LSTM and BART: comparison in title generation from abstracts in papers

Castañeda Quichis, Nicolas
Camacho Valencia, Aaron
Remon Flores, Juaquin

Abstract

This is a comparative paper between Long Short-Term Memory Network [LSTM] and Bidirectional and Auto-Regressive Transformer [BART]. It aims to verify the cost/benefit of one over the other under certain circunstances. Although BART is a more consistent model and yields better results in text summarization, LSTM is not far behind. This work seek to demostrate that in small text bodies, LSTM can achieve results comparable to those of BART, being a lighter and less expensive model.

Introduction

Context and Background

Natural Language Processing (NLP), a fundamental branch of artificial intelligence, focused on the interaction between computers and human language. In the field of machine learning, NLP has driven notable advancements, especially in areas such as text translation, sentimental anlysis, and largely in text generation (4). The latter, text generation, is an application of NLP that has found a variety of uses in various sectors such as medicine, law, and education, marking a milestone in the way humans interact with information.

In the process of NLP evolution, there are two particulary consistent models. The first is the deep learning model known as Recurrent Neural Networks (RNN), and more especially, the Long Short-Term Memory (LSTM) variant (10). LSTMs have a significantly improved the handling of context and sequence in texts, allowing the model to capture information in extensive data sequences.

The second model is based on Transformers and attention mechanism, first introduced in the paper "Attention Is All You Need" (14). This model marked a milestone in NLP, especially through its ability to weight the relative importance of the content from different parts of the text. Transformers, and in particular the Bidirectional and Auto-Regressive Transformer (BART) model (5), have proven to be exceptionally effective in text generation tasks, surpassing all previous architectures in terms of accuracy and coherence.

The relevance of these models in text generation, and more specifically in the creation of titles from abstracts in papers, constitutes the core of this study. Our search seeks to evaluate and compare the performance and efficacy of LSTM and fine-tuned version of BART, in the specific task of analysis and generation on short texts.

Purpose of the Paper

The main purpose of this paper is to conduct a comparison between the two prominent models in the field of natural language processing (NLP): Long Short-Term Memory (LSTM) and Bidirectional and Auto-Regressive Transformer (BART). The focus of the comparison between these two model is to evaluate their performance in a specific task, text summarization for title generation.

Final Paper for the ML3061-2023.2 - Machine Learning Class at UTEC. Lima, Peru.

This research aims to explore the extent to which each model is capable of handling the complexity and nuances of human language in the context of text summarization. By comparing LSTM and BART, it's expected to find how the different NLP architectures influences the results of specific tasks.

Model Overview

LSTM

Long Short-Term Memory networks, known as LSTM (3), represent an improvement over Recurrent Neural Networks (RNNs), specifically designed to address limitations in managing long-term dependencies. These networks, introduced in 1997, represent an evolution in the field of natural language processing. The LSTM architecture is distinguished from traditional RNNs by incorporating memory cells, equipped with input, forget, and output gates, which regulate the flow of information. This structure allows LSTM models to retain relevant information and discard unnecessary information, improving the neural network's ability to learn from extensive and complex data sequences.

BART

The Bidirectional and Auto-Regressive Transformer (BART) model represents a significant advancement in the field of transformer models in natural language processing. This model successfully integrates the efficacy of two approaches: bidirectional encoding, similar to BERT (2), and autoregressive generation, characteristic of models like GPT (11). This combination of features allows the model to effectively understand the context of a text while enabling it to coherently generate new content.

Evaluation Metrics

Evaluating the results produced by LSTM and BART models is an important aspect of the study. The generation of effective titles demands that the sentences be complete, coherent, and closely related to the content of the source text. The goal is then to ensure that the results are accurate, cohesive, and relevant. To carry out this evaluation, three standardized metrics in the field of natural language processing will be used: BLEU, ROUGE, and BERTScore.

BLEU

BLEU (9) is a quantitative evaluation metric, primarily used to measure the quality of machine translation and, by extension, text generation. The BLEU metric compares phrases generated by a model with a set of 'references', thereby calculating its score based on the matching of n-grams between them.

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

- c: is the length of the generated text.
- r: is the length of the reference text.

$$BLUE = BP \cdot exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

- p_n: is the precision of n-grams, which is the proportion of n-grams in the generated text that appear in the references
- w_n: are the weights assigned to each n-gram precision (generally, the same weight is given to each).
- N: is the maximum size of n-grams used (up to 4-grams are often used).

