University of Cape Town

Independent Research in Computer Science

Efficient Transformer Language Models

Author:
Jaedon Van Schalkwyk

Supervisor:
Dr Jan Buys

Research Proposal for course CSC2005Z
for the degree of Bachelor of Science

September 2, 2022

Contents

1 Project Description

Natural Language Processing (NLP) is a branch of artificial intelligence that helps computers understand, interpret, and manipulate human language. Language models are used in NLP applications, and focus on using statistical and probability models to determine the sequence of words occurring in a sentence based on the previous words (Chaudhuri 2022). One primary area of use for such a model is text generation.

The main focus of this paper will be in evaluating language models' text generation ability, measured through accuracy and speed.

In recent years the advancement in Language Models and NLP has been synonymous with the betterment of the Transformer architecture. Transformers are a neural network architecture, first proposed by Vaswani et al. 2017, that are used mainly for processing long sequences such as text passages. These models make use of an encoder-decoder (sequence-to-sequence) structure that takes words/sentences as input and produces words/sentences as output. They are especially useful in NLP as they are able to effectively learn from large text corpora and transfer over that knowledge to practical applications.

However, Transformers can be computationally expensive, limiting the length of sequences they are able to process. This is due to a key mechanism that they employ: attention. Attention is a mechanism that compares the words (tokens) in a sequence to other tokens in the sequence, generating relation and context between them. In the original Transformer proposed by Vaswani et al. 2017, a full attention mechanism is used, where every token attends to every other token, resulting in a *quadratic* complexity.

This paper aims to compare the original Transformer, to two newly proposed modification, namely: BigBird (Zaheer et al. 2020), and Longformer (Beltagy et al. 2020). Both of these models use different versions of a modified attention mechanism, Sparse Attention, which performs with linear time complexity. These models improve on the inefficiencies of the original Transformer, allowing longer sequences to be processed.

2 Research Questions

This research aims to compare the performance of newer Transformer architectures to the original Transformer, determining the differences in performance. Moreover, this research focuses on low-resource language evaluation, conducting all training and testing using South African Nguni languages.

Specifically, the research aims to address the following:

1. ***How does BigBird (Zaheer et al. 2020) compare to a standard Transformer (Vaswani et al. 2017) in terms of both speed and accuracy[1] ?***

[1] Accuracy measured by testing the bits per character (see section 4)

2. ***How does BigBird compare to the Longformer (Beltagy et al. 2020) architecture in terms of both speed and accuracy[1] ?***

3. ***Time permitting – What is the effect of the amount of training data on the models' performance?***

The contribution of this research will not only provide a clearer scope into the improvements made by newer Transformer models and the sparse attention mechanism, but also demonstrate the efficacy of Transformer models on low resource language applications.

3 Methodology

3.1 Attention

Attention is a function that maps a query to a set of key-value pairs to produce an output, where the query ($\mathbf{Q}$), key ($\mathbf{K}$), value ($\mathbf{V}$), and output are all vectors (Vaswani et al. 2017). The output is the softmax[2] of single matrix multiplication $\mathbf{Q}\mathbf{K}^T$ divided by the square root of their dimensionality, multiplied by $\mathbf{V}$ (Jurafsky et al. 2022). This can be seen in equation 1.

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V} \tag{1}$$

The attention function produces a weighted sum of values, whereby important relation and context between certain words are mapped out. This allows the Transformer model to determine which tokens are more meaningful than others (in defining the overall meaning of the input sequence), allowing it to perform well in all NLP applications.

3.1.1 Full attention

The original Transformer implemented a full attention mechanism (shown in Figure 1), whereby each token was attended to by every other token in the sequence. While allowing for a complete comparison for each token, this approach performs with a quadratic complexity, posing limitations on the length of sequence the model can process.

3.1.2 Sparse Attention - BigBird

A solution to this problem was proposed by Zaheer et al. 2020 in the BigBird Transformer architecture, implementing a sparse attention mechanism. This modified approach differs from the original attention mechanism, proposing an amalgamation of: Random (tokens randomly attend to other tokens), Window (tokens attend to the tokens directly next to them), and Global attention (tokens always attend to a set of predetermined tokens), illustrated in Figure 2.

This attention mechanism allows BigBird to perform with linear complexity, and thus improves over the standard Transformer's full attention mechanism.

