An Empirical Analysis of Attention Based Solution applied to Text Summarization Task

Simone Deola[1,†], Ricardo A. Matamoros A.[1,†] and Luca Leo Del Vescovo[2,†]

[1]*Università degli studi di Milano-Bicocca, Piazza dell'Ateneo Nuovo, 1, 20126 Milano MI*

[2]*Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133 Milano MI*

Abstract

During the last years, we have seen the blooming of attention based solution in the NLP field, improving all the state-of-the-art performances set by traditional methods. While analyzing these solutions, for solving the text summarization task, we find different methodologies of evaluation and training in each of them. The evaluation could change for dataset used, resourced used for training and scoring used, making the comparison hard to be evaluated. In this paper we show the results on text summarization of some solutions analyzed, done on the same dataset (CNN/Dailymail), using the same amount of resources and using the same score functions (ROUGE). These results are used to give a fair empirical comparison on this specific task.

Keywords

Abstractive Text Summarization, Transformer

1. Introduction

The Abstractive Text summarization is the Natural Language Processing task that consist of generating concise, coherent summaries by understanding and rephrasing the input text, unlike extractive summarization which selects and rearranges existing sentences. It involves natural language processing techniques and deep learning models to produce human-like summaries from long documents or articles. In this paper we will show the results of an empirical analysis of some of the solution proposed in literature, focusing mainly on new solutions based on the Transformer architecture. We tried to maintain the comparison between these results as fair as possible by using similar amount of resources for each training.

2. State-of-the-Art

The Text Summarization problem is an extensively studied task explored in the Natural Language Processing field since the 50s, and the solutions proposed to solve it range from traditional methods like Bag of Words, words embedding and Words2Vec to the more recent usage of deep learning techniques like RNN and LSTM.[1]

*Workshop on Artificial Intelligence and Applications for Business and Industries (AIABI 2023) co-located with 22th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2023), Rome, Italy*

s.deola1@campus.unimib.it (S. Deola); r.matamorosaragon@campus.unimib.it (R. A. M. A.); 10692678@polimi.it (. L. L. D. Vescovo)

0000-0001-5531-6684 (S. Deola); 0000-0002-1957-2530 (R. A. M. A.); 0009-0007-0100-6091 (. L. L. D. Vescovo)

In the last years, with the introduction of the Attention Mechanism and the consequent usage inside the Transformer model, we have seen an improvement on most of the NLP solutions performances, including the one used for solving the text summarization task. The attention Mechanism, introduced by Bahdanau [2], aims to simulate the cognitive attention by giving the model the ability to focus on specific parts of the input when managing large amount of text. The specific NLP task analyzed in the paper was machine translation, but this solution has been then applied to different tasks.

In the following year, Vaswani [3] proposed a new architecture that uses a specific kind of attention, Self Attention, combined to Positional Encoding, improving the time efficiency of the Attention Mechanism on sequence to sequence NLP tasks. This new architectures show state-of-the-art performances on different benchmark NLP tasks, setting up a new standard for the following NLP solutions.

The Transformer model has been then used to solve a variety of problems that involves Natural Language data, Computer vision problems, managing Graph data, and others. In this paper, we will analyze mainly solutions based on the Transformer model, including a set of the most promising Large Language Models. These kinds of models are usually based on the Transformer's architecture, and they use a new paradigm of training, that is usually called pre-training/fine-tuning, after the name given to the two steps that compose it. [4]

In the pre-training phase, the model is trained on a huge amount of data in order to solve one or more generic tasks. These tasks are usually based on datasets that can be automatically built from free text, in this way these models can be pre-trained on an arbitrary large set of data.

For example, the BERT model[4] is trained on the Masked Language Model task, that consist of predicting one word "masked" inside the original text. The dataset for this task is simply built by removing words at random from the input and substituting it with the [MASK] token. The model goal is to predict the original word. These generic tasks are meant to give to this pre-trained model the ability to understand the rules that guide the language. The pre-training step is the heaviest part of the training but is computed only once for each model and each language.

After that the model can be fine-tuned for any specific task, like Q&A, text summarization, translation, etc. During this step, the model is further trained using a supervised dataset. During last years some solutions proposed the removing of the fine-tuning step and the usage of the pre-trained model directly to solve the supervised tasks. We consider this kind of solution out of the scope of this analysis, but we will explore some of this solutions in future publications.

In this paper we will mainly focus on the Transformer model [3], LLM like GPT-2[5] and T5[6] and some techniques of transformer weights initialization, using the LLM BERT[7].

2.1. Models

Since all the models we used are based on slightly modified versions of the Transformer architecture, we start explaining briefly the Transformer architecture itself. Transformers are a groundbreaking neural network architecture that revolutionized natural language processing tasks. They introduced the concept of self-attention mechanism, which allows the model to weigh the relevance of different words in a sentence. This attention mechanism enables transformers to capture long-range dependencies and contextual information effectively.

Besides, unlike traditional recurrent neural networks, transformers process input sequences in parallel, making them highly efficient for both training and inference. Transformers excel in various NLP tasks, including machine translation, text classification, and language generation. Their ability to capture rich semantic relationships and handle long-range dependencies has made them a fundamental building block in modern language models such as GPT and BERT.

BERT [4] is a language representation model, based on a stack of Transformer encoders. Released by Google in 2018, it shows state-of the-art performances on eleven benchmark NLP task. Due to the composition of its architecture, BERT is not suited for generative task so, in order to apply it on the text summarization, we explore the solution proposed by Lewis et al.[7]. The technique proposed uses the weights of BERT in order to initialize a basic Transformer model, that can be then fine-tuned to solve the task. The initialization process shows improvement on the performances of the downstream task with respect to the non initialized version, demonstrating the usefulness of the pre-training process for general understanding of the language.

GPT-2 [5] (Generative Pre-trained Transformer 2) is a (ex) state-of-the-art language model, developed by OpenAI. In 2020, It gained significant attention due to its impressive language generation capabilities. It is a transformer-based (decoder-only) model with 1.5B parameters, enough to learn complex patterns and generate coherent, fluent and human-like text in different NLP tasks, such as translations, summaries, and answers. Its success has paved the way for advancements in language understanding and generation models.

T5 [6] (Text-to-Text Transfer Transformer) is a state-of-the-art language model developed by Google. It is also based on the transformer architecture (it just removes the Layer Norm bias, placing the layer normalization outside the residual path, and uses a different position embedding scheme). It was designed to perform a wide range of NLP tasks using a unified framework by introducing the concept of “text-to-text” transfer learning, where different NLP tasks are cast as text-to-text transformations. In this way, T5 can be trained on a diverse set of tasks, including machine translation, text summarization, question answering, and more. It exhibits remarkable performance across various tasks, showcasing its ability to generalize and transfer knowledge effectively. Its flexible and modular design, coupled with its impressive results, has made T5 a widely adopted model for solving diverse NLP problems.

3. Experiments

The experiments were conducted using Vertex AI by Google, utilizing its powerful computational resources according to the dimension of the model. To have comparable results, we used the same dataset for all the different models, and we tried to keep the same hyperparameters as discussed below.

3.1. Dataset

In our experiments, we used the widely known CNN-Dailymail dataset[8]. The CNN-DailyMail dataset is by far the most complete, and it's extensively used as a benchmark dataset in the field of abstractive text summarization. It consists of 300k news articles paired with corresponding human-generated summaries from two major news sources, CNN and Daily Mail.
The dataset was originally created for machine reading, comprehension, abstractive Q&A and development of automatic text summarization models, but then it became popular for abstractive text summarization too. There are two main reasons why the CNN-DailyMail dataset is suitable for studying abstractive summarization:

- it contains a large collection of different news articles, providing a diverse range of topics and writing styles. The dataset covers various domains such as politics, sports, entertainment, and more;
- the summaries in this dataset are abstractive in nature, meaning they do not simply extract sentences or phrases from the source text but instead generate human-like summaries that capture the key information and essence of the articles.

3.2. Data Preprocessing

One of the most significant challenge encountered in our research is the input limitation imposed by our models, which can only handle a maximum of 512 tokens(360 words), like T5 and BERT, or 1024 tokens(720 words), like GPT-2. The transformer model can have input of any dimension, in theory, but the time and memory cost are quadratic with respect to the input dimension, so we decide to consider a 512 input dimension Transformer. This poses a problem since our dataset consists mainly of samples that are in between 500 and 1500 tokens in length (Fig. 1).

As a result, the models are inherently unable to fully capture the entirety of these longer instances, potentially leading to information loss and incomplete summarization. In order to deal with such long-form texts, we excluded outliers (above 2000 tokens) and truncated the articles, since usually the most relevant information is in the first part of the article. On the other hand, this may impact the overall coherence and effectiveness of the generated summaries. Addressing this issue is crucial to ensure comprehensive and accurate abstractive text summarization on lengthy input samples.

In addition to the data truncation, we also adapt the input texts to the required input for the specific model, using two different methodologies: concatenation and teacher forcing.
For the generative model (GPT-2), that uses only an input sequence to be trained on the task of next word prediction, we used the approach explored in the original GPT-2 paper. This approach consists of concatenating the article text with the summary, divided by a new separator token [TL;DR] added to the original tokenizer. In this way, the model will be trained on the relation between a text and its summary and, during prediction, the model will use the newly added token as start point of the summarization process.
The Transformer models and T5 need a pair of text as inputs, one for the encoder, the other for the decoder. Since the decoder input needs to be the output of the previous prediction,

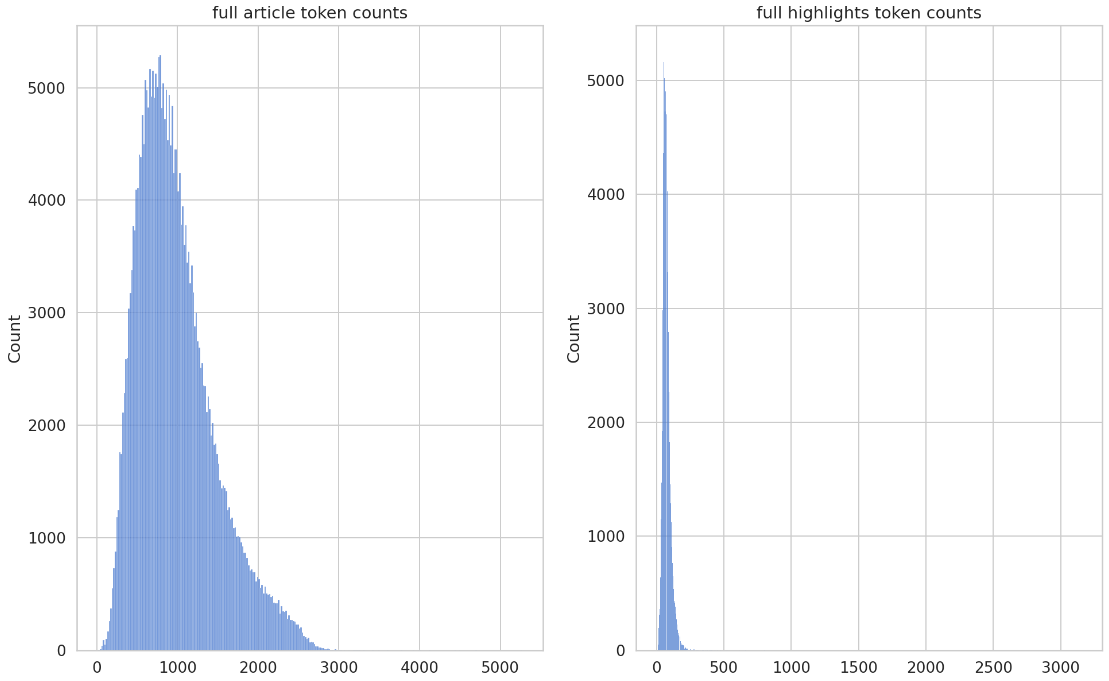

Figure 1: T5 tokenizer (other tokenizers produce similar results)

the training process needs one record for each token in each phrase of the summary in the dataset. This procedure is time expensive and doesn't fully exploit the parallelization power of the decoder self attention. In order to avoid that, the teacher forcing procedure uses the output, shifted by one in the right direction, as input of the decoder, instead of the produced output. The Transformer decoder, through the usage of a causal mask, can only attend to token on previous position during the prediction, so the shift avoid the model to attend to the correct token given in input during prediction.
No other text preprocessing is performed. The standard tokenizer for the given model is applied.

3.3. Hyperparameters

The models have been trained using Adam optimizers[9], with slightly different configurations, defined after some preliminary experiments. We trained the basic transformer using a learning rate up to 0.0005, linear warm-up of 6715 steps, normalization by the square root of the hidden size, and square root decay. The same schedule as been applied to the initialized Transformer, but using a learning rate up to 0.004 and 40k warm-up steps. For the T5 model and the GPT one, we used a fixed learning rate of 0.0001 and 0.0003 respectively.
The batch size used is mainly driven by the dimension of the models. We used a batch size of 128 for the base Transformer, 128 for the initialized Transformer, 16 for the GPT-2 model and 128 for the T5 model.
The number of epochs has been set to a high value and then limited using the early stopping

strategy on validation loss, with Patience of 4 epochs at max. The number of epochs has been set to 20 for all the models, but in all the cases the early stopping optimization technique stopped the training earlier.

4. Evaluation metric

Rouge Score[10] was chosen for abstractive text summarization due to its common usage and ease of implementation. It allows for quantitative assessment by measuring overlap between generated summaries and references. In particular, we adopted the Rouge-2 F1 metric, as it is widely used in research to evaluate the quality and effectiveness of generated summaries by comparing them against human-generated references.

F1 score is a metric commonly used in binary classification tasks to measure the model's performance. It combines precision and recall into a single value, providing a balanced evaluation of the model's effectiveness.

$$F1 = \frac{2*Precision*Recall}{Precision+Recall}$$

where Precision and Recall are calculated as follows:

$$Precision = \frac{count_{match}(gram_2)}{count(gram_2)} = \frac{number of 2-grams found in model and reference}{number of 2-grams in \mathbf{model}}$$

$$Recall = \frac{count_{match}(gram_2)}{count(gram_2)} = \frac{number of 2-grams found in model and reference}{number of 2-grams in \mathbf{reference}}$$

However, Rouge has limitations, notably its bias towards extractive methods, insensitivity to semantic quality, and inability to evaluate coherence and fluency. Hence, a need for human evaluation arises to assess these crucial aspects, providing a more holistic understanding of the model's performance in abstractive text summarization.[11] An example of this problem is provided below:

SCORES				
REFERENCE	MODEL	rouge2-f1	rouge2-precision	rouge2-recall
The quick brown fox jumps over the lazy dog	The fast orange fox leaps on the slow dog	0	0	0
The quick brown fox jumps over the lazy dog	The **quick** orange fox leaps on the **lazy** dog	0.375	0.375	0.375

5. Results and analysis

Hyperparameters				
Hyperpar.	Gpt-2	T5	Bert2Rand	Transformer
batch_size epochs	128	128	128	128

ROUGE F1 scores				
Rouge	Gpt-2	T5	Bert2Rand	Transformer
1-F1	0.26	0.4	0.36	0.3
2-F1	0.08	0.19	0.15	0.11
L-F1	0.18	0.28	0.26	0.22

Data	
Article	Summary
(CNN)Never mind cats having nine lives. A stray pooch in Washington State has used up at least three of her own after being hit by a car, apparently whacked on the head with a hammer in a misguided mercy killing and then buried in a field – only to survive. That’s according to Washington State University, where the dog – a friendly white-and-black bully breed mix now named Theia – has been receiving care at the Veterinary Teaching Hospital. Four days after her apparent death, the dog managed to stagger to a nearby farm, dirt-covered and emaciated, where she was found by a worker who took her to a vet for help. She was taken in by Moses Lake, Washington, resident Sara Mellado. ”Considering everything that she’s been through, she’s incredibly gentle and loving,” Mellado said, according to WSU News. ”She’s a true miracle dog and she deserves a good life.” Theia is only one year old but the dog’s brush with death did not leave her unscathed. She suffered a dislocated jaw, leg injuries and a caved-in sinus cavity – and still requires surgery to help her breathe. The veterinary hospital’s Good Samaritan Fund committee awarded some money to help pay for the dog’s treatment, but Mellado has set up a fundraising page to help meet the remaining cost of the dog’s care. She’s also created a Facebook page to keep supporters updated. Donors have already surpassed the $10,000 target, inspired by Theia’s tale of survival against the odds. On the fundraising page, Mellado writes, ”She is in desperate need of extensive medical procedures to fix her nasal damage and reset her jaw. I agreed to foster her until she finally found a loving home.” She is dedicated to making sure Theia gets the medical attention she needs, Mellado adds, and wants to ”make sure she gets placed in a family where this will never happen to her again!” Any additional funds raised will be ”paid forward” to help other animals. Theia is not the only animal to apparently rise from the grave in recent weeks. A cat in Tampa, Florida, found seemingly dead after he was hit by a car in January, showed up alive in a neighbor’s yard five days after he was buried by his owner. The cat was in bad shape, with maggots covering open wounds on his body and a ruined left eye, but remarkably survived with the help of treatment from the Humane Society.	Theia, a bully breed mix, was apparently hit by a car, whacked with a hammer and buried in a field . ”She’s a true miracle dog and she deserves a good life,” says Sara Mellado, who is looking for a home for Theia .

EVALUATION			
Gpt-2	T5	Bert2Rand	Transformer
A stray cat has apparently been hit by a car, apparently whacked on the head with a hammer. The dog is in need of extensive medical procedures to help the dog recover. She has a dislocated jaw, leg injuries and a caved-in sinus cavity.	A stray pooch in Washington State has used up at least three of her own after being hit by a car. The dog staggered to a nearby farm, dirt-covered and emaciated, where she was found. She still requires surgery to help her breathe.	a stray pooch in washington state has been hit by a car and then buried in a field. the dog was found by a worker who took her to a vet for help. theia is in desperate need of extensive medical procedures to fix her nose.	stray pooch killed and buried in a field in washington state, washington. dog's brush with death did not leave her in a critical condition. dog's brush with death did not leave her unscathed.

5.1. Results analysis

The comparison of the resulting performances shows that:

- the usage of pre-training LLM improves the performances of transformer based architecture on the text summarization task. In fact, the transformer initialized using BERT improves the basic Transformer architecture, while the T5 model (full transformer architecture, pre-trained) outperform both the previous.
- the big difference in performances between T5 and the two Transformers model can be addressed by the difference in pre-training dataset dimension (T5 used the bigger Common Crawl dataset wrt the datasets used for BERT). Also, the initialized transformer is pre-trained only on the decoder part, while T5 is pre-trained in its entirety.
- The GPT-2 models in this experiment perform badly wrt. the other models. This is probably due to the limited fine-tuning that we applied on this specific model with respect to the other models, especially the batch size. This is mainly due to the dimension of the model, composed of 1.5B parameter, harder to fine-tune with the same parameters of the other models, having a dimension of 200M parameters, for computational costs reason. We didn't consider the results achieved comparable to the others model results. However, we left it in this analysis to show the limitation on training bigger models. Also, the resulting texts produced by GPT-2 are still coherent, grammatically and with the textual content of the input, so will be further analyzed outside the ROUGE evaluation, for comparison.

6. Conclusion

6.1. Known limitations

To conclude the analysis we want to address the known limitations of the methodologies applied during these experiments. The results obtained heavily relies on the hyperparameters used for the training, such as the batch size, the learning rate and optimizer, the number of epochs, ecc. In order to obtain the best result for each of the models, a strategy of hyperparameter tuning should be applied, instead of the preliminary experiments that we used in this analysis. Also, the GPT-2 training was not sufficient as explained in the previous chapter. Both limitations was mainly driven by the limited resources available for this analysis.

6.2. Final Remarks

In this paper we proposed an empirical analysis of some Transformer based model, applied to the text summarization tasks. We selected four models: Transformer, Transformer initialized with BERT, GPT-2 and T5. We run an experiment on the benchmark dataset CNN/Dailymail to evaluate empirically the performances of such models on the summarization task. The results show that, when trained using similar resources, the usage of LLM (T5, BERT) improves the ability of such model to perform the task. The analysis also explore the challenge and advantage of using such models for solving the task of abstractive text summarization, showing the limitation encountered in a real case scenario.

7. Citations and Bibliographies

References

[1] M. F. Mridha, et al., A survey of automatic text summarization: Progress, process and challenges, IEEE Access 9 (2021) 156043–156070.

[2] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).

[3] A. Vaswani, et al., Attention is all you need, Advances in neural information processing systems 30 (2017).

[4] J. Devlin, et al., Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[5] A. Radford, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.

[6] C. Raffel, et al., Exploring the limits of transfer learning with a unified text-to-text transformer, The Journal of Machine Learning Research 21 (2020) 5485–5551.

[7] S. Rothe, S. Narayan, A. Severyn, Leveraging pre-trained checkpoints for sequence generation tasks, Transactions of the Association for Computational Linguistics 8 (2020) 264–280.

[8] R. Nallapati, et al., Abstractive text summarization using sequence-to-sequence rnns and beyond, arXiv preprint arXiv:1602.06023 (2016).

[9] Z. Zhang, Improved adam optimizer for deep neural networks, in: 2018 IEEE/ACM 26th international symposium on quality of service (IWQoS), IEEE, 2018.

[10] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text summarization branches out, 2004.

[11] N. Schluter, The limits of automatic summarisation according to rouge, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2017.