Exploring Clustering Alternatives To Attention

Tristan Peat
tpeat@gatech.edu

Sophia Imhof
simhof3@gatech.edu

Meredith Rush
mrush30@gatech.edu

1 Introduction & Background

The machine learning field is moving towards using Transformers for a wide variety of tasks (Vaswani et al., 2023). Transformers have been shown to be extremely successful when scaling to compute and dataset size (Zhai et al., 2022). At a high level, transformers use the scaled dot product attention mechanism to compute relationship scores between all vectors in the embedding space, resulting in quadratic time complexity. We propose a new model ECATA (**E**xploring **C**lustering **A**laternatives **t**o **A**ttention) that provides an alternative the attention mechanism bottleneck of Transformers. This substitution will be done by implementing traditional clustering methods to group similar vectors together in the embedding space.

Transformers have been instrumental to setting state of the art benchmarks for natural language processing (NLP) tasks over the past few years. For instance, pre-trained language model, BERT, released by GoogleAI in 2018, surpassed eleven NLP benchmarks (Devlin et al., 2019) upon its release. Transformer architecture is widely used in other popular large language models such as the GPT series (Radford et al., 2019).

As these models improve in performance, they are also increasing in size and thus computational intensity. These transformers rely on the attention mechanism, a quadratic time complexity that becomes increasingly expensive with model size. Thus, there has been a large push in research towards ways to optimize transformer performance. Various solutions in this area include reducing the time complexity of the the attention mechanism through techniques like Sliding Window attention(Beltagy et al., 2020) and patch attention (Cheng et al., 2022). There has also been memory speedups and memory efficiency improvements such as FlashAttention (Dao et al., 2022), FlashAttention-2 (Dao, 2023) and PagedAttention (Kwon et al., 2023).

2 Motivation

The computational efficiency of Transformer models is a critical factor to be able to deploy these models on mobile devices. The limitations of compute and storage on most mobile devices prevent inference operations to be run in a reasonable time (Li et al., 2024). Advancements in transformer architectures have focused on making small tweaks in complexity or memory while maintaining performance. We seek to continue this research and explore the gap in performance of a transformer model if we completely replace the most computationally expensive part of the transformer, the attention mechanism, with traditional clustering methods. Traditional clustering methods, such as K-means, require less computational power than the attention mechanism. By replacing the attention mechanism with a clustering method, we aim to reduce the computational power required to run a Transformer model. By examining the trade offs between model efficiency and performance through ECATA, we hope to provide new insights and potential areas of exploration for the future of transformer optimization.

3 Problem Statement

We hypothesize that traditional clustering methods can replace the most inefficient parts of the transformer architecture, with some marginal performance trade offs. We want to explore how much the performance drops and compare it against other transformer optimizations.

While this is a heavily saturated research area, the originality of our research spawns from our novel approach to replacing the attention blocks of the transformer completely and analyzing performance.

4 Methodology

We will implement 3 versions of a language translation model with: baseline LSTM, baseline transformer and ECATA. We will benchmark each model on a simple language translation task using an English to French Translation dataset from the Tatoeba Project (Kelly, 2020). This dataset consists of roughly 200,000 English words, phrases, and sentences along with their corresponding French translation. The data preprocessing and tokenization for each model will remain consistent. The transformer will be initialized with pre-trained weights to accelerate the time consuming task of training from random starting weights.

First, we will train the baseline LSTM and modern transformer model (eg. BERT) with the English to French Translation dataset. To create the models, we will leverage third-party libraries, such as PyTorch. Then, we will create and train ECATA. To create ECATA, we plan to build a transformer-like architecture and replace some or all of the multi-head attention blocks with clustering algorithms. We want to experiment with various clustering techniques, such as K-means and DBSCAN algorithms, which have a time complexity less than $O(n^2)$. Our methodology will be further refined after completing the our first project task of exploration of clustering techniques.

The ideal outcome of our project would be marginal or no loss in performance accuracy of our model, ECATA, with a decrease the runtime and/or memory usage when compared to our baseline implementations. This outcome would demonstrate promising opportunities for future research in leveraging clustering techniques in place of attention mechanisms within Transformers.