[2] A function that converts a vector of size N into a probability distribution of N outcomes

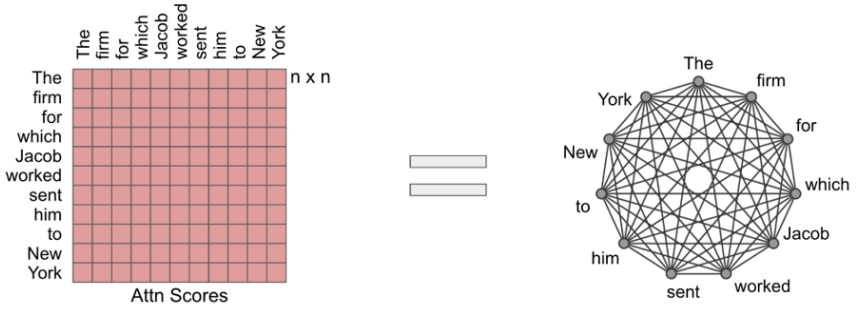

Figure 1: Standard Transformer's Full Attention

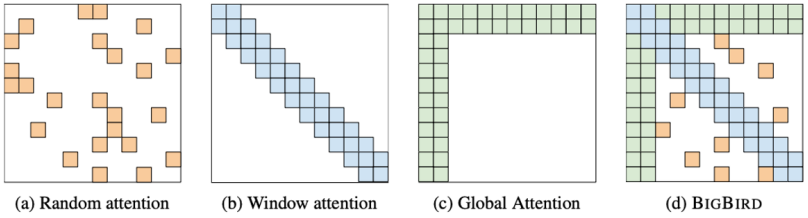

Figure 2: BigBird's Sparse Attention

3.1.3 Sparse Attention - Longformer

Similarly to BigBird, the attention mechanism proposed by Beltagy et al. 2020 in the Longformer architecture also performs with linear complexity. The implementation of attention here focuses on: Sliding Window (same as window), Dilated Sliding Window (tokens attend to tokens next to them, in a wider range), and Global attention (see Figure 3). The performance benefits of this architecture are most apparent for long input sequences.

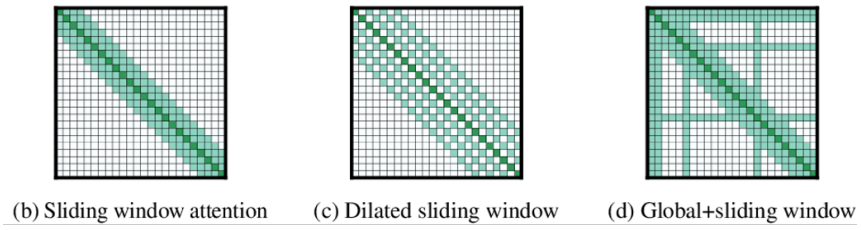

Figure 3: Longformer's Sparse Attention

3.2 Experimentation

Background research will be conducted in order to gain a full understanding of all the architectures and systems needed. Further research and experimentation with PyTorch will then be done. Using these tools, the relevant models will be trained on a high performance cluster, using pre-screened data provided by Mesham et al. 2021. The initial training will be done on the original Transformer, followed by BigBird, and finally completed on Longformer.

Once the training is complete, testing will commence. The testing will encompass evaluating the trained Transformer models on the provided dataset, computing the test set perplexity of the model, and finally comparing the outputs against a reserved section from the dataset (specifically reserved for testing purposes). The findings will then be collected and analyzed. Finally, a final report will be drafted detailing all the findings.

This experimental strategy is optimal for this research as it allows for all the architectures to be trained, tested, evaluated, and compared fairly.

4 Evalution of Research Questions

The quality of language models can be evaluated both extrinsically and intrinsically (Mesham et al. 2021). Extrinsic evaluation measures the model against end-to-end tasks such as question-answering or machine translation. Intrinsic evaluation however, evaluates the model using statistical measures to assess quality. This paper will make use of intrinsic testing, more specifically on the evaluation metrics related to perplexity.

Perplexity can be thought of as the models ability to predict uniformly among the set of specified tokens in a corpus (Jurafsky et al. 2022). It is the exponentiated average negative-log likelihood of a sequence, calculated with exponent base 'e', as defined below.

$$PPL(X) = exp\{-\frac{1}{t}\sum_{i}^{t} log(p_\theta)(x_i|x_{<i})\} \tag{2}$$

The vocabulary of the architecture and tokenization approach thus has an implication when evaluating using perplexity. In this research all architectures will have the same vocabulary, as well as tokenization strategy, therefore making perplexity a suitable choice of evaluation.

Furthermore, an evaluation metric of bits per character (BPC) will be used. BPC is used as a measure of perplexity normalised by character length of text. Moreover, to ensure optimal performance, a sliding window strategy will be used during evaluation. This entails choosing an appropriate window size to slide across the sequence of tokens for evaluation, avoiding the limits each model has on the length of sequence it can process. This ensures optimal execution, while still giving the architectures a sufficiently large context to evaluate off of.

5 Work Details

5.1 Risks

The following risks have been identified, as well as their respective mitigation strategies:

Risk	Effect	Mitigation Strategy
Data Loss	All data and results will be lost, forcing the research to be restarted	Store constant backups both locally and on the cloud
Lack of suitable hardware	Training and testing will be too time intensive, possibly leading to an incompletion of research	Access to high performance clusters will be provided
Sensitive / Inappropriate data in training set	Architectures will be trained to have inappropriate bias, and could even leak sensitive and/or confidential information	The dataset will be pre-screened and verified before being used as training input

Table 1: Risks and Mitigation Strategies

5.2 Ethical Issues

This research has no major ethical issues. No human testing or experimentation will be done, with all training and testing processes conducted by the author and the supervisor only. Moreover, the data that is used for testing and training has been pre-screened before implementation, and as such no sensitive, inappropriate, or personal information will be passed into the various Transformer architectures.