- BP: is the Brevity Penalty factor, which penalizes generated texts that are too short compared to the references.

ROUGE

ROUGE (6) is a metric whose main objective is to evaluate the quality of automatically generated summaries, in the context of text summarization task and tex generation. The ROUGE-n variant assesses the similarity at the level of n-grams between the generated text and a set of reference texts. The comparison of n-grams measures the level of matching of sequences of n words between the generated text and the reference text.

$$\text{Rouge-N} = \frac{\sum_{S \in \{\text{ReferenceSummaries}\}} \sum_{\text{gram}_n \in S} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \{\text{ReferenceSummaries}\}} \sum_{\text{gram}_n \in S} \text{Count}(\text{gram}_n)}$$

BERTScore

BERTScore (15) is a text evaluation metric that uses contextualized language models, like BERT, to assess the quality of generated texts. This metric is used in applications such as machine translation and text summarization, where quality goes beyond the literal matching of words, but rather in their context. BERTScore compares the word embeddings of the generated text with those of a reference text. It uses a pre-trained BERT model to obtain the embeddings, capturing the semantic context of each word within the text. The cosine similarity between these embeddings is calculated for each word in the generated text, identifying the corresponding word in the reference text that shows the greatest similarity. This process is carried out in both directions, from the generated text to the reference text and vice versa, providing measures of text quality in semantic and contextual similarity.

$$BERTScore = \frac{1}{|c|} \sum_{c \in C} max_{r \in R} \cos\left(BERT_c, BERT_r\right)$$

- C: represents each token in the generated text.
- R: represents each token in the reference text.
- $\cos$: It is the cosine similarity between the BERT embeddings of the tokens.
- $BERT_c$ $BERT_r$: they are the BERT embeddings for the tokens of the generated text and the reference text, respectively.

Experimentation

Dataset

The dataset that we have used for the experiments and to train the models is called *Papers by subject*. It contains a number of 52399 papers and their metadata, within it is the Title (input) and Abstract (output).

LSTM

Different architectures using LSTM cells were tested until acceptable results were achieved. We start from the Sequence-to-Sequence based (12) architecture since what we want to solve is to generate text by receiving text as input. The architecture, within the changes that have been made, remained with constant components throughout the modifications: tokenizer and loss parameters. Regarding the tokenizer we used, it only assigns a number to each word in the input, resulting in a vector of integers. After the tokenizer, the vector is padded up to the maximum length of the Abstract. On the other hand, the Sparse Categorical Cross-entropy (13) has been used as a loss function because of the way our words (classes) are categorized.

The iterated part of the architecture was based on seq2seq, we started initially with an Embedding layer of 200 dimensions, an encoder and decoder with 3 LSTM layers of LSTM (1), each with 300

dimensions and an Attention layer (14). The results were very bad for this design, generating only the same title for any input. Because of this and the small amount of datapoints we have, we decided to reduce the parameters and simplify the architecture. The result of several iterations is the following architecture (1), with which the experiments and comparisons were made.

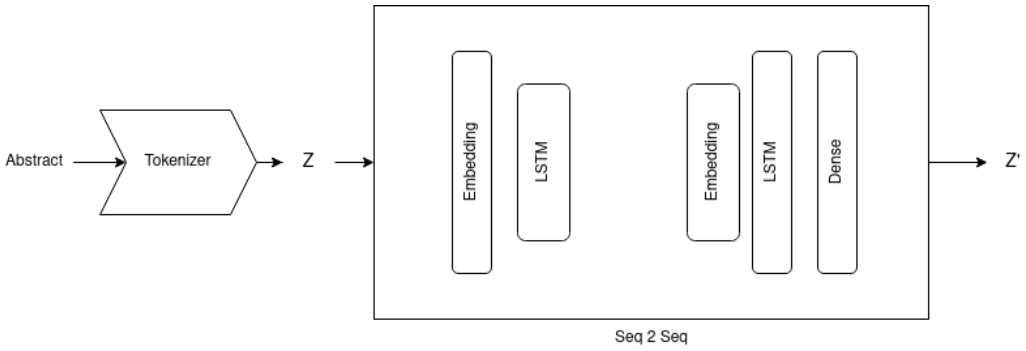

Figure 1: LSTM Model: Latent Space dimensions = 100, Embedded Space dimensions = 50

BART

A fine-tuning process was applied to the entire BART(5) model, which compared to partial training methods it optimally adapts its pre-trained capabilities to our dataset, hence improving the performance. Hyperparams as the optimizer, loss function, and the learning rate scheduler were set to achieve this. The optimizer employed was the AdamW(7) optimizer, capitalizing on its adaptive learning rate and weight decay for model generalization and to prevent overfitting. The Cross-Entropy Loss function(8) was utilized, because of the multi-class nature of predicting the next word in the sequence of our title generation task. Finally a Linear Learning Rate Scheduler was implemented to stabilize training and optimize convergence.

