep reinforcement and transfer learning for abstractive text summarization: A review. Computer Speech & Language, 71, 101276. doi: https://doi.org/10.1016/j.csl.2021.101276
4. Aluri, L., & Latha, D. (2023). HSFO: Hunter Sail Fish Optimizer enabled deep learning for single document abstractive summarization based on semantic role labelling for Telugu text. doi: https://doi.org/10.21203/rs.3.rs-2889668/v1
5. Anandarajan, M., Hill, C., & Nolan, T. (2019). Practical text analytics. Maximizing the Value of Text Data. (Advances in Analytics and Data Science. Vol. 2.) Springer, 45-59. doi: https://doi.org/10.1007/978-3-319-95663-3
6. Argade, D., Khairnar, V., Vora, D., Patil, S., Kotecha, K., & Alfarhood, S. (2024). Multimodal abstractive summarization using bidirectional encoder representations from transformer with attention mechanism. Heliyon. doi: https://doi.org/10.1016/j.heliyon.2024.e26162
7. Bao, H., Dong, L., Wang, W., Yang, N., Piao, S., & Wei, F. (2024). Fine-tuning pretrained transformer encoders for sequence-to-sequence learning. International Journal of Machine Learning and Cybernetics, 15(5), 1711-1728. doi: https://doi.org/10.1007/s13042-023-01992-6
8. Cao, Y., Cui, L., Zhang, L., Yu, F., Li, Z., & Xu, Y. (2023, June). MMTN: multi-modal memory transformer network for image-report consistent medical report generation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 37, No. 1, pp. 277-285) doi: https://doi.org/10.1609/aaai.v37i1.25100
9. Chen, T., Wang, X., Yue, T., Bai, X., Le, C. X., & Wang, W. (2023). Enhancing Abstractive Summarization with Extracted Knowledge Graphs and Multi-Source Transformers. Applied Sciences, 13(13), 7753. doi: https://doi.org/10.3390/app13137753
10. Chen, Y., Wu, L., & Zaki, M. (2020). Iterative deep graph learning for graph neural networks: Better and robust node embeddings. Advances in neural information processing systems, 33, 19314-19326. doi: https://doi.org/10.48550/arXiv.2006.13009