5 Potential Results and Comparisons

To analyze and compare ECATA results against the baseline LSTM and Transformer model, our team will calculate the BLEU (bilingual evaluation understudy) score of each model on our dataset. This is a popular evaluation metric for machine translation tasks that outputs values in a range of 0 to 1 with values closer to 1 indicating higher translation accuracy. We will also evaluate model runtime and model memory usage for comparisons.

With most machine learning models, there is a trade off between performance and time complexity. Our team expects to see a decrease in performance when we remove components of the transformer and replace them with clustering methods. We believe that with some engineering effort and theoretical guidance, we can present empirical and theoretical bounds for the impact that clustering can make. Potentially, there will be a case for further study down this avenue of using unique architectures instead of building dependence on the transformer.

6 Feasibility Analysis

We chose this project design to stray away from large models, which are dominant in the NLP field today. We also wanted the ability to iterate more frequently on smaller experiments, rather than be constrained by time or compute. Creating a more efficient transformer model is a widely researched topic that could have a significant impact on the future of large language models and NLP. The most difficult task for this project will be replacing multi-head attention blocks with clustering methods in our transformer model, ECATA. Our team is anticipating this roadblock and will spend a large portion of time researching previous successful attempts at replacing transformer blocks. Additionally, our team anticipates the possibility of a roadblock in the mathematical theory behind ECATA. We have access to a Ph.D. candidate at the Georgia Institute of Technology who will be able to assist us in the event of a theoretical roadblock.

We do not anticipate roadblocks with creating and training the baseline LSTM and transformer models. There are many resources, such as PyTorch and HuggingFace, to assist the creation and training of these models. Lastly, our model should theoretically reduce the amount of compute time and space from the typical transformer. Our team does not anticipate roadblocks relating to compute time or space for any of the models we are creating and training.

A rough timeline for the project is as follows: a) In 2-3 weeks, implement the three models mentioned earlier and spend time surveying the literature for high dimensional clustering or unique approaches to replacing transformer blocks b) In 3-4 weeks, design and conduct several experiments, taking out pieces from pretrained transformers and replacing them with our custom clustering blocks c) In 1-2 weeks, summarize and report our findings.

7 Expanded Literature Review

7.1 Improvements of Transformer Efficiency

Prior to the transformer model, recurrent neural-networks (RNN) produced state-of-the-art results for machine translation tasks. RNNs perform tasks through sequential processing, which requires that previous computations are completed the next computation. Sequential processing limits parallelization of RNNs, which increases the amount of time to train these models. The transformer model was created to increase parallelization and accuracy of generative tasks. To accomplish this, the transformer relies on multi-head self-attention mechanisms, which require quadratic time to compute (Vaswani et al., 2023).

There have been many attempts to reduce the computational cost of the transformer to linear time. One method called Sliding Window Attention is introduced by the LongFormer model, created to address the limitations of transformers in processing lengthy documents (Beltagy et al., 2020). The sliding window method only pays attention a fixed size window around a given token, allowing it to scale linearly rather than quadratically. However, limitations include that sliding window attention is not flexible in learning task-specific representations.

Another method called Linear Attention replaces the traditional softmax attention with a linear approach(Katharopoulos et al., 2020). A feature map is used to transform queries and keys, and attention is computed through the dot product with the values, reducing complexity to O(n). While the research notes a significant speedup in performance, the effectiveness is dependent on the choice of feature map and the loss of expressiveness through linear attention can cause a drop in performance for certain tasks.

7.2 Clustering similar vectors within Q, K, and V

The bipartite matching algorithm from Token Merging (ToMe) can be viewed as a form of constrained-clustering (Bolya et al., 2023). In traditional clustering, a group of data points (tokens, in this case) are partitioned into clusters based on some similarity metric. Bipartite matching in ToM operates similarly but with more control over how tokens are combined, introducing a structure that avoids the problems of arbitrary or overly large clusters. Bipartite matching splits the set of tokens into two disjoint sets, A and B, with each set roughly equal in size. This is similar to forming two clusters but ensures that tokens are paired across these sets rather than allowing arbitrary cluster sizes. Tokens in set A are connected to tokens in set B based on a similarity metric, such as cosine similarity on the key vectors from the self-attention mechanism.

7.3 Clustering highly-correlated Q and K matrices

The Clustered Head Attention (CHAI) model takes advantage of high redundancy between self-attention heads by combining similar heads through K-Means. CHAI resulted in a 5x speedup in next token generation for NLP tasks compared to models leveraging multi-head self-attention without clustering (Agarwal et al., 2024). Another model focuses on clustering query matrices and performing self-attention only on the centroids of the clusters (Vyas et al., 2020). This clustered attention approach resulted in a 2x speedup and demonstrated competitive performance when compared to models using "full" attention when trained under minimal computational resources. However, this research notes that the model has significantly worse performance on tasks that require complex attention patterns, as well as having a slower inference time on some tasks.

7.4 Mathematical Support for Clustering Intuition

Self-attention lies at the heart of transformer and can be represented by the following equations:

$$\alpha_{i,j}^{(h)} = \text{softmax}_j\left(\frac{\langle Q^{(h)}(x_i), K^{(h)}(x_j)\rangle}{\sqrt{k}}\right)$$

$$u_i' = \sum_{h=1}^{H} W_{c,h}^T \sum_{j=1}^{n} \alpha_{i,j}^{(h)} V^{(h)}(x_j)$$

where Q, K, V are each parameterized by $W_{i,h} \in \mathbb{R}^{k\times d}$ and h represents an attention head. The self-attention block is succeeded by a residual connection and LayerNorm, followed by an MLP with ReLU activation, and another residual connection and LayerNorm layer. Previous research highlights the importance of each of these components for the effectiveness of self-attention, particularly the skip connections, which play a key role in preventing rank collapse in the attention matrix (Dong et al., 2023).

It is reasonable to assume that "clusters" in Transformers do not emerge immediately, whether in

terms of training epochs or layer depth, particularly when dealing with complex inputs. Otherwise, clustering could be used universally for all NLP tasks. Therefore, we aim to analyze the asymptotic behavior of a time-dependent self-attention mechanism. We parameterize the operation as:

$$\alpha_{i,j}^{(h)}(t) = \text{softmax}_j\left(\frac{\langle Q^{(h)}(x_i,t), K^{(h)}(x_j,t)\rangle}{\sqrt{k}}\right)$$

Our goal is to study $\alpha_{i,j}^{(h)}(t)$ alongside $W_{h,q}(t), W_{h,k}(t), W_{h,v}(t)$ as $t \to \infty$, to further explore the clustering behavior:

$$\lim_{t\to\infty} \alpha_{i,j}^{(h)}(t) = \text{softmax}_j\left(\frac{\langle Q^{(h)}(x_i,t), K^{(h)}(x_j,t)\rangle}{\sqrt{k}}\right)$$

We hypothesize that as training progresses, similar tokens become more alike, clustering together, while dissimilar tokens diverge. Mathematically, this can be expressed as a limit:

$$\lim_{t\to\infty} \alpha_{i,j}(t) = \begin{cases} 1 & \text{if } i,j \in C \\ 0 & \text{if } i,j \notin C \end{cases}$$

where C denotes a cluster.

This idea stems from the observation that the dot product emphasizes similar vectors, and the softmax function ensures that the elements of each row sum to 1, forming a valid probability distribution. However, Geshkovski et al. (2024a) provides the theoretical foundation for this by examining the asymptotic behavior of attention through the lens of a dynamic particle system. Extending the work of Dong et al. (2023), they demonstrate that, with skip connections in place, rich geometric clusters naturally emerge. Their analysis further suggests that certain tokens, termed leaders, attract the most attention from neighboring tokens, pulling them into their vicinity (Geshkovski et al., 2024b).

The hypothesis that clusters may form within attention blocks as $t \to \infty$ is closely related to the Neural Collapse (NC) phenomenon, observed in classification tasks, where feature vectors in the final layer converge to the class mean as $t \to \infty$ (Papyan et al., 2020). Wu and Papyan (2024) further investigates this concept in the context of large language models, focusing on token prediction tasks with Transformer architectures, showing that NC occurs as model size and training scale. To explore this in our context, we can design an experiment to track the same metrics—global and intra-class variance—over the course of training to observe NC.

8 Detailed Plan, Milestones, and Timeline

Our project timeline with weekly milestones is broken down in Table 1.

8.1 Experiments

It is important to recognize that there are multiple avenues to explore when investigating clustering phenomena within Transformer blocks. The TokenMerging approach, as described by its authors, identifies clusters by merging the r most similar tokens between arbitrary source and destination sets—specifically within the Q, K, V matrices. This method highlights the potential for clustering within the internal representations of the attention mechanism. Additionally, the Agarwal et al. (2024) took a different perspective, focusing on inter-head correlations by identifying clusters across attention heads, where they measured the degree of correlation between the Q, K matrices for different attention heads. Finally, Geshkovski et al. (2024a) analyzed clustering at the token level, observing how tokens themselves can form clusters based on the dynamics of the attention mechanism.

For our experiments, we propose a comprehensive, multifaceted analysis of clustering behavior in Transformers optimized for machine translation. Specifically, we will examine clustering at three levels: **inter-head** relationships across Q, K, V matrices; **intra-head** structures within each attention head; and clusters among the tokens themselves. This approach allows us to capture clustering phenomena at different levels of granularity, offering a more complete picture of how clusters manifest within the Transformer architecture.

In terms of methodology, we will draw on techniques used to measure Neural Collapse and rank collapse, providing insight into both global and intra-class variance over the course of training. In addition, we will employ traditional clustering methods, such as K-Means, to detect explicit cluster formations in the Q, K, V spaces. By combining both theoretical and empirical analyses, we aim to shed light on how clustering dynamics evolve as training progresses and how they contribute to model performance.

9 Contribution Table

All members contributed equally to this submission.

Week	Objective	Expected Outcomes
6	Download dataset and set up environment on PACE	Have an environment set up that allows for efficient training of a large dataset.
7	Implement Translation Transformer	Establish a transformer baseline for our text translation task.
8	Implement LSTM on Translation Task	Establish LSTM baseline for translation task. Progress Report 1 Completed.
9	Replace nn.Transformer with from-scratch implementation	Set up code to allow for changes to the transformer module for our experiments.
10	Experiment 1: Token Merging	Implement clustering based on token merging and analyze performance.
11	Experiment 2: Clustering Self-Attention Heads and theory analysis complete.	Implement clustering of self-attention heads. Have a detailed plan for our own experiments based on theory research and analysis. Progress Report 2 Completed.
12	Experiment 3: K-Means Attention Clustering	Implement our attention clustering experiment idea and examine results.
13	Experiment 3: K-Means Attention Clustering	Fine tune experiment by tuning hyper parameters and recording results.
14	Analysis of Performance/Results	Compare performance of baseline models, research-based clustering experiments, and our proposed K-Means clustering method.
15	Complete Final Paper	Conclusions made on our proposed method and future work is identified. Final paper completed.

Table 1: Project timeline.

References

Saurabh Agarwal, Bilge Acun, Basil Hosmer, Mostafa Elhoushi, Yejin Lee, Shivaram Venkataraman, Dimitris Papailiopoulos, and Carole-Jean Wu. 2024. Chai: Clustered head attention for efficient llm inference.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your vit but faster.

Zhang Cheng, Haocheng Wan, Xinyi Shen, and Zizhao Wu. 2022. Patchformer: An efficient point transformer with patch attention.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. 2023. Attention is not all you need: Pure attention loses rank doubly exponentially with depth.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. 2024a. The emergence of clusters in self-attention dynamics.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. 2024b. A mathematical perspective on transformers.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention.

Charles Kelly. 2020. Language translation (english-french).

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention.

Xiang Li, Zhenyan Lu, Dongqi Cai, Xiao Ma, and Mengwei Xu. 2024. Large language models on mobile devices: Measurements, analysis, and insights. In *Proceedings of the Workshop on Edge and Mobile*

Foundation Models, EdgeFM '24, page 1–6, New York, NY, USA. Association for Computing Machinery.

Vardan Papyan, X. Y. Han, and David L. Donoho. 2020. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. 2020. Fast transformers with clustered attention.

Robert Wu and Vardan Papyan. 2024. Linguistic collapse: Neural collapse in (large) language models.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. 2022. Scaling vision transformers.