Three experiments were done by selecting different number of epochs. First, the model was trained on one epoch by which the generated titles didn't show an extend in BARD's vocabulary, thence the predicted titles were too similar to the original ones. Six epochs were used in the second experiment, the model's inferences had titles which encapsulated the idea of the abstract with a different vocabulary and remained consistent. The third experiment utilized eight epochs by which the model didn't show an increase in the quality of predictions. The second fine-tuned model showed the best overall BERT(15) Score and produced titles which made sense in the context of the abstract.

Results

The predictions made by the models were tested on four types of inputs:

- Clean Input: The original abstract from the papers without any alterations.
- Words shuffled input: The abstract with all the words shuffled.
- Sentence shuffled input: The abstract with all the sentences shuffled.
- Noisy input: The abstract with some altered and erased words.

Based on this, the displayed results are for the fine-tuned BART model and the LSTM model. The scores were obtained by making inferences over the four types of inputs in the test set and averaging the results for each one.

The scores that appear in table 1 show that the BART model performs well with clean inputs and with sentence shuffled inputs obtaining a precision score greater than 0.85 and overall high scores across all metrics. This suggests that the model could associate abstracts with the correct structure in the sentences rather than their order when generating a title.

Table 1: Fine-tuned BART Model Metrics Result

INPUT	BERT Score	Bleu Score	ROUGE-1	ROUGE-2	ROUGE-L
Clean Input	0.8639/0.8664/0.8649	0.0857	0.3811	0.2195	0.3495
Words shuffled input	8.44e-01/8.46e-01/8.45e-01	7.11e-232	2.39e-01	2.43e-02	1.91e-01
Sentence shuffled input	0.8833/0.8848/0.8838	0.0614	0.4595	0.2528	0.4185
Noisy input	8.46e-01/8.64e-01/8.54e-01	2.22e-79	3.50e-01	1.73e-01	3.42e-01

When words are shuffled, the performance drops significantly, as evidenced by lower scores. This indicates that the model relies heavily on the sequential structure of language to generate coherent titles.

The lowest scores are observed with noisy inputs, meaning the model struggles with abstracts that contain misspellings or grammatical errors. This highlights the importance of clean, well-structured data for training and inference in NLP models.

Results demonstrate that while BART is robust when handling well-formed sentences, its performance degrades with poor input quality, emphasizing the need for clean data to maintain high-quality output.

Table 2: LSTM Model Metrics Result

INPUT	BERT Score	Bleu Score	ROUGE-1	ROUGE-2	ROUGE-L
Clean Input	0.8664/0.8359/0.8508	0.0	0.1140	0.0382	0.114
Words shuffled input	0.8618/0.8284/0.8446	0.0	0.0940	0.0382	0.0940
Sentence shuffled input	0.8643/0.8307/0.8469	0.0	0.0994	0.02	0.0994
Noisy input	0.8572/0.8295/0.8429	0.0	0.1378	0.0382	0.1378

The LSTM has scored reasonably well on the BERTScore metric across different input scenarios, which suggests that the generated titles are semantically similar to the reference titles to a certain extent. The BERTScore is more focused on the semantic content than the exact wording or order, which may explain why it captures the theme of the abstracts even if the exact phrasing isn't matched.

However, when we look at the BLEU scores and some of the ROUGE metrics, particularly ROUGE-2, which focus more on the exact word sequences and bigram matches, the scores are significantly lower. This indicates that while the LSTM is capturing the semantic essence well (as reflected by BERTScore), it is not as successful in generating syntactically coherent or lexically precise titles. The BLEU score being zero for shuffled inputs suggests that the LSTM model completely fails to recreate any meaningful n-gram overlaps with the reference titles under these conditions.

The titles generated by the LSTM from all types of inputs seem to align more with the gist of the abstract but is not directly aligned with the reference titles, reflecting a lower precision in word choice and order. This could be due to the inherent limitations of LSTM networks in capturing long-range dependencies and complex patterns within the text.

Table 3: Bart and LSTM Inferences

INPUT	BART	LSTM
Clean Input	Functional Linear Regression using Functional Principal Component Analysis	sparse estimation via bayesian inference
Words shuffled input	Asymptotic pressure predictors of functional longitudinal nonincreasing construct from data bands trajectories	bayesian inference multivariate data
Sentence shuffled input	Functional Principal Component Analysis for Functional Linear Regression	bayesian inference semiparametric regression models
Noisy input	Functional Linear Regression using Functional Principal Component Anealysis	sparse estimation linear models

The table 3 showcases a comparison of title generation from abstracts by BART and LSTM models under different input conditions. BART consistently produces more coherent titles closely aligned with the clean input and when the sentences are shuffled. Contrary to this, the output quality decreases with noisy input as the last word of the title is generated as Anealysis instead of Analysis. In contrast, LSTM's outputs diverge more significantly from the expected titles, the generated title shows words related to the original abstract, but does not form a coherent title.

Conclusions

In conclusion, our comparative analysis shows that BART outperforms LSTM in generating coherent and contextually relevant titles from abstracts. BART's architecture, which effectively captures the structure of language, allows it to generate titles that are not only semantically aligned with the abstracts but also maintain the syntactic integrity of the language.

While LSTM can infer relevant words within the context of the abstracts, it falls short in crafting concise titles with the same level of coherence. This is likely due to the LSTM's limitations in modeling longer-term dependencies and more complex sentence structures.

The BART-generated models demonstrate a better understanding of word order rather than just sentence structure, enabling them to produce titles that are more grammatically and contextually accurate.

References

[1] CHO, K., VAN MERRIËNBOER, B., GULCEHRE, C., BAHDANAU, D., BOUGARES, F., SCHWENK, H., AND BENGIO, Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Doha, Qatar, Oct. 2014), A. Moschitti, B. Pang, and W. Daelemans, Eds., Association for Computational Linguistics, pp. 1724–1734.

[2] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[3] HOCHREITER, S., AND SCHMIDHUBER, J. Long Short-Term Memory. *Neural Computation 9*, 8 (11 1997), 1735–1780.

[4] JOHN BASHA, M., VIJAYAKUMAR, S., JAYASHANKARI, J., ALAWADI, AHMED HUSSEIN, AND DURDONA, PULATOVA. Advancements in natural language processing for text understanding. *E3S Web Conf. 399* (2023), 04031.

[5] LEWIS, M., LIU, Y., GOYAL, N., GHAZVININEJAD, M., MOHAMED, A., LEVY, O., STOYANOV, V., AND ZETTLEMOYER, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[6] LIN, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out* (Barcelona, Spain, July 2004), Association for Computational Linguistics, pp. 74–81.

[7] LOSHCHILOV, I., AND HUTTER, F. Decoupled weight decay regularization, 2019.

[8] MAO, A., MOHRI, M., AND ZHONG, Y. Cross-entropy loss functions: Theoretical analysis and applications, 2023.

[9] PAPINENI, K., ROUKOS, S., WARD, T., AND ZHU, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (Philadelphia, Pennsylvania, USA, July 2002), P. Isabelle, E. Charniak, and D. Lin, Eds., Association for Computational Linguistics, pp. 311–318.

[10] PAWADE, D., SAKHAPARA, A., JAIN, M., JAIN, N., AND GADA, K. Story scrambler - automatic text generation using word level rnn-lstm. *International Journal of Information Technology and Computer Science 10* (06 2018), 44–53.

[11] RADFORD, A., WU, J., CHILD, R., LUAN, D., AMODEI, D., AND SUTSKEVER, I. Language models are unsupervised multitask learners.

[12] SUTSKEVER, I., VINYALS, O., AND LE, Q. V. Sequence to sequence learning with neural networks, 2014.

[13] TERVEN, J., CORDOVA-ESPARZA, D. M., RAMIREZ-PEDRAZA, A., AND CHAVEZ-URBIOLA, E. A. Loss functions and metrics in deep learning, 2023.

[14] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L., AND POLOSUKHIN, I. Attention is all you need, 2023.

[15] ZHANG, T., KISHORE, V., WU, F., WEINBERGER, K. Q., AND ARTZI, Y. Bertscore: Evaluating text generation with bert, 2020.