11. Choi, Y., Na, C., Kim, H., & Lee, J. H. (2023). READSUM: Retrieval-Augmented Adaptive Transformer for Source Code Summarization. IEEE Access. doi: 10.1109/ACCESS.2023.3271992
12. Feijo, D. D. V., & Moreira, V. P. (2023). Improving abstractive summarization of legal rulings through textual entailment. Artificial intelligence and law, 31(1), 91-113. doi: https://doi.org/10.1007/s10506-021-09305-4
13. Gunel, B., Zhu, C., Zeng, M., & Huang, X. (2020). Mind the facts: Knowledge-boosted coherent abstractive text summarization. arXiv preprint arXiv:2006.15435. doi: https://doi.org/10.48550/arXiv.2006.15435
14. Keswani, G., Bisen, W., Padwad, H., Wankhedkar, Y., Pandey, S., & Soni, A. (2024). Abstractive Long Text Summarization Using Large Language Models. International Journal of Intelligent Systems and Applications in Engineering, 12(12s), 160-168. doi: https://ijisae.org/index.php/IJISAE/article/view/4500
15. Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907. doi: https://doi.org/10.48550/arXiv.1609.02907
16. Laskar, M. T. R., Hoque, E., & Huang, J. X. (2022). Domain adaptation with pre-trained transformers for query-focused abstractive text summarization. Computational Linguistics, 48(2), 279-320. doi: https://doi.org/10.1162/coli_a_00434
17. Mishra, G., Sethi, N., & Hu, Y. C. (2023). Intelligent Abstractive Text Summarization using Hybrid Word2Vec and Swin Transformer for Long Documents. International Journal of Computer Information Systems & Industrial Management Applications, 15. https://openurl.ebsco.com/EPDB%3Agcd%3A1%3A19932737/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A174486579&crl=c
18. Naruei, I., Keynia, F., & Sabbagh Molahosseini, A. (2022). Hunter–prey optimization: Algorithm and applications. Soft Computing, 26(3), 1279-1314. doi: https://doi.org/10.1007/s00500-021-06401-0
19. Pilault, J., Li, R., Subramanian, S., & Pal, C. (2020, November). On extractive and abstractive neural document summarization with transformer language models. In Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP) (pp. 9308-9319). doi: 10.18653/v1/2020.emnlp-main.748
20. Rahman, M. M., & Siddiqui, F. H. (2019). An optimized abstractive text summarization model using peephole convolutional LSTM. Symmetry, 11(10), 1290. doi: https://doi.org/10.3390/sym11101290
21. Rao, R., Sharma, S., & Malik, N. (2024). Automatic text summarization using transformer-based language models. International Journal of System Assurance Engineering and Management, 1-7. doi: https://doi.org/10.1007/s13198-024-02280-4
22. Saeed, N., & Naveed, H. (2024). MedTSS: transforming abstractive summarization of scientific articles with linguistic analysis and concept reinforcement. Knowledge and Information Systems, 1-18. doi: https://doi.org/10.1007/s10115-023-02055-6
23. Shadravan, S., Naji, H. R., & Bardsiri, V. K. (2019). The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. Engineering Applications of Artificial Intelligence, 80, 20-34. doi: https://doi.org/10.1016/j.engappai.2019.01.001
24. Steinberger, J., & Jezek, K. (2009). Evaluation measures for text summarization. Computing and Informatics, 28(2), 251. doi: https://www.cai.sk/ojs/index.php/cai/article/view/37
25. Su, M. H., Wu, C. H., & Cheng, H. T. (2020). A two-stage transformer-based approach for variable-length abstractive summarization. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 28, 2061-2072. doi: 10.1109/TASLP.2020.3006731
26. Sun, Y., & Platoš, J. (2024). Abstractive Text Summarization Model Combining a Hierarchical Attention Mechanism and Multiobjective Reinforcement Learning. Expert Systems with

Applications, 123356. doi: https://doi.org/10.1016/j.eswa.2024.123356

27. Tu, Z., Lu, Z., Liu, Y., Liu, X., & Li, H. (2016). Modeling coverage for neural machine translation. arXiv preprint arXiv:1601.04811. doi: https://doi.org/10.48550/arXiv.1601.04811
28. 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, 30. https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
29. Wang, Z. (2021, April). An Automatic Abstractive Text Summarization Model based on Hybrid Attention Mechanism. In Journal of Physics: Conference Series (Vol. 1848, No. 1, p. 012057). IOP Publishing. doi: 10.1088/1742-6596/1848/1/012057
30. Zeng, Z., Hawkins, C., Hong, M., Zhang, A., Pappas, N., Singh, V., & Zheng, S. (2024). Vcc: Scaling Transformers to 128K Tokens or More by Prioritizing Important Tokens. Advances in Neural Information Processing Systems, 36. https://proceedings.neurips.cc/paper_files/paper/2023/hash/4054556fcaa934b0bf76da52cf4f92cb-Abstract-Conference.html
31. Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020, November). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In International conference on machine learning (pp. 11328-11339). PMLR. https://proceedings.mlr.press/v119/zhang20ae
32. Zheng, W., Gong, G., Tian, J., Lu, S., Wang, R., Yin, Z., & Yin, L. (2023). Design of a modified transformer architecture based on relative position coding. International Journal of Computational Intelligence Systems, 16(1), 168. doi: https://doi.org/10.1007/s44196-023-00345-z
33. Zolotareva, E., Tashu, T. M., & Horváth, T. (2020, September). Abstractive Text Summarization using Transfer Learning. In ITAT (pp. 75-80). https://ceur-ws.org/Vol-2718/paper28.pdf