5.3 Timeline

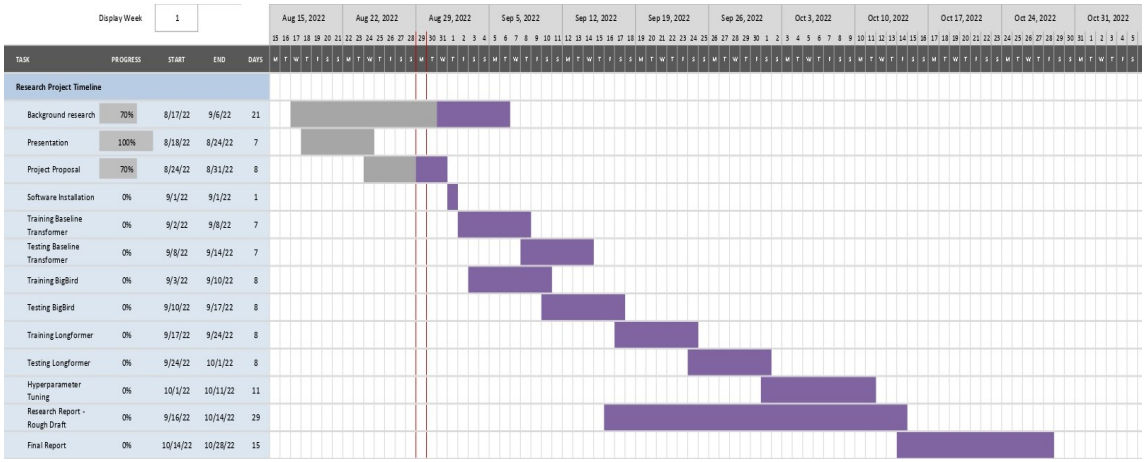

Figure 4: Project Timeline

5.4 Resources required

- Language Datasets
  - The datasets used focus on South African languages, specifically isiZulu and Sepedi
  - The two text corpora sources come from: the **National Centre for Human Language Technology (NCHLT)** text project (Eiselen et al. 2014), as well as news articles from the isiZulu **Isolezwe** newspaper.
  - The datasets have been preprocessed and normalized, and will be split into training, validation, and testing subsets following a 80/10/10 split
  - The size of the datasets for each language (isiZulu and Sepedi) roughly amounts to one million words each
- High-Performance Cluster
- Python
  - PyTorch
  - Matplotlib

5.5 Deliverables

- Project Proposal
- Final Research Paper

5.6 Milestones

- Research Proposal - August 2022
- Software installed and ready - Beginning September 2022
- Baseline Transformer Trained - Beginning September 2022
- Baseline Transformer Tested - Mid-September 2022
- BigBird Trained - Beginning September 2022
- BigBird Tested - Mid-September 2022
- Longformer Trained - Mid-September 2022
- Longformer Tested - September 2022
- Hyperparameter Training - Beginning October 2022
- Draft and Final report compiled - October 2022

6 Anticipated Outcomes

After all training, testing, and evaluation is carried out, the following outcomes are expected:

1. BigBird outperforms the standard Transformer architecture, demonstrating a faster execution time as well as a higher accuracy for Language Modelling
2. BigBird and Longformer perform similarly, with BigBird performing faster on shorter input sequences, and Longformer performing faster on longer sequences. Both models will have a similar accuracy and speed
3. Time permitting: Newer architectures (BigBird and Longformer) perform better than the standard Transformer when trained on less data

Bibliography and Previous Systems

Eiselen, Roald and Martin Puttkammer (May 2014). "Developing Text Resources for Ten South African Languages". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), pp. 3698–3703. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/1151_Paper.pdf.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). *Attention Is All You Need*. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.

Beltagy, Iz, Matthew E. Peters, Arman Cohan, and (2020). *Longformer: The Long-Document Transformer*. DOI: 10.48550/ARXIV.2004.05150. URL: https://arxiv.org/abs/2004.05150.

Zaheer, Manzil, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed (2020). "Big Bird: Transformers for Longer Sequences". In: DOI: 10.48550/ARXIV.2007.14062. URL: https://arxiv.org/abs/2007.14062.

Mesham, Stuart, Luc Hayward, Jared Shapiro, and Jan Buys (2021). *Low-Resource Language Modelling of South African Languages*. DOI: 10.48550/ARXIV.2104.00772. URL: https://arxiv.org/abs/2104.00772.

Chaudhuri, Koushiki Dasgupta (Jan. 2022). *Building language models in NLP*. URL: https://www.analyticsvidhya.com/blog/2022/01/building-language-models-in-nlp/#:~:text=A%20language%20model%20in%20NLP,appear%20next%20in%20the%20sentence..

Jurafsky, Dan and James H. Martin (2022). "Evaluating Language Models". In: *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson.