Mitigation of Hallucination and Interpretations of Self Attention of Mistral 7B AI to Analyze and Visualize Context Understanding Ability of Large Language Models

by

S.M.ABRAR MUSTAKIM TAKI
20301125
SHOWMICK KAR
20301177
SOUMIK DEB NILOY
20301207
MAZHARUL ISLAM RAKIB
20101408
ABDULLAH AL NAHID BISWAS
20301024

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
January 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

S.M.ABRAR MUSTAKIM TAKI
20301125

SHOWMICK KAR
20301177

SOUMIK DEB NILOY
20301207

MAZHARUL ISLAM RAKIB
20101408

ABDULLAH AL NAHID BISWAS
20301024

Approval

The thesis/project titled "Mitigation of Hallucination and Interpretations of Self Attention of Mistral 7B AI to Analyze and Visualize Context Understanding Ability of Large Language Models" submitted by

1. S.M.ABRAR MUSTAKIM TAKI(20301125)
2. SHOWMICK KAR(20301177)
3. SOUMIK DEB NILOY(20301207)
4. MAZHARUL ISLAM RAKIB(20101408)
5. ABDULLAH AL NAHID BISWAS(20301024)

Of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 22, 2024.

Examining Committee:

Supervisor:
(Member)

Farig Yousuf Sadeque, PHD
Assistant Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement (Optional)

This is optional, if you don't have an ethics statement then omit this page

Abstract

In recent years, Large Language Models(LLM) have shown excellent performance in a variety of Natural Language Processing tasks. However, they often produce hallucinated content. Contents that are seemingly correct and make sense linguistically, but are factually incorrect. Since researchers have started working on LLM hallucinations very recently, the problem of mitigating hallucination and understanding which factors play a role in correcting hallucinated content is relatively new. In this paper, we modified a multi-step pipeline called 'Chain of Verification' that reduces hallucination in Large Language Models by itself without having to feed in external resources. This method is particularly useful for reasoning and reading comprehension types of language tasks. In addition, we extracted the decoder layers of an large language model Mistral 7B to interpret and analyze how the correction was done under the hood. A custom attention weight pruning method was used to prune the defective layers and after pruning, the LLM model passed 3/4 test cases to give proper and correct output results.

Keywords: Transformers; Interpret; Attention; Self Attention; Hallucination; BlackBoxNLP; Large Language Model; Mistral; Mistral 7B AI; LLM; Attention pruning; random weight pruning; Attention head Analysis

Dedication (Optional)

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our Advisor Dr. Farig Yousuf Sadeque Sir for his kind support and advice in our work. He helped us whenever we needed help.
Thirdly, to Dr. Sudipta Kar(Senior Applied Scientist at Amazon Alexa AI) for his kind support and advice in our work
And finally to our parents without their throughout supports it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Contents

Chapter 1

Introduction

Hallucinations in large language models have become a topic of both fascination and concern in recent years. As artificial intelligence continues to advance, these sophisticated systems, like GPT (Generative Pre-trained Transformer)[1] models, have demonstrated unparalleled capabilities in generating human-like text. However, amidst their remarkable linguistic prowess lies a shadow—instances where these models produce seemingly plausible yet entirely fictional or misleading information, termed as hallucinations. This paper delves into the nuances of hallucinations within large language models, exploring their causes, implications, and the ethical considerations that surround their existence. By examining the mechanisms behind these phenomena, this study aims to shed light on the complexities inherent in AI-generated content, thereby fostering a deeper understanding of the opportunities and challenges presented by these groundbreaking technologies.

Several studies from 2019-2023 have tried to address this matter by developing hallucination mitigation techniques in LLMs as well as analyzing the behavior of transformers and developing methods for interpreting their output. In terms of Hallucinations, some examples might be correction of training time, generation time correction, prompt engineering,self refinement as well as Chain of verification[2].

Furthermore, we used our custom Random Unstructured Pruning method to detect and eliminate defective and useless heads from the attention layers of Mistral 7B[3]. We showed that randomly pruning flawed attention heads increases accuracy in LLM generation.

In this research, we will explore the reason behind the hallucination of LLM models and why the model does not understand the context of input sentences even after verification. We will analyze each attention head of each attention layer to interpret and visualize what the model is learning and understanding.

1.1 Research Problem

Since the emergence of transformer models in 2017[4], their lack of interpretability has posed a significant challenge in the field of natural language processing (NLP). While these models like BERT[5] and GPT[1] have exhibited exceptional performance across various language tasks, comprehending the rationale behind their predictions remains a significant challenge. This lack of transparency in their decision-making processes, especially in grasping contextual understanding, poses limitations to their practical applicability, particularly in critical domains.

One significant concern associated with transformer models is the ambiguity regarding whether the attention layer truly comprehends context linguistically or merely performs mathematical computations. Despite efforts to introduce interpretability methods such as attention visualization and feature attribution, the quest to enhance the interpretability of transformer models, specifically at the self-attention head layer, necessitates further research.
Furthermore, it's imperative to acknowledge the issue of "hallucination"[6] within large language models. The phenomenon of hallucination refers to instances where these models generate outputs that are contextually or factually incorrect but confidently presented as valid. These hallucinations can undermine the trustworthiness and reliability of these models in practical deployment scenarios, emphasizing the critical need to improve interpretability and mitigate such errors.
Hence, the primary research objective is to explore novel approaches and conduct experiments aimed at augmenting the interpretability of transformer models. By rendering these models more transparent, understandable, and trustworthy, the aim is to overcome the interpretability challenges and address the concerns related to the hallucination problem, thereby enhancing the practical utility of these models across diverse domains.

1.2 Aims and Objectives

Aim:

1. To investigate and mitigate hallucination in language models, focusing on enhancing the accuracy and reliability of model responses while minimizing the generation of false or misleading information.

Objectives:

1. Reproduce and customize the Chain of Verification pipeline to self-correct Large Language Model generations without external resources.

2. Improve the generations of the Large Language model without using Retrieval Augmentation Generation or continuously providing feedback.

3. Figure out a way to evaluate and measure hallucinated AI generation.

4. Reduce human intervention when correcting hallucinated LLM outputs.

5. Interpret the reasoning behind LLM responses both before and after correction using pipeline for Improved NLP performance.

1.3 Types of Hallucination in Large Language Models

There are two types of hallucination in Large Language Models. Those are responsible for the model's false output generation. The types are stated below in the table along with their concept and example. The expected output can be defined as the original fact which is the answer of the stated question in the real world.

Types	**Concept**	**User Input**	**Expected Output**	**Model's Output**
Factual Inconsistency	Generates inaccurate information as if it is correct	Who was the first person to land on the moon?	Neil Armstrong was the first person to land on the moon	Yuri Gagarin was the first person to land on the moon
Factual Fabrication	Generates content that is nonsensical or unfaithful to the provided source content	Tell me what was seen in the sky of Dhaka	Birds were seen to be flying in the sky of Dhaka	Elephants were seen to be flying in the sky of Dhaka

Table 1.1: Types of Hallucination

Chapter 2

Related Work

2.1 Hallucinations in Large Language Models

Recent studies on LLM have become more focused on a major issue which is model hallucination. According to recent studies on LLM, different researchers have different approaches or parameters to go through this major issue which arises due to guess-based patterns from input data instead of factual accuracy.
In the context of large language models, ”hallucination” refers to the generation of plausible yet incorrect factual information by the model. This issue arises when the model produces responses that contain inaccuracies or factual hallucinations, leading to potentially misleading or incorrect information being presented as if it were true. A major drawback of LLMs is the issue of hallucination, where they generate unfaithful or inconsistent content that deviates from the input source[7]. Hallucination is a significant challenge in language model generation across various tasks, and it has not been fully resolved by simply scaling up training data or model size. Model hallucination is an obstacle to language-based AI systems which can harm people who rely on their outputs [8].
Additionally, hallucinations in machine translation are categorized into two types: hallucinations under perturbation and natural hallucinations [9].
Most importantly, a recent study shows that passage-level hallucination detection is often required instead of sentence-level hallucination detection in real-world applications which is a new benchmark for the evaluation of passage-level detection methods[6]. According to a recent survey, in LLMs, the scope of hallucination encompasses a broader concept, primarily centering on factual errors [7].
Recent research on LLMs has presented many solutions to overcome the model hallucination Among them, Chain-of-Thought prompting is a method for improving the reasoning abilities of large language models. It involves generating a series of intermediate reasoning steps, or a ”chain of thought,” to help the model arrive at a correct answer for a given task. By guiding the model through a series of intermediate reasoning steps, this method encourages the generation of more coherent and contextually supported outputs. Additionally, the interpretability provided by the chain of thought can potentially help identify and address instances of model hallucination by allowing for a better understanding of the model's decision-making process. While Chain-of-Thought prompting may help reduce model hallucination in some cases, that is not its primary purpose. Model hallucination refers to the phenomenon where a language model generates text that is not supported by the

input or context. While Chain-of-Thought prompting may help improve the accuracy of language models on reasoning tasks, it is not specifically designed to address model hallucination [10]. Additionally, Chain-of-Verification (CoVe) is an approach designed to reduce hallucinations in large language models (LLMs) by allowing the model to fact-check its responses and self-correct them. The method involves several steps, including drafting an initial response, planning verification questions to fact-check the draft, answering those questions independently to avoid bias, and generating a final verified response. CoVe has been shown to decrease hallucinations across a variety of tasks, including list-based questions from Wikidata, closed book MultiSpanQA, and long-form text generation. The method has demonstrated improvements in precision on list-based tasks, general QA problems, and long-form generation, leading to substantial gains in performance over the original language model response. The approach aims to provide substantial performance gains over the original language model response just by asking the same model to deliberate on (verify) its answer. CoVe has the potential to improve precision on list-based answer tasks, closed book QA, and long-form generation, making it a promising method for enhancing the accuracy and reliability of large language models [2].
Furthermore, The gap that Knowledge Injection (KI) aims to address is the risk of false information, or "hallucination," being included in the output of Large Language Models (LLMs). LLMs, while powerful, are susceptible to generating incorrect or misleading information, especially in enterprise use cases where reliable, fact-based text generation at scale is crucial. KI bridges this gap by leveraging contextual data from a knowledge graph to guide the content generation process, thereby reducing the likelihood of false assertions and improving the overall quality and reliability of the generated text. By filling this gap, KI enables businesses to generate more reliable, fact-based, and higher-quality text from LLMs, addressing a critical need for accurate and trustworthy content generation in enterprise settings [11].
Last but not least, the latest update (December 31, 2023) in the field of LLM is to alleviate hallucinations using Retrieval-Augmented Generation where the creation of a benchmark dataset is important to measure the extent of hallucination [12].

2.2 Interpretability and Analysis of Transformers

Recent research has focused a lot of emphasis on the interpretation and analysis of the attention processes in transformer-based models. This topic has been covered in depth in a large number of studies, giving insight into the behavior of attention heads and how they affect the performance of these models as a whole. We have read approximately 30 research papers on related topics from top-tier conferences and journals to explore the area. We will examine the methods researchers have used to interpret and analyze transformers in this literature review.
The literature review starts with a paper by Elena Voita et al in 2019. Before this paper, there was no specific research on the interpretation and analysis of each one of the encoder attention heads in multi-head attention transformers. The author of the paper Elena Voita has analyzed every attention head using layer-wise relevance propagation(LRP)[13]. After the analysis, they found the important and confident heads. The author developed novel pruning systems that keep the important and confident heads and prune other encoder heads with affecting performance. They applied this method on the WMT dataset(ENGLISH to RUSSIAN Language)[14],

pruning 38 encoder heads out of 48 resulting in a drop of only 0.15 BLEU[15].
Another notable contribution to the understanding of attention behavior in Transformer language models is Jesse Vig's paper which seeks to address the question in terms of targeting different parts of speech, alignment with dependency relations, and capturing distant relationships.
The authors provide three levels of detail they used to visualize attention in Transformer-based language models: Attention-head Level, Model Level, and Neuron Level. They analyze the model's behavior over a large text corpus to answer the following questions: "Does attention exhibit any alignment with syntactic dependency?" "Which parts of speech are notably addressed or attended by specific attention heads?" and "In which process does attention capture long-distance dependencies compared to short-distance ones?" This analysis is applied to a small pre-trained model for GPT-2.
The primary findings from the analysis are as follows: The model's middle layers exhibit the highest emphasis on dependency relations when it comes to attention. Depending on the depth of the layer, specific parts of speech are targeted by attention heads. Finally, the deepest layers of the model display the longest attention spans, although the extent of attention varies considerably across different heads [16].
In the same year, Jesse Vig et al released an informative research article that unveiled several significant discoveries: Firstly, the Transformer model's attention showcases a hierarchical arrangement, engaging distinct linguistic components at varying layer depths. Secondly, the alignment between attention and dependency relations manifests most prominently within the intermediate layers of the model, while the deepest layers capture the furthest connections. Lastly, the paper presents illustrative sentences that exemplify specific patterns identified by individual attention heads[17].
The authors Clark et al. (2019) provide a detailed analysis and examination of the attention mechanism employed in the BERT language model. The authors investigate how BERT processes input text by analyzing various linguistic phenomena and propose a visualization method for understanding its attention patterns.
Initially, the authors introduce the concept of attention and elucidate its functioning within transformer-based models like BERT. They go into detail about the BERT's self-attention mechanism, which enables the model to selectively attend to various input sequence segments during processing. Subsequently, the authors put forth a methodology for scrutinizing BERT's attention patterns. They employ a set of linguistic phenomena, including long-distance dependencies, negation, and tense, to assess how BERT attends to these aspects. For each phenomenon, the authors construct test sentences incorporating the particular linguistic feature and examine BERT's attention patterns in response.
To visualize BERT's attention patterns, the authors propose the generation of heatmaps that highlight the most influential input tokens for a given prediction. These heatmaps serve to identify the significant sections of the input sequence relevant to specific predictions, offering insights into the decision-making process of the model.
The authors analyze BERT's attention patterns across a range of linguistic phenomena. The findings reveal that BERT largely focuses on content words and demonstrates a preference for processing syntactic dependencies over long distances. Additionally, BERT displays attention towards negation and tense, albeit to a lesser extent compared to other linguistic phenomena.
Overall, the paper presents a methodological framework for analyzing the attention

patterns in BERT and contributes insights into the model's text processing. The authors propose that further exploration of attention patterns in BERT and similar transformer-based models can enhance our understanding of these models and their behaviors [18].

The paper by Wiegreffe and Pinter et al(2019) is a critique of the use of attention mechanisms in deep learning models as a means of providing explanations for their behavior. The authors argue that attention is not sufficient to provide a complete explanation of a model's behavior and propose an alternative approach based on counterfactual explanations.

The authors first define the concept of explanation in the context of machine learning models and argue that a complete explanation should include both causal and counterfactual information. They argue that attention mechanisms, which provide a weighted representation of input features, do not provide sufficient causal information to fully explain a model's behavior. To illustrate their point, the authors present a case study of a sentiment analysis task, in which they train a model on a dataset of movie reviews. They show that the attention weights of the model do not always correspond to the features that are most important for predicting sentiment and that the attention mechanism can be misled by superficial features of the input text.

The authors then propose an alternative approach based on counterfactual explanations, which involve changing the input features of a model to see how its output changes. They argue that counterfactual explanations provide a more complete picture of a model's behavior by allowing researchers to explore the causal relationships between input features and output predictions. The authors conclude that attention mechanisms are not sufficient to provide complete explanations of a model's behavior and that researchers should instead focus on developing more comprehensive explanations based on causal and counterfactual information. They suggest that their proposed approach based on counterfactual explanations can provide a more robust and interpretable understanding of how deep learning models work[19].

Around the same time, a counter paper by Sarthak Jain and Byron C. Wallace et al (2019) came out that examines the question of whether attention mechanisms in neural models serve as reliable explanations for model predictions. The main findings of the study suggest that attention weights alone cannot be considered reliable explanations, as they often exhibit inconsistency and are influenced by irrelevant factors. The authors emphasize the need for caution when interpreting attention mechanisms as explanations in neural models[20].

Bastings and Filippova introduced a different approach called input saliency methods for explaining NLP models. Unlike attention, which doesn't always align with important features, saliency methods consider the actual relevance of each word in making predictions. This allows for a more accurate understanding of why certain decisions are made. While attention can be improved, it can also lead to easier changes in decisions. On the other hand, saliency methods provide direct access to the inner workings of the model, enabling a comprehensive evaluation of each word's importance. However, it's important to note that some saliency methods have limitations and may produce unexpected outcomes. Ultimately, the goal is to determine the most relevant inputs for accurate predictions[21].

In 2020, Multimodal Routing was introduced by Tsai and Salakhutdinov et al. This approach involves adjusting weights between input modalities and output represen-

tations for each sample, enabling the identification of the significance of individual modalities and cross-modality features. This allows for global and local interpretability. The process involves encoding raw inputs to unimodal, bimodal, and trimodal features, followed by routing that updates hidden representations and adjusts weights based on similarity. Compared to other methods, Multimodal Routing shows improved performance and robust predictions by dynamically associating features with concepts. It outperforms EF-LSTM, LF-LSTM, and RAVEN models and competes well with MulT. In summary, Multimodal Routing offers dynamic interpretability while maintaining strong performance[22].
The Author Yaru et al proposed a self-attention attribution method named ATTATTR in 2020. The purpose of ATTATTR is to enhance the interpretability of self-attention by analyzing information interactions within transformers. The authors used BERT for extensive studies, where they applied ATTATTR to detect the most important attention heads and pruned non-important heads, extracting the most saliency dependencies to derive attribution trees to visualize information flow. The authors demonstrated that ATTATTR can generate adversarial triggers, which can be utilized for carrying out non-targeted attacks. The results show that ATT A TTR is effective in interpreting information interactions and can lead to better performance in Transformer-based models. There was also a paper in 2019 by Voita[15] published in ACL which also detected important heads and pruned non-important heads, but the difference was they did the work on NMT models and they used LRP[13] whether in this case, the author used their noble proposed architecture[23].
Following the successful research on interpretations of the NMT(Neural Machine Translation based on Transformers) model in 2019[15], Voita proposed another contribution to the Interpretability NLP field in 2021. This time, the authors used LRP[13] to analyze the contribution of each word in the input sentence to the final translation. The authors found that when the model relies too much on the target sentence, which they need to translate, they can make mistakes. But when the model gets more data, they rely more on the input sentence and they make fewer mistakes. So, if we give more examples of sentences in both languages, it will be better for translations. This research provides a better understanding of how the NMT model works, how they make predictions, and how they could be improved which could lead to a better machine translation system in the future[24].
The paper of Weicheng Ma studied the contribution of attention heads in Transformer based models for multi-language and cross-lingual tasks. The authors conduct experiments on several datasets including the Universal Dependencies (UD) treebanks for 50 languages, the CoNLL 2002 & 2003 datasets for named entity recognition (NER), and the XNLI dataset to evaluate the performance of models(The author used two-pre-trained models, one is bert- base -multilingual -cased and xlm-roberta-base) with a different number of attention heads. The author also proposes a gradient-based method to identify the heads to prune to remove unimportant attention heads and compare the performance of pruned models with full models, and show that pruned models can achieve comparable or better performance than full models in cross-lingual and multi-lingual sequence labeling tasks. The finding of this paper can help researchers understand the interpretability of Transformer based models in cross-lingual and multi-lingual tasks, and improve the performance of these models. This paper may sound like the 2019 Voita paper but the Voita paper was applied to NMT models where this paper is applied on the cross and

multi-lingual tasks[15], also Voita paper applied LRP[13] whether this paper applied a gradient-based method. So two papers are different in their approaches but the same in their targets[25].
In Clara Meister et al.'s paper, the researchers investigate the interpretability and understanding of transformer models by sparsifying attention. They find that attention weights are not directly linked to model inputs but heavily influenced by internal representations. This raises the question of deriving meaningful insights from "interpretable attention weights" when the internal representations are not interpretable. The authors explore generative tasks and post-hoc interpretability, utilizing feature importance metrics to assess constraints. The experiments reveal that inputs and intermediate representations are not interchangeable, sparse attention does not equate to sparse input feature importance, and there is only a weak correlation between attention and feature importance. The paper emphasizes the limitations of using attention as an indication of input influence on the model's decisions[26].
The research paper of Yuchen focused on investigating attention redundancy in multi-layer multi-head attention mechanisms, with a specific emphasis on the BERT-base model. The main research question was to explore the patterns and characteristics of attention redundancy and its implications for model interpretation and compression. The authors utilized the GLUE datasets and analyzed the redundancy using various token-based and sentence-based distance functions. The findings revealed clear redundancy patterns among attention heads, which persisted throughout both the pre-training and fine-tuning phases. Surprisingly, these patterns were found to be task-agnostic, as they also occurred in redundancy matrices generated from randomly generated token sequences. The paper introduced a simple zero-shot pruning method based on these patterns, demonstrating its effectiveness in significantly reducing the number of attention heads while maintaining comparable fine-tuning performance. Overall, this comprehensive analysis improved our understanding of neural language models and provided valuable insights for model interpretation and optimization[27].
The article author by Mattia and Christoph et al proposes a method that enhances the interpretability of attention mechanisms in deep neural networks. Despite the success of attention in improving model performance, its interpretability has been a subject of debate. The authors suggest expanding attention from low-level input features to high-level concepts to make them more interpretable. They introduce the Concept Transformer, which is a deep learning module that explains model outputs using attention over user-defined high-level concepts. These explanations are feasible and faithful, aligning with human reasoning and accurately reflecting the model's decision-making process. To ensure feasibility, attention heads are trained to conform to known relations between inputs, concepts, and outputs. Additionally, faithfulness is ensured by enforcing a linear relationship between concept representations and their contribution to classification probabilities. The paper discusses existing methods and their limitations, highlighting the need for high-level, human-understandable concepts. The Concept Transformer is presented as a solution that generalizes attention to such concepts, providing interpretability that aligns with human reasoning. The module achieves state-of-the-art performance on image datasets while being versatile, less complex, and easier to train than other techniques.[28].
The usefulness of self-attention for direct speech translation is covered in Belen

Alastruey's paper. First, the layer-wise token contributions to the encoder's self-attention are examined. To demonstrate that some attention weights are avoidable, the local self-attention is proposed to be replaced with an efficient one. In this work, layer-wise contributions are proposed to analyze the patterns of self-attention. Due to the high cost of computational self-attention, the new architecture is made to increase model efficiency while reducing information loss. It is accomplished by replacing regular self-attention in those layers with local attention. For the verification, finally, the trained model substitutes the regular self-attention with the local. By analyzing contribution matrics it is seen that relevant scores are placed in a diagonal pattern. Most importantly, regarding interpretability, it is found that the transformers establish a connection between words in speech sequence[29].
The authors Carolin and Simon et al investigate differences and clusters in large pre-trained language models of the BERT family by utilizing a new methodology that predicts coarse features of BERT's representations. The paper discusses related research on self-attention mechanisms, individual neurons, layer activations, and the geometrical space of layer activations while emphasizing the unique approach of examining contextual representations from BERT's perspective. The findings reveal that the space of contextual information exhibits clusters, with syntax and positional information playing a significant role. The performance of Part-of-Speech models aligns with these patterns, highlighting the correlation between simple positional information and representations. Since different information types are correlated, the study warns against misinterpreting learned patterns and overestimating model sophistication. They stress the importance of understanding language model limitations to drive future improvements in training objectives and processes. The paper presents a new perspective on contextual representations in BERT models, uncovering the influence of syntax and positional information on their high-dimensional spaces. It also acknowledges the models' limitations and provides insights into potential enhancements[30].
Dairui and Derek et al propose a two-tier attention architecture for explainable deep learning-based NLP tasks. Their paper introduces the "Bi-level Attention Based Topical Modes" (BATM) model, which uses two attention layers to capture topic-word and document-topic distributions. They evaluate the model on large-scale datasets for news article classification, demonstrating competitive performance and improved explainability. The findings reveal that specific attention heads focus on words related to specific topics. Their approach aims to enhance model interpretability while achieving state-of-the-art results[31].
The authors, Fan and Zhouxing et al have developed new methods for post hoc interpretations of machine learning models. These methods aim to measure the faithfulness of interpretations using two criteria: sensitivity and stability. The study shows that these methods are effective in overcoming the limitations of previous gradient-based and removal-based approaches. However, there is still no clear consensus on how to define and evaluate interpretation techniques. The sensitivity criterion helps to add small perturbations in a local region of the token embedding, while the stability criterion ensures that similar inputs produce consistent explanations. The study proposes new criteria for evaluating interpretation methods and introduces new robustness-based interpretations inspired by the sensitivity metric. The results show that existing interpretations have inconsistent performance regarding different criteria, but the proposed robustness-based interpretations achieve the best

performance under sensitivity and stability.[32].
Mor et al concentrated on revealing the core prediction construction mechanism by reverse engineering the Feed Forward Network layers. Furthermore, the internal use of the FFN layer outputs to create predictions is being examined. FFN layers, according to their technique, calculate updates that can be interpreted in terms of the output vocabulary. Tokens are represented as an evolving distribution over the vocabulary in this case, and the FFN output is decomposed as a collection of updates to the output distribution. By comparing ideas in the top 10 random vectors from a normal distribution with the top tokens of value vectors, the method is proven to work. It was easy to use and quick to find the non-toxic updates manually. In addition, value vectors are frequently displayed as encoded human interpretable notions. Finally, FFN output is studied as a linear combination of output vectors, denoted values.[33].
GlobEnc is a novel method developed by Ali Modarressi and Mohsen Fayyaz et al for studying token attribution in a single-layer model. This approach aggregates the analysis across layers and takes into account the full encoder layer, including the attention block and output layer normalization. The authors used two types of explainability techniques: norm- and weight-based methods. In their tests, weight-based approaches averaged raw attention maps across all heads, whereas Norm-based methods determined the relative relevance of each token by layer-wise normalization. The authors showed that their approach beats previous techniques in quantifying global token attributions through thorough quantitative and qualitative examination. Additionally, they demonstrated that including nearly all of the encoder's components improves the accuracy of analyses in both local (single layer) and global (whole model) contexts. The authors intend to use their global analysis method to analyze diverse datasets and models in the future to get important insights about model choices and interpretability[34].
The paper of Hassan Sajjad et al introduces a new framework called ConceptX, which analyzes how latent concepts are encoded in pre-trained language models. It uses clustering techniques to discover these latent concepts and provides explanations by aligning them with human-defined concepts. Lower layers capture lexical concepts, while middle and higher layers represent core-linguistic concepts like morphology and syntax. The paper discusses previous research on interpretability in deep NLP, focusing on decomposability, representation analysis, attention heads, and language compositionality. It aligns with previous studies but distinguishes itself by conducting an unsupervised analysis and using human-defined concepts to generate explanations. The findings highlight the distribution of linguistic concepts across model layers and the varying alignment between latent and human-defined concepts. The paper suggests that compositionality can lead to novel explanations and improved concept coverage[35].
Following the paper from Voita[24] in 2021, there were many papers on the Interpretability of NMT models that were solely focused on source sentence token attributions. Javier and Gerard found a lack of understanding of every input token in the NMT model predictions. From this, the author targeted to propose an Interpretability method called ALTI+ that measures each input token contributions for both contexts to the encoder-decoder transformer predictions allowing for a better understanding of the inner workings of current NMT models and providing insights into how contextual information in mixed across the encoder. The authors

concluded that their research provides a better understanding of how NMT models work and can be extended to any encoder-decoder Transformer-based model for better interpretability[36].
The authors Damai Dai, and Li Dong researched pre-trained transformers to explore how factual knowledge is stored. They proposed a method for identifying knowledge neurons that contain factual information in a fill-in-the-blank task. The authors also performed both qualitative and quantitative analyses to demonstrate that these knowledge neurons are positively correlated with knowledge expression. Additionally, they experimented with suppressing or amplifying the knowledge neurons to modify factual information without the need for fine-tuning. The authors conducted the experiment on Bert Base Case pre-trained model and the used PARAREL Datasets and their self-made BINGREL Datasets. The author proved their points of view, but there were such limitations in the paper they only extracted factual knowledge where they can extract more types of information and they did this only for fill-in-blank cloze tasks which it can be done for many other tasks[37].
Kayo Yin and Graham Neubig conducted a study on language models (LMs) using contrastive explanations. Their goal was to explain why LMs make certain decisions and identify the factors that influence those decisions. They used three methods, including saliency scores, to compute contrastive explanations. These explanations help us understand why certain words are more important than others. Users found contrastive explanations more useful, especially when trying to erase specific words. They also discovered clusters called "foil clusters" that relate to different aspects of language. Overall, contrastive explanations improved our understanding and simulation of language models[38].
The paper of Raymon and Wen proposes a human-in-the-loop pipeline that combines understanding the multi-head self-attention mechanism in transformers and developing attention augmentation methods for improved accuracy, efficiency, and interpretability of transformer models. The pipeline involves analyzing task-specific attention patterns, injecting them into smaller and original models, and evaluating their benefits. Case studies on extractive summarization and topic segmentation demonstrate significant improvements when injecting the discovered patterns into attention heads. The related work section discusses attention analysis in transformers, investigating attention head matrices and their functions in various tasks. Attention augmentation techniques, such as replacing attention weights with fixed matrices or applying masks, are also explored. The paper briefly covers model interpretability, highlighting the goal of making interpretability inherent in neural models through the discovery of human-distinguishable patterns. The paper presents two case studies on extractive summarization and topic segmentation using BERT-based models. Results show performance improvements and insights gained from the injected attention patterns. Future work includes applying the pipeline to other NLP tasks, exploring pattern transferability and different model variants, and automating the pipeline. User studies will be conducted to assess the trade-off between human cost and the coverage of discovered patterns. The paper acknowledges limitations regarding the focus on English datasets with long documents and the use of a specific visual interface for attention analysis[39].
Patrick Huber and Giuseppe Carenini introduce a new method for understanding the structure of long documents. They apply the sliding window Approach to break down lengthy texts into smaller sub-word sequences, which are then grouped into

segments of the maximum input length allowed by the PLM. To test their approach, they use BERT-base and BART-large models, which offer diverse objectives. Additionally, they explore fine-tuning the BERT model and investigate abstractive summarization and question-answering tasks as part of the analysis. The authors evaluate the performance of various models on two classification tasks. They are sentiment analysis and natural language inference (NLI). They utilize seven datasets for fine-tuning and find that sentiment analysis models consistently outperform NLI models in capturing discourse structures, which is not surprising since sentiments are more aligned with such structures. The randomly initialized transformer model delivers the poorest results, while fine-tuned models perform either equal to or worse than the standard PLM. However, the drop in performance for fine-tuned models is minimal. The authors observe that discourse information is consistent both locally and generally, and similar information is captured in the same heads across the experiment[40].

Xiaofeng and Yikang examined the Mixture of Attention Heads (MoA), an attention mechanism that selects different heads for different inputs. Their experiments in machine translation and language modeling revealed that MoA outperformed the original Transformer architecture. Surprisingly, MoA achieved comparable results to larger models while using only half the computational resources. They employed a shared dictionary for word embeddings and adopted RoBERTa's masked language modeling experiment to replicate the training of Pre-trained Language Models (PLMs). Notably, even a base-scale MoA model performed as well as or better than larger Transformer models. By adding more attention experts, MoA improved performance without significantly increasing computational complexity. Importantly, the analysis showed that different attention experts specialized in processing specific input tokens[41].

Kiron and Xuan discuss hyperparameter tuning, which is crucial for achieving high accuracy in models. Hyperparameters like the number of layers and learning rates can significantly impact model performance. The authors suggest using a post-hoc interpretation for hyperparameters with EBMs, which involves training different neural network models with various hyperparameters and recording their resulting accuracy metrics. There are two main ways to explore the effect of hyperparameters: the prescriptive approach, which offers advice based on large-scale experimental runs, and the development of tools to improve understanding. The authors use machine translation datasets to demonstrate how EBM can be used to interpret Transformer hyperparameters and report three types of findings. They find that hyperparameters may not be uniformly sampled from the space, and a large number of samples for EBM fitting may not be necessary. To gain a better understanding of EBM's ability, several experiments are conducted. It is also important to note that not all hyperparameters are equally important, and some exhibit a non-monotonic correlation with BLEU scores. Finally, the authors found that although EBM fits well under limited data, it struggles with transfer across different MT datasets[42].

Pre-trained language models based on Transformers have proven highly effective in various language processing tasks. Researchers Xiaozhi and Kaiyue made a noteworthy discovery about "skill neurons" within Transformers. These neurons play a vital role in handling specific tasks by encoding task-specific abilities. Importantly, skill neurons are largely formed during pre-training rather than fine-tuning. The study's contributions include identifying skill neurons, confirming their significance in encoding task skills, and exploring potential applications. When these neurons are randomly disrupted, model performance declines significantly, underscoring their critical role. These findings greatly advance our understanding of pre-trained language models[43].

Michael, Hao, and Daniel proposed PAPA as an alternative to attention processes. This method replaces input-dependent attention matrices with constant ones by averaging attention weights across many inputs. The authors evaluated six tasks using various pre-trained Transformers and observed that all models performed well even without input-dependent attention. The average relative drop from the probing baseline was only 8%. They also noted that performance remained unaffected even after swapping half of the input-dependent attention matrices for constant ones. The study suggests exploring this process further within the Transformer architecture and highlights the possibility of simpler substitutes for input-dependent attention[44].

Chapter 3

Dataset Collection

In this paper, we tried quite a few datasets with varying difficulties in LLM tasks to test out our modified pipeline. At the beginning, we wanted the recreate this pipeline[45] . However the final dataset versions for all three of the tasks described in the paper was private. We emailed the paper authors if they would give us the final dataset version as well the results of the intermediate steps to help us reduce computation time. They said, they would require internal approval to release data. Hence we tried to take some different approaches.
Firstly, following the original pipeline[45], we generated 100 list-like questions using ChatGPT. For example, 'List some prominent poets of the 19th century'. There were a few problems following that approach. Firstly, since, there was no true labels in the generated texts, we could not test out our pipeline against those questions and get a tangible accuracy measure. Since Mistral is a generative model, unlike classification tasks, an LLM response can not be directly classified as correct or the wrong answer. Thus we tried to work on some already labeled publicly available datasets.
Because we were working on a pipeline, where the LLM would correct its response without any external interventions or resources, we then looked for a dataset where the answer format was easier to evaluate. Hence we chose the RACE(a large dataset of Reading Comprehension Questions). The dataset looks like the bellow figure. In short, given a medium length article, a questions and four corresponding answer choices, the LLM has generate a correct response to that question.

Chapter 4

Model Selection

For our research, We used the Large Language Model called Mistral 7B[3]. Which was a nice middle ground for us both in terms of performance and computational cost. The models with lower parameters than Mistral 7B were way too poor in terms of performance such as FLAN T5 Small[46]. On the other hand, Models that were bigger in terms of parameters such as GPT-4[1] or LLAMA[47] Large were way too large to load on our computer. In addition to that, The APIs for most of them were paid based on daily data usage. Mistral 7B is superior in terms of Common Sense Reasoning and Reading Comprehension to other models of a similar number of parameters.

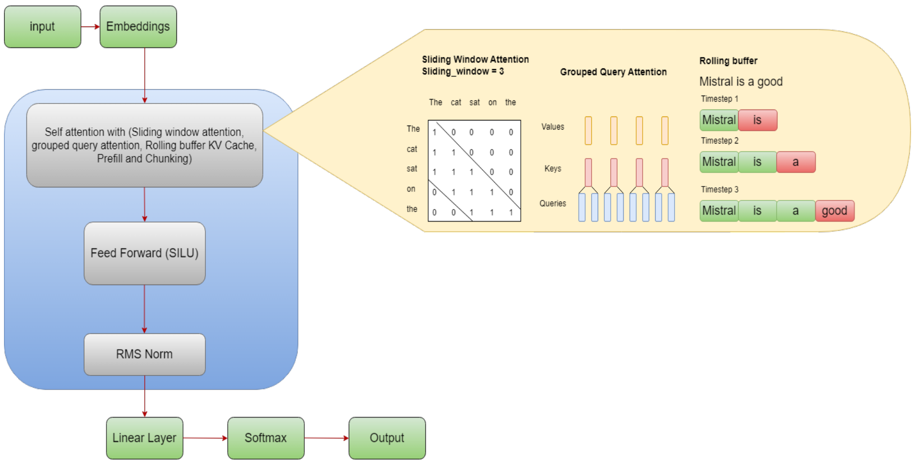

Figure 4.1: Mistral 7B architecture

Mistral 7B[3] follows three main components such as self-attention[4], Feed forwarded layer (Sigmoid Linear Unit)[48], RMS norm[49], etc.
Initially, input text is embedded into high-dimensional vectors
Then the most important thing in this architecture, self-attention is implemented through sliding window attention[50], grouped query attention[51], and rolling buffer KV cache[52] where sliding window along with rolling buffer KV cache is what makes mistral fast, fewer parameters and ability to handle a large number of tokens. The grouped query attention allows faster inference and lower cache size.
After that, the activation function, the Sigmoid Linear Unit helps for improved accuracy and high efficiency.
Most importantly, Root Mean Square normalization[49] makes this architecture computationally simple.

Chapter 5

Methodology

In order to unravel the intricacies of Large Language Models (LLMs) and delve into their functioning, a methodical approach is imperative. This section delves into the methodology employed to interrogate and analyze LLMs. Beginning with the meticulous preparation of prompt templates (Section 5.1) and the generation of baseline responses (Section 5.2), this process forms the groundwork for subsequent stages. Moving forward, Sections 5.3 and 5.4 detail the generation of a series of verification questions and their respective answers, an essential phase in evaluating the accuracy and coherence of LLM-generated content. Finally, Section 5.5 encapsulates the generation of revised responses, reflecting the iterative nature of assessing and refining the outputs obtained from these models. Each step in this methodology contributes significantly to a comprehensive understanding of LLM capabilities, limitations, and the reliability of their outputs.

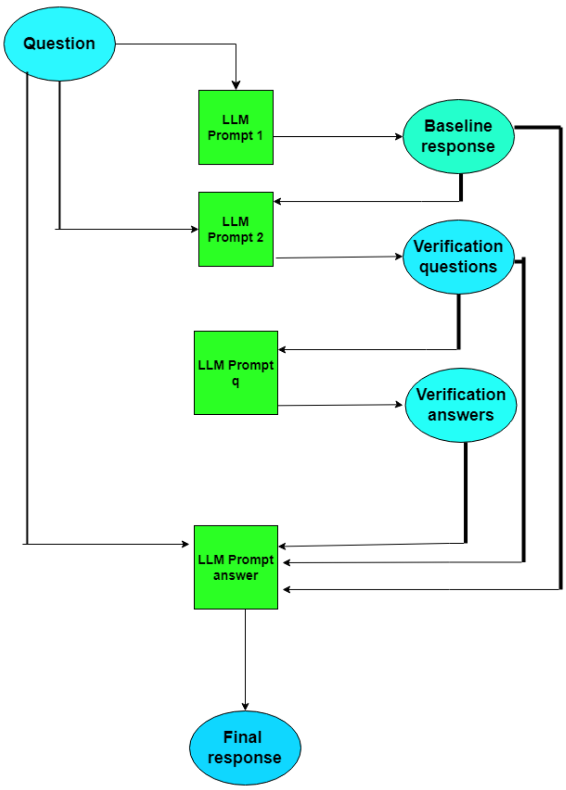

Figure 5.1: Workflow of Pipeline

5.1 Preparation of LLM prompt templates

First we prepared some prompt templates within which we kept placeholders for feeding the LLM the passage, question and the corresponding answer choices. In total we prepared four separate prompt templates which was used in different stages of our pipeline. I will explain them all one by one when describing the respective step of our pipeline. Basically, when running each stages of the pipeline dynamically changed the prompt templates according to which data row it corresponds to.

5.2 Generating a Baseline Response

In this stage we utilized a prompt template called *BASELINE PROMPT MCQ QUESTION*. In this template we carefully curated some instructions that the LLM has to follow. Within the LLM prompt template, we incorporated the passage, the question and the answer choices that the LLM had to choose from. We place instructions for the LLM to just select and generate the answer choice that it thinks is appropriate. We instructed it to avoid generating additional words and to not answer in full sentences. For example, if the question had these four answer choices - 'England', 'The United States', 'Germany', 'Brazil'; our prompt instruction was to generate only 'The United States' if it was correct and nothing else. That way, testing it against the true labels would be simpler.

```
BASELINE_PROMPT_MCQ_QUESTION = """Given an article, a question, and 4 choices,
choose the best choice among the four. When choosing make sure your choice is
based on critically analyzing the article, question and the answer choices.

Answer format: Just output the best choice. Don't write any explanation. Don't write
anything extra. Just the answer choice.

Article: {article}

Question: {question}

Options: {options}
"""
```

Figure 5.2: Baseline Prompt Template

5.3 Generating a Series of Verification Questions

In this step, we utilized another prompt template that we crafted called *VERIFICATION QUESTION PROMPT MCQ QUESTION*. In this prompt template, we again included the passage, question and the options. In additional to that, we incorporated the baseline response that LLM provided earlier in this LLM prompt. We placed instructions to read the question and the baseline response. Then based on the baseline response, generate a series of verification questions from the contents of the article. The goal was to prepare some questions that would help the LLM determine and mitigate some errors that the LLM made in the first response. After fetching the LLM response, we performed some text processing to filter out the questions and placed it in a separate list. Since the good generative models were way to computationally expensive for the computers in out lab, the LLM often failed to generate a response that obeyed the instructions on the answer format placed in the prompt. Hence we had to take an extra step to clean up the generated questions before it was ready to be sent to the next stage of our pipeline.

```
VERIFICATION_QUESTION_PROMPT_MCQ_QUESTION = """ Your task is to ask 2-3
verification questions(answerable in 3-4 words) based on an article, the original
question with 4 choices and a baseline response. The output format should be like this.
Each line of the output should have only one question. So new questions appear in a
new line.

Article: {article}

Actual Question: {question}

Options: {options}

Baseline Response: {baseline_response}
"""

EXECUTE_PLAN_PROMPT_SELF_LLM = """Given an article, Answer the following
question correctly in one sentence.
```

Figure 5.3: Verification Question Prompt Template

5.4 Generating Answers to the Verification Questions

In this stage, answering those verification questions could be done in several ways. Either we could ask the LLM to generate some verification questions in the same prompt that we asked it to generate the baseline response. This approach has one advantage and several disadvantages. The advantage is, it would require us to run less steps in out pipeline this reducing computational cost. However, the problem with a single prompt is that verification questions and the answers would then be conditioned upon the baseline response which would lead to bias in answer generation the mistakes would propagate in future steps. Hence a better approach was to use a separate LLM prompt to both generate and answer the verification questions. That way there is less risk of mistake propagation. However, the third approach that we followed in our research was to put the verification question and answer generation in separate LLM prompts. We generated answers to each individual verification questions in separate prompts. That way, the answers to the verification questions won't see the baseline response and would only answer using the article as reference. This approach, although much more computationally expensive helps in two ways. Firstly since, we are using a relatively small model, it helps keep the individual prompts small. Secondly, answers to individual verification questions are not conditioned on the baseline response, thus avoiding bias.

```
EXECUTE_PLAN_PROMPT_SELF_LLM = """Given an article, Answer the following
question correctly in one sentence.


Article: {article}

Question: {verification_question}
Answer:"""
```

Figure 5.4: Verification Execution Prompt Template

5.5 Generate a Revised Response

In the Final step of our pipeline, we used another prompt template - *FINAL REFINED PROMPT MCQ QUESTION*. In this stage, we concatenated our article, question, answer choices, the baseline, response, verification question answer pairs all together, fed it to LLM and instructed it to generate a revised response based on that. We then compared how accurate the revised responses were compared to the baseline responses.

```
FINAL_REFINED_PROMPT_MCQ_QUESTION = """Given the below `Article`, `Original
question` and `Baseline Answer`, output the answer to the original question. If the
baseline response seems correct, don't change it. Otherwise change it to another
choice.

Answer format: Just output the best choice. Don't write any explanation. Don't write
anything extra. Just the answer choice.

Article: {article}

Original Question: {question}

Options: {options}

Baseline Answer: {baseline_response}

Verification Questions & Answer Pairs:

{verification_answers}
```

Figure 5.5: Final Response Prompt Template

5.6 Attention Weight Pruning Method

We have used our custom random unstructured attention weight pruning method. In this method, we input the layers list we want to prune weights and percentages. It chooses a random index of weights for each query, key, and value vector from those layer inputs according to the measurement of the percentage, then traverses through them and prunes the weights

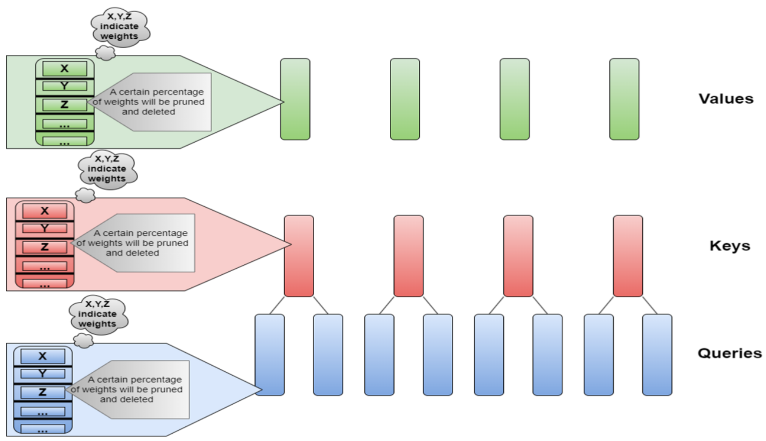

Figure 5.6: Simple Diagram for representing the Attention Weight Pruning Method

Chapter 6

Interpretations and Analysis of Self-Attention

We have interpreted and visualized the self-attention layer of Mistral to decode the understanding ability. The self-attention layer is a nonlinear layer of a transformer-based Large Language Models refers to each word giving other words attention to understand their importance. The self-attention layer tries to understand and extract the context from the input sentence and based on the attention output maps, the model performs some other computations and predicts the next words.
Mistral 7B Ai has 31 layers and each layer has 31 heads. After getting the attention layer output, we traversed through each layer and each head to analyze if it was a good attention head or not. The version of the model used decoder self-attention. In decoder attention, each word can only concentrate on itself and before its sequences, but not the next word tokens. We have visualized each attention head of each layer and also the whole layer's attention heads.
After analyzing each head on each layer, we have selected some layers that have many bad attention heads. Here, bad heads refer to the heads that do not give good enough attention to other words. We found many attention heads that were almost dead. Below is a figure of a dead attention head

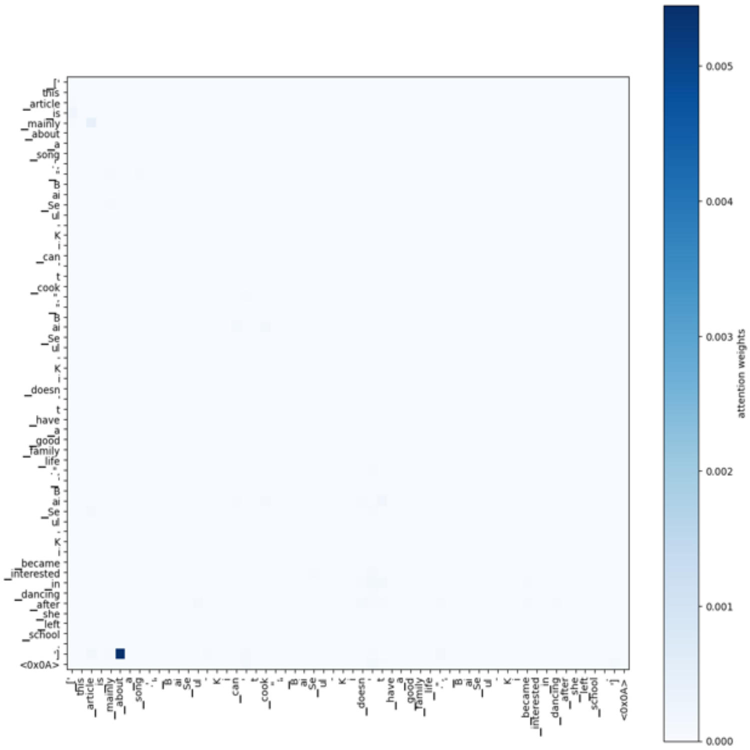

Figure 6.1: Example of Bad Attention heads

We found so many good attention heads that gave proper attention to their respective word tokens. Here is an image of a good attention head

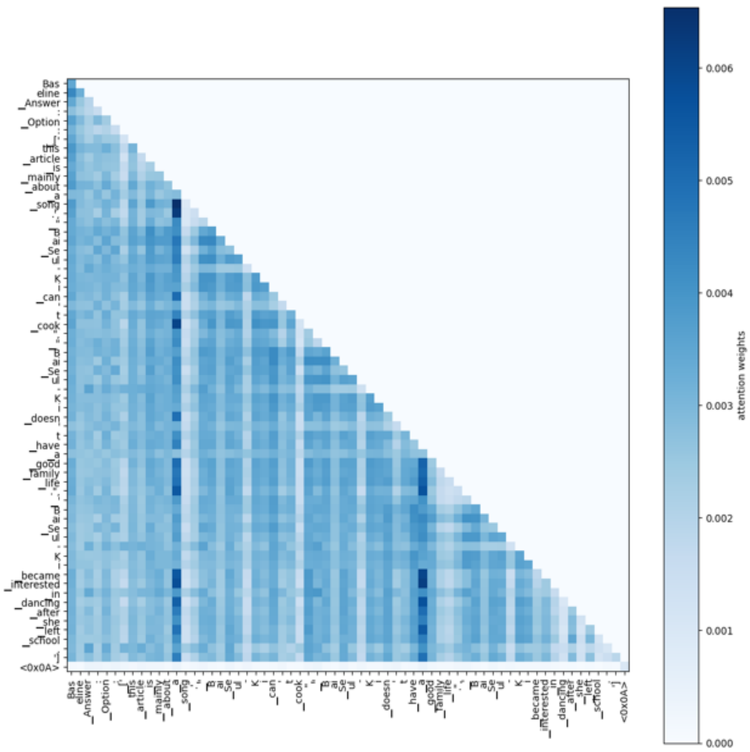

Figure 6.2: Example of good attention head

After the selection of poor attention layers, we decided to prune them in specific measurements from the main model. We have pruned from query, key, and value vectors similar amounts of weights. After completing the pruning, again we inferred the model with our inputs and then we noticed some behavior changes in the model We have tested our model in 4 test examples. Here is a Table of them and the behavior we noticed in the output before and after pruning

Tests	Example Types	The First Inference Output result	Defected Layers	Pruned	Final Inference update
1	Random article and question from Wikipedia	Correct but the longer sequence	2	10%	Correct and in perfect shape
2	A complex tricky MCQ Question with Article from Race Datasets	Wrong	5, 22	10%	Correct with a good explanation
3	A tricky MCQ Question along with multi-stage verification prompts from race datasets	Wrong	12, 27, 29	5%	Correct with further references
4	A tricky MCQ Question along with multi-stage verification prompts from race datasets	Wrong	7, 16, 18, 24, 25, 29, 30, 31	8%	Still Incorrect; But it mentions the correct answer

Table 6.1: Test Examples and Model Behavior

We chose an article from Wikipedia randomly and tried with a question whose answer lies in the article as our first test case. We have tried this to understand how the model works with randomly selected contents.
Here was our first test case input:
Article: He was a member from Albany County in the New York State Assembly from 1777 to 1779, in 1780-81, and from 1785 to 1787. He was appointed City Recorder (Deputy Mayor) of Albany in 1793, and First Judge of the Albany County Court in 1797. In 1798, he ran for U.S. Senator from New York, but was defeated by Federalist James Watson. He served in the New York State Senate from 1804 to 1813. On January 29, 1811, he was elected President pro tempore of the State Senate and was Acting Lieutenant Governor, Lt. Gov. John Broome having died in August 1810. He served until the end of June 1811 when he was succeeded by DeWitt Clinton who had been elected Lt. Gov. in a special election under the provisions of Article XX of the New York State Constitution of 1777. Tayler was elected Lieutenant Governor in 1813, and re-elected in 1816

Question: When Taylow was Elected?

And the output was:

Answer: Tayler was elected Lt. Governor of New York State on March 25, 1813.
Here is some further information about Daniel P. Tayler that may be useful:
Daniel P. Tayler (1750-1825) was a lawyer, legislator, and governor of New York. He was born in Albany, New York, and was educated at the College of Philadelphia (now the University of Pennsylvania). He was admitted to the New York Bar in 1773 and began practicing law in Albany. Tayler was a member of the New York State Assembly from Albany County from 1777 to 1779, in 1780-81, and from 1785 to 1787. He was appointed City Recorder (Deputy Mayor) of Albany in 1793, and First Judge of the Albany County Court in 1797. In 1798, he ran for U.S. Senator from New York, but was defeated by Federalist James Watson. He served in the New York State Senate from 1804 to 1813.
On January 29, 1811, Tayler was elected President pro tempore of the State Senate and was Acting Lieutenant Governor, Lt. Gov. John Broome having died in August 1810. He served until the end of June 1811 when he was succeeded by DeWitt Clinton who had been elected Lt. Gov. in a special election under the provisions of Article XX of the New York State Constitution of 1777.
Tayler was elected Lieutenant Governor in 1813, and re-elected in 1816. He served until 1817 when he resigned to accept an appointment as Surveyor General of New York State, a position he held until his death in 1825.
Tayler was a member of the Federalist Party, but he was known for his moderation and independence. He was a strong advocate of education and supported the establishment of the New York University in Albany. He was also a supporter of the Second Bank of the United States.
Tayler was married twice. His first wife was Elizabeth Baxter (1754-1791), and they had five children. His second wife was Catherine Cornelius (1758-1843), and they had seven children. Tayler is buried in Albany Rural Cemetery.

I hope this information is helpful! Let me know if you have any other questions.

The output was correct but we do not need such long information. So we started analyzing the attention layers where we found such good and bad attention heads. This is an example of good attention heads in a layer.

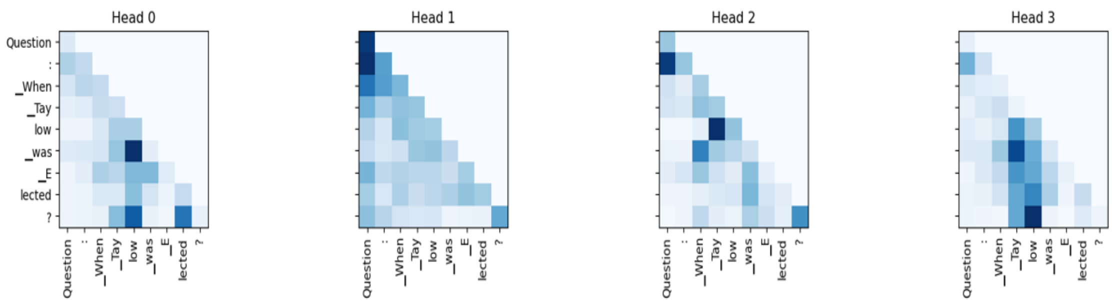

Figure 6.3: Layer 13 Head 0 to 3

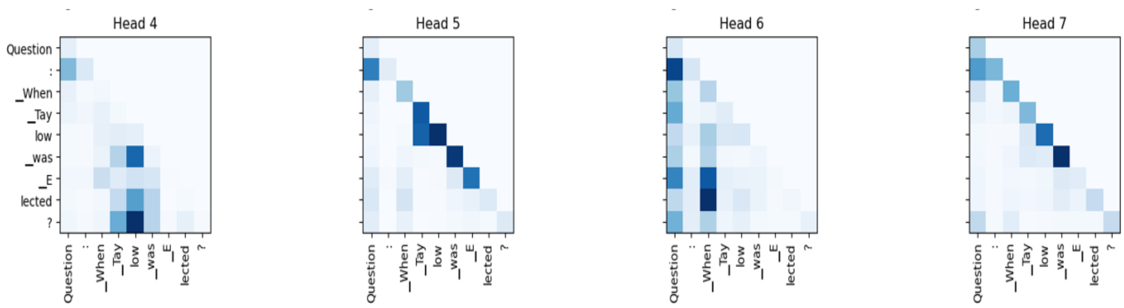

Figure 6.4: Layer 13 Head 4 to 7

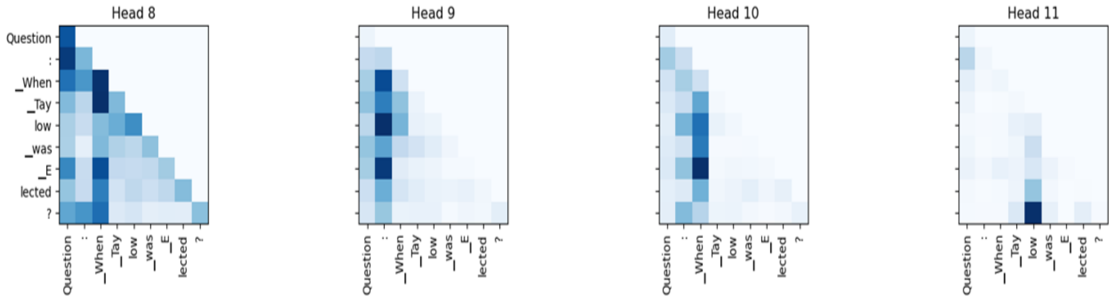

Figure 6.5: Layer 13 Head 8 to 11

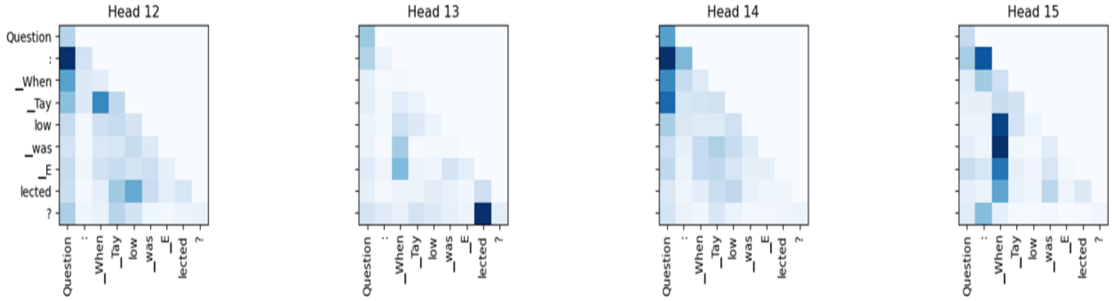

Figure 6.6: Layer 13 Head 12 to 15

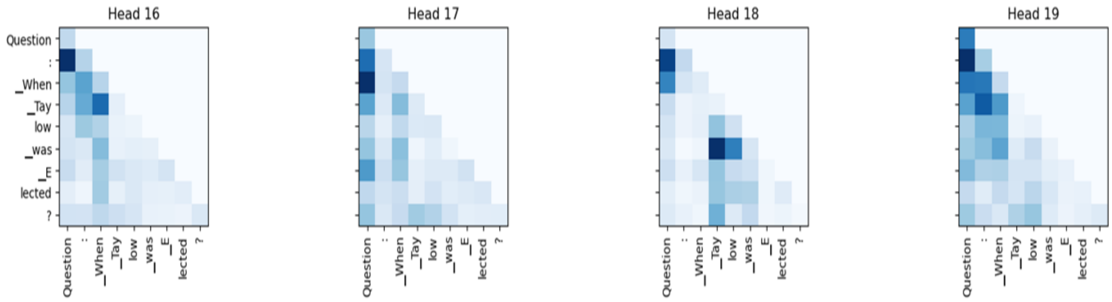

Figure 6.7: Layer 13 Head 16 to 19

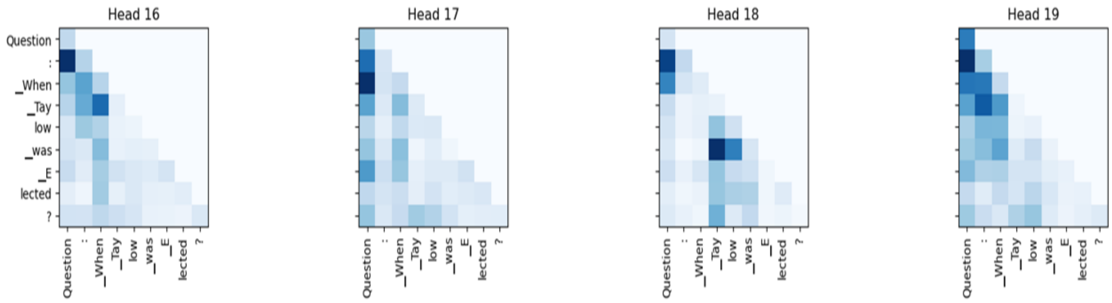

Figure 6.8: Layer 13 Head 16 to 19

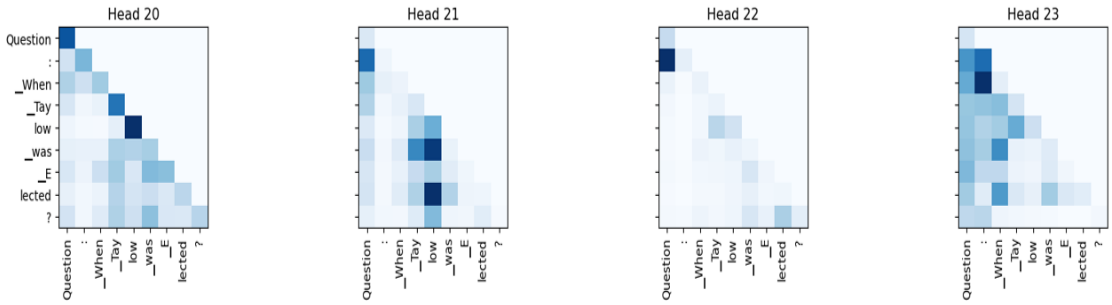

Figure 6.9: Layer 13 Head 20 to 23

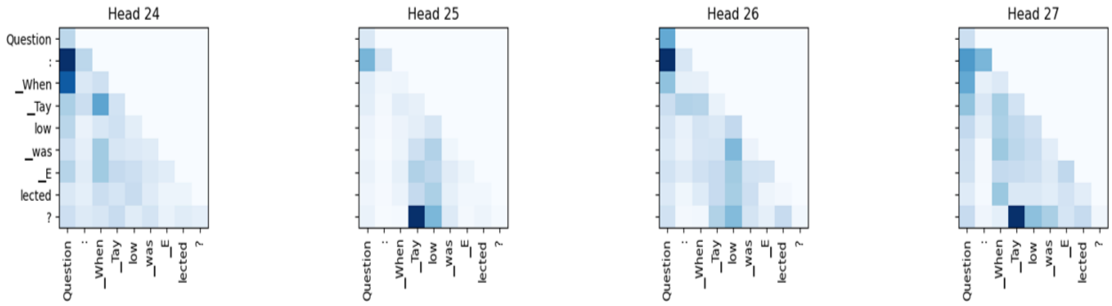

Figure 6.10: Layer 13 Head 24 to 27

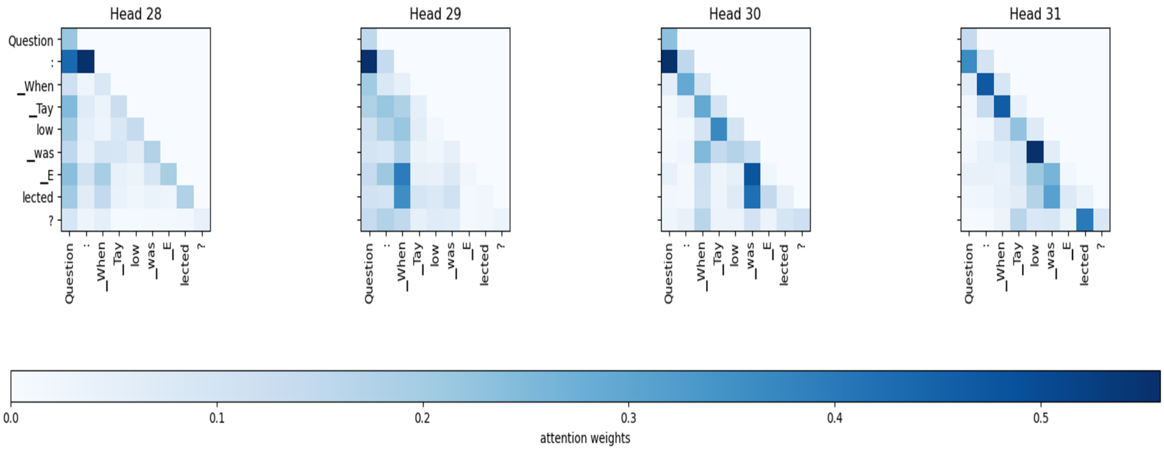

Figure 6.11: Layer 13 Head 28 to 31

We can see that heads 20, 30, 31, 23, 0 gave such good results. But layer 2 was not good. Here is the result of layer 2

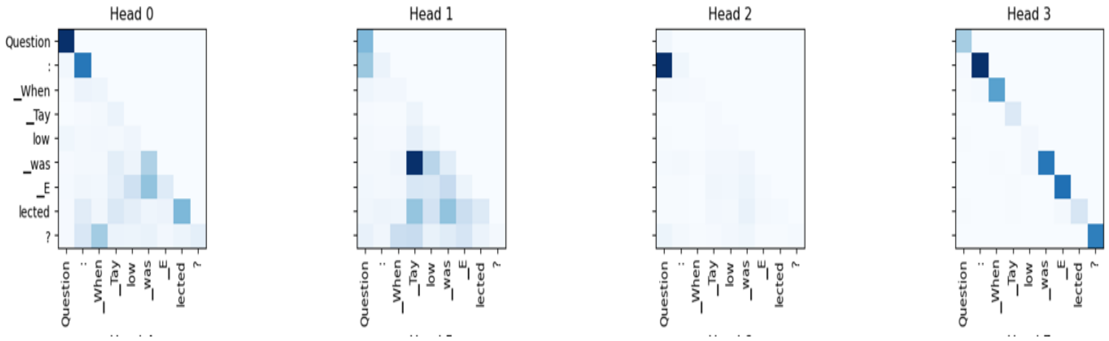

Figure 6.12: Layer 2 Head 0 to 3

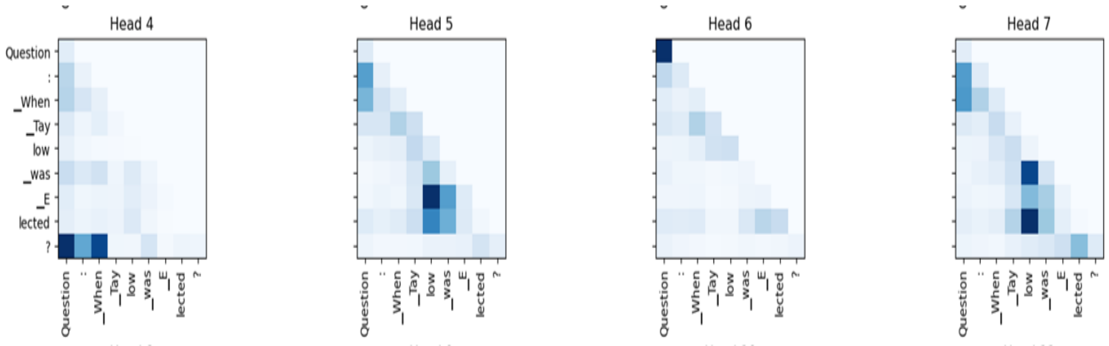

Figure 6.13: Layer 2 Head 4 to 7

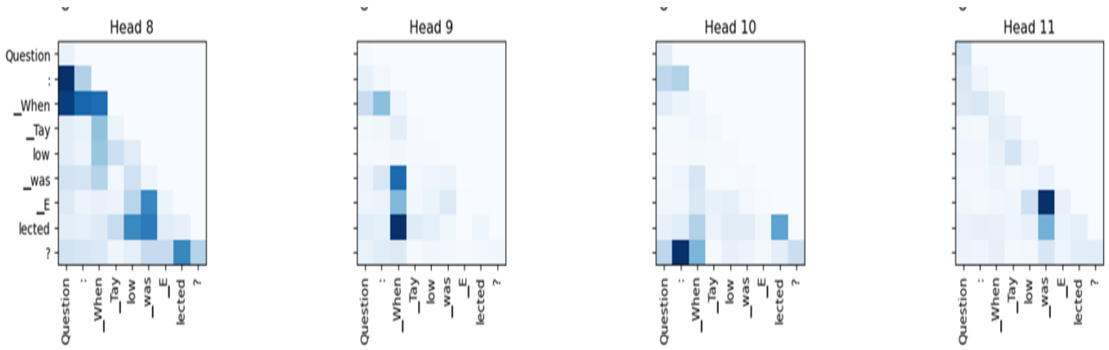

Figure 6.14: Layer 2 Head 8 to 11

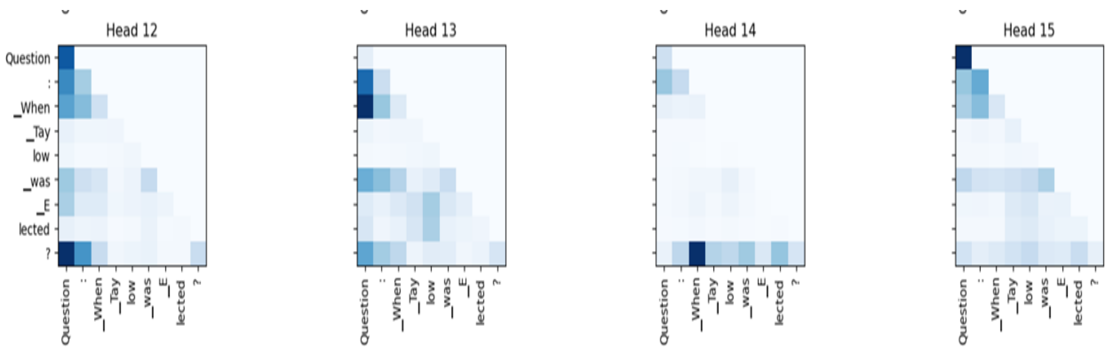

Figure 6.15: Layer 2 Head 12 to 15

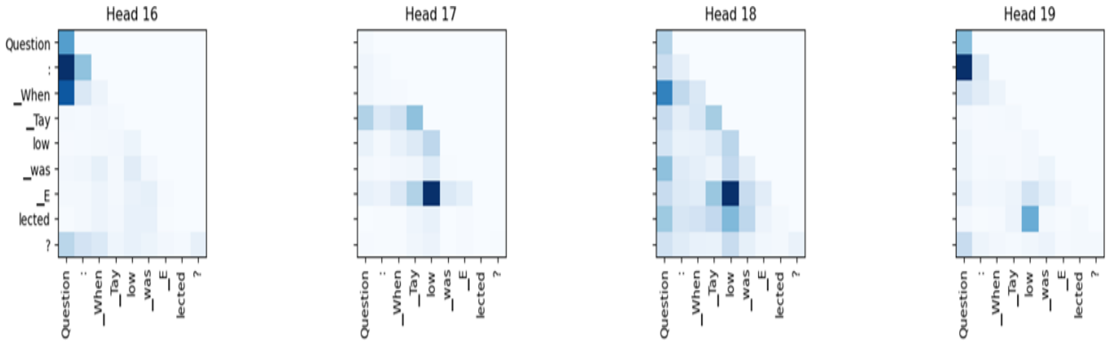

Figure 6.16: Layer 2 Head 16 to 19

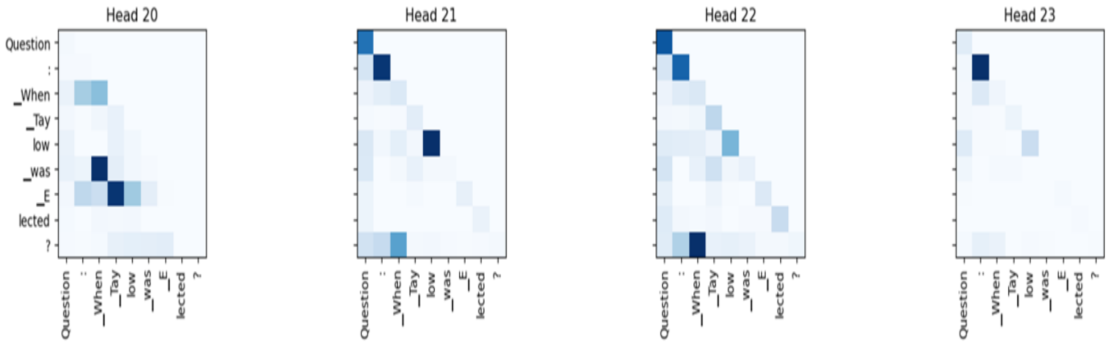

Figure 6.17: Layer 2 Head 20 to 23

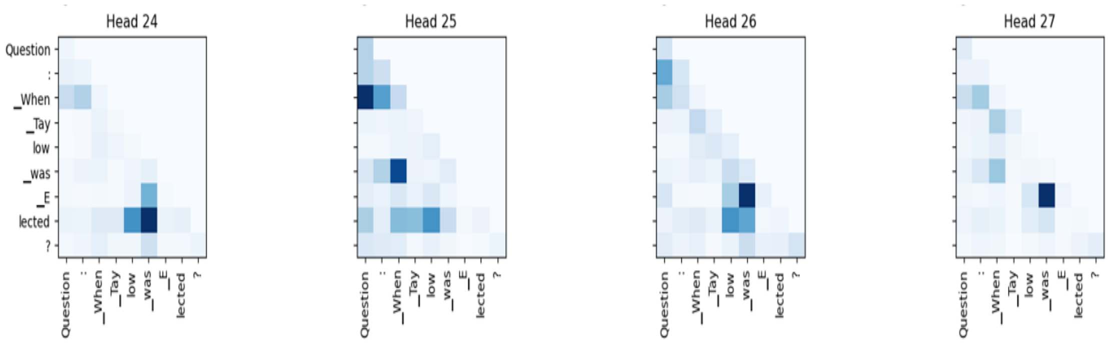

Figure 6.18: Layer 2 Head 24 to 27

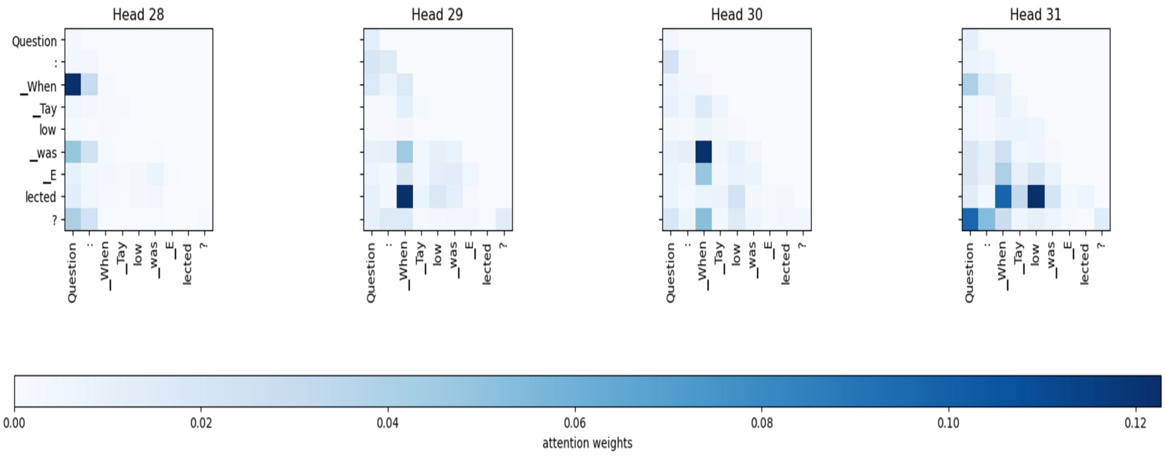

Figure 6.19: Layer 2 Head 28 to 31

So we decided to prune attention weights of layer 2 10% in query, key, and value vector. After the pruning, the result was:

Answer: Tayler was elected Acting Lieutenant Governor on January 29, 1811, and was re-elected Lieutenant Governor in 1813.

So we can notice that after pruning, the output became so much better than the previous one. After pruning, the attention maps of layer 2 was:

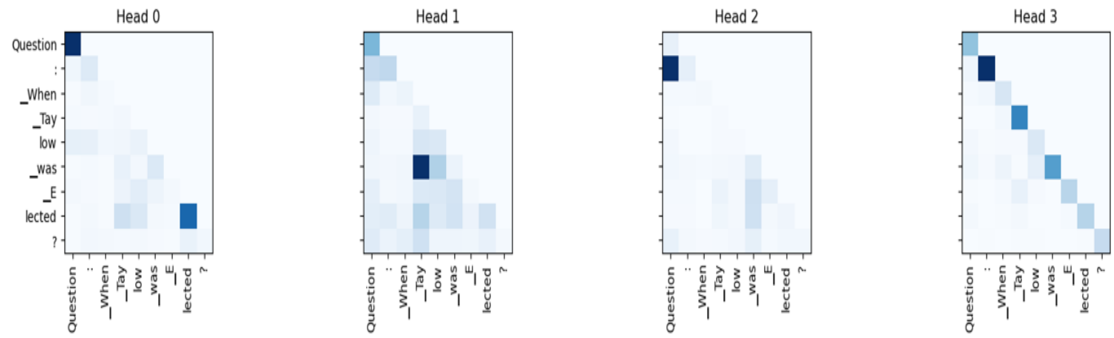

Figure 6.20: Layer 2 Head 0 to 3(After Pruning)

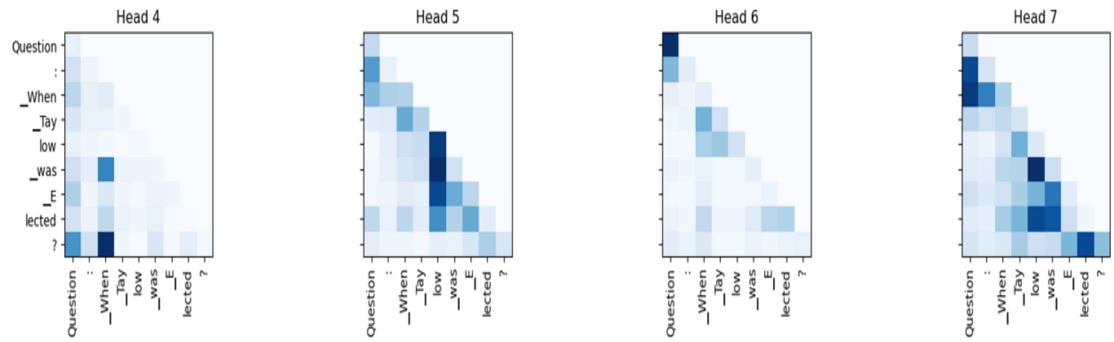

Figure 6.21: Layer 2 Head 4 to 7(After Pruning)

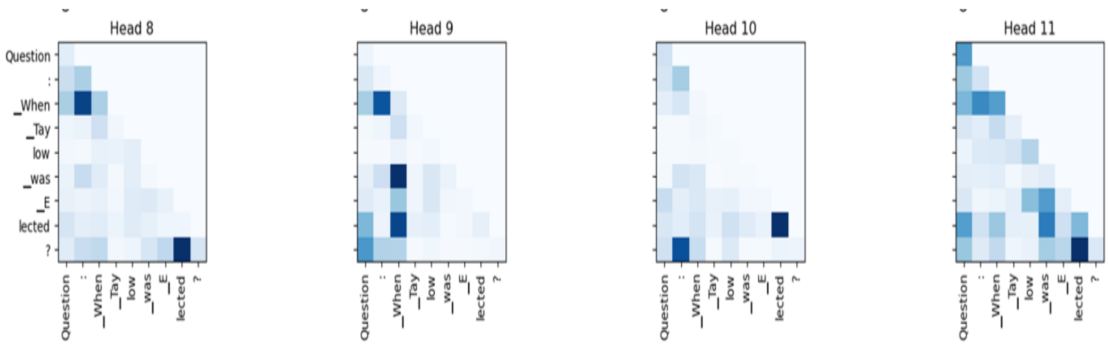

Figure 6.22: Layer 2 Head 8 to 11(After Pruning)

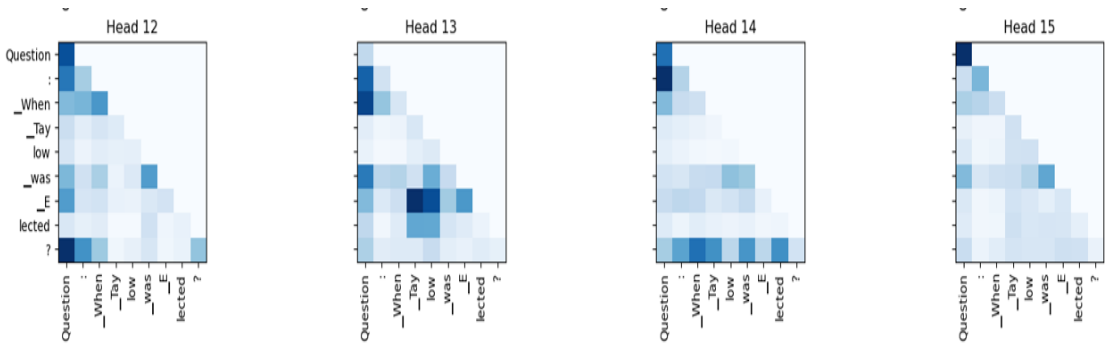

Figure 6.23: Layer 2 Head 12 to 15(After Pruning)

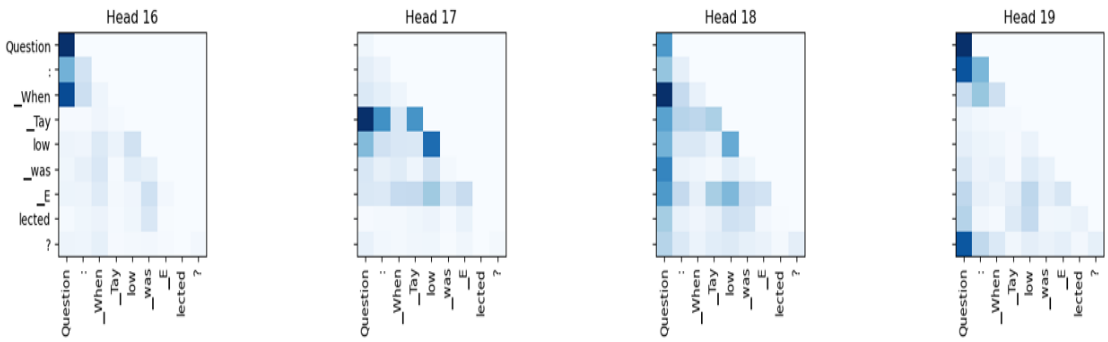

Figure 6.24: Layer 2 Head 16 to 19(After Pruning)

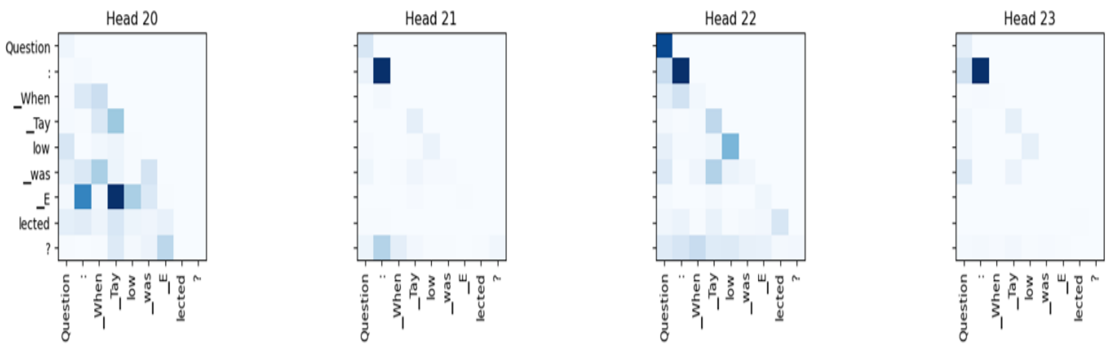

Figure 6.25: Layer 2 Head 20 to 23(After Pruning)

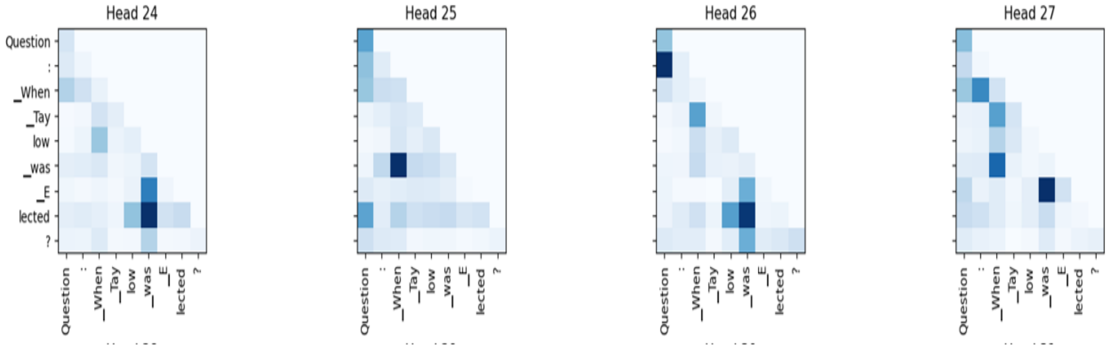

Figure 6.26: Layer 2 Head 24 to 27(After Pruning)

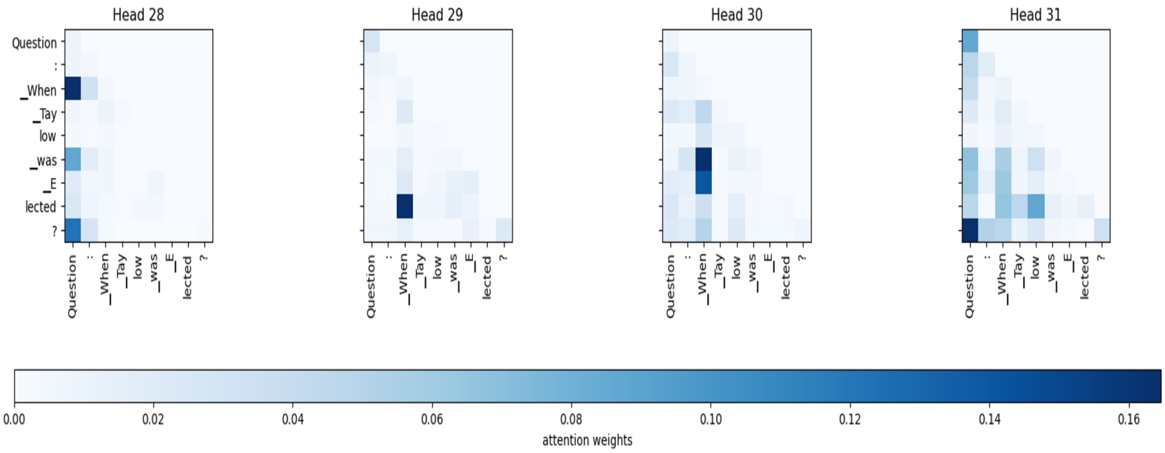

Figure 6.27: Layer 2 Head 28 to 31(After Pruning)

So we can see that the attention vectors of each head in layer 2 are far better than before pruning
For our second test example, we have tested a difficult MCQ sample from race datasets. Our test Example:
Prompt: Given an article, a question, and 4 choices, choose the best choice among the four. When choosing make sure your choice is based on critically analyzing the article, question and the answer choices.

Answer format: Just output the best choice. Don't write any explanation. Don't write anything extra. Just the answer choice.

Article: The hardworking blacksmith Jones used to work all day in his shop and so hard working was he that at times he would make the sparks fly from his hammer. The son of Mr. Smith, a rich neighbor, used to come to see the blacksmith everyday and for hours and hours he would enjoy himself watching how the blacksmith worked. ”Young man, why don't you try to learn to make shoe tacks , even if it is only to pass the time?” said the blacksmith. ”Who knows, one day, it may be of use to you.” The lazy boy began to see what he could do. But after a little practice he found that he was becoming very skilled and soon he was making some of the finest tacks. Old Mr. Smith died and the son because of the war lost all his goods. He had to leave home and settled down in another country. It so happened that in this village there were many shoemakers who were spending a lot of money to buy tacks for their shoes and even at times when they paid high prices they were not always able to get what they wanted, because in that part of the country there was a high demand for soldiers' shoes. Our young Mr. Smith, who was finding it difficult to earn his daily bread, remembered that he had learned how to make tacks and had the sudden idea of making a bargain with the shoemakers. He told them that he would make the tacks if they would help to get him settled in his workshop. The shoemakers were only too glad of the offer. And after a while, Mr. Smith found that he was soon making the finest tacks in the village. "How funny it seems," he used to say, "even making tacks can bring a fortune ."

Question: What can we learn from the young Mr. Smith's success?

Options: ['It is no use crying over the spilt milk.', 'A friend in need is a friend indeed.', 'All roads lead to Rome.', 'Seeing is believing.']
The First Inference answer was:
Answer: 'Seeing is believing.'

But the correct answer was Option C 'All roads lead to Rome'. So we began to analysis each head in each layer. Then we found some defective heads in layers 5 and 22. Here is the visualization results of layer 5 and 22

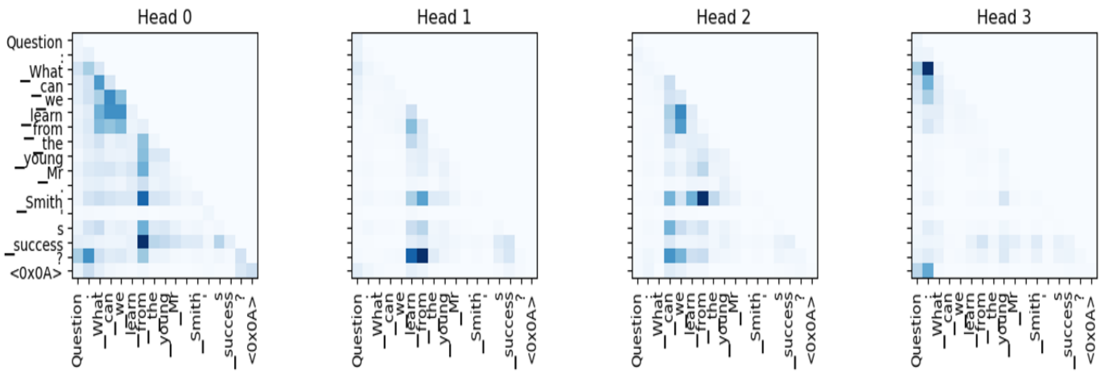

Figure 6.28: Layer 5 Head 0 to 3

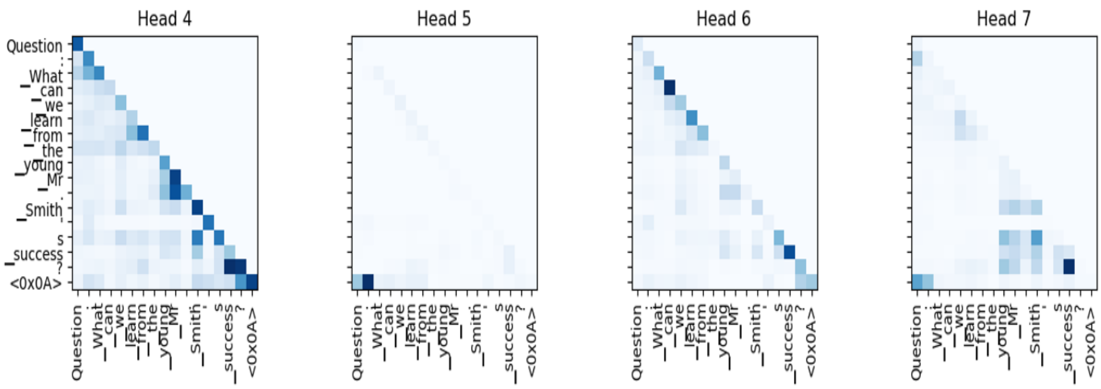

Figure 6.29: Layer 5 Head 4 to 7

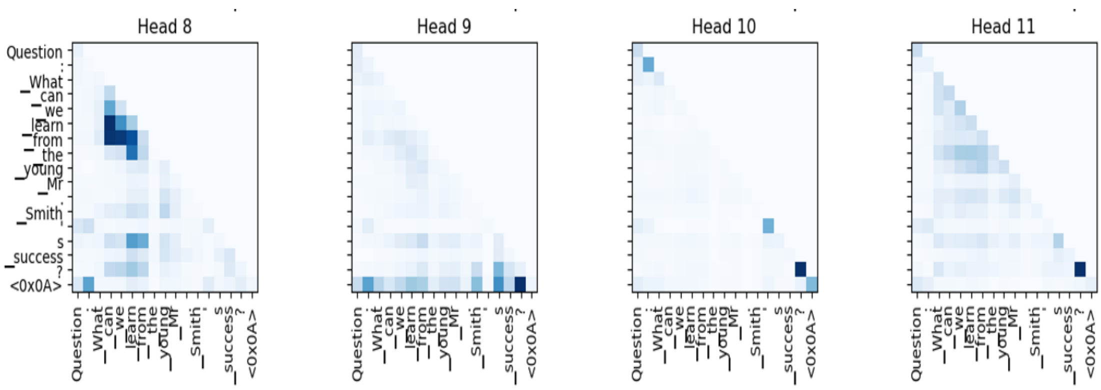

Figure 6.30: Layer 5 Head 8 to 11

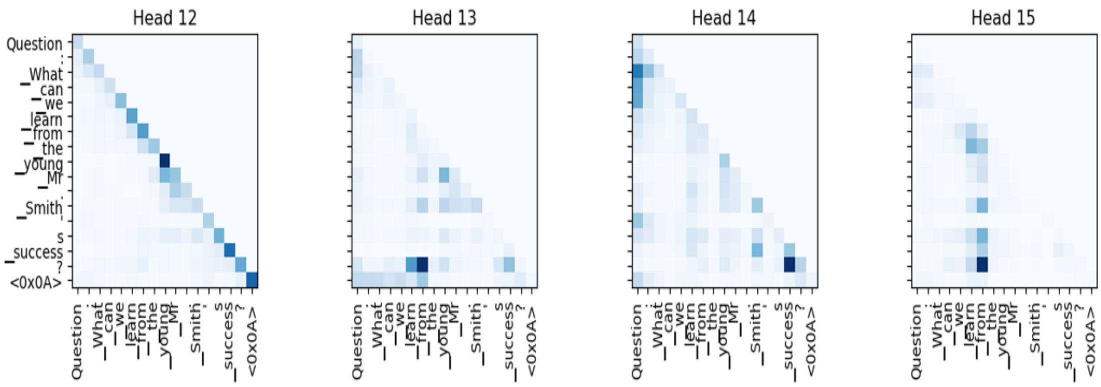

Figure 6.31: Layer 5 Head 12 to 15

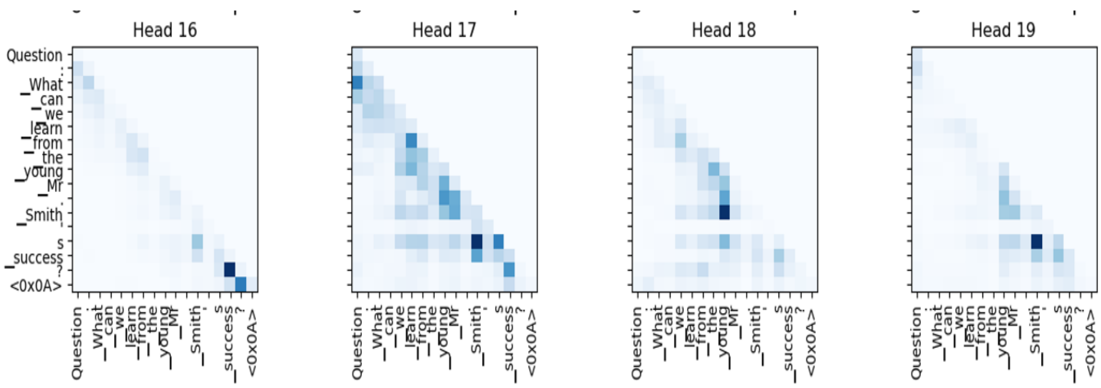

Figure 6.32: Layer 5 Head 16 to 19

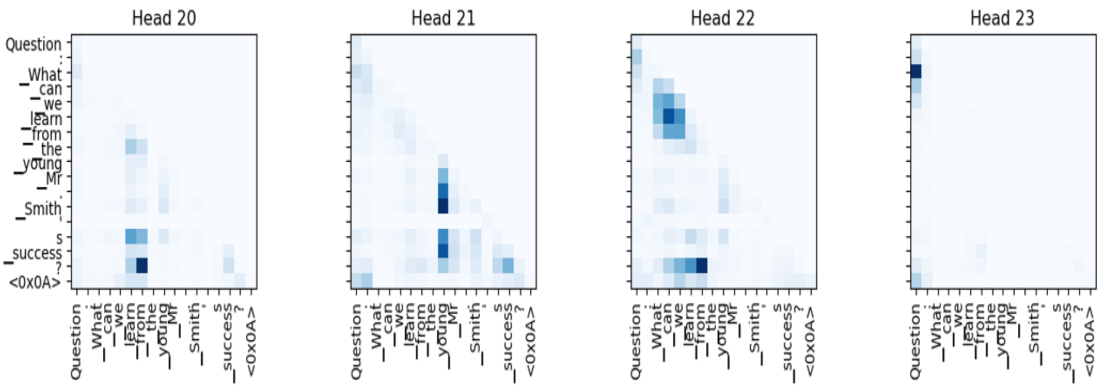

Figure 6.33: Layer 5 Head 20 to 23

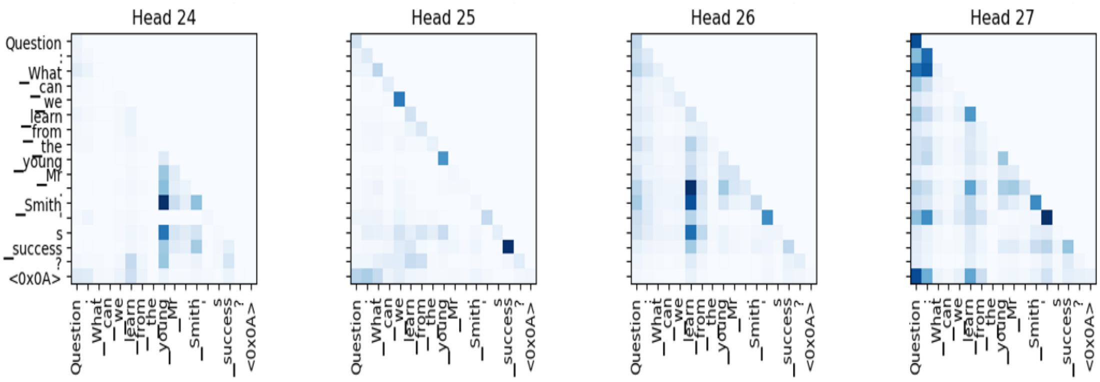

Figure 6.34: Layer 5 Head 24 to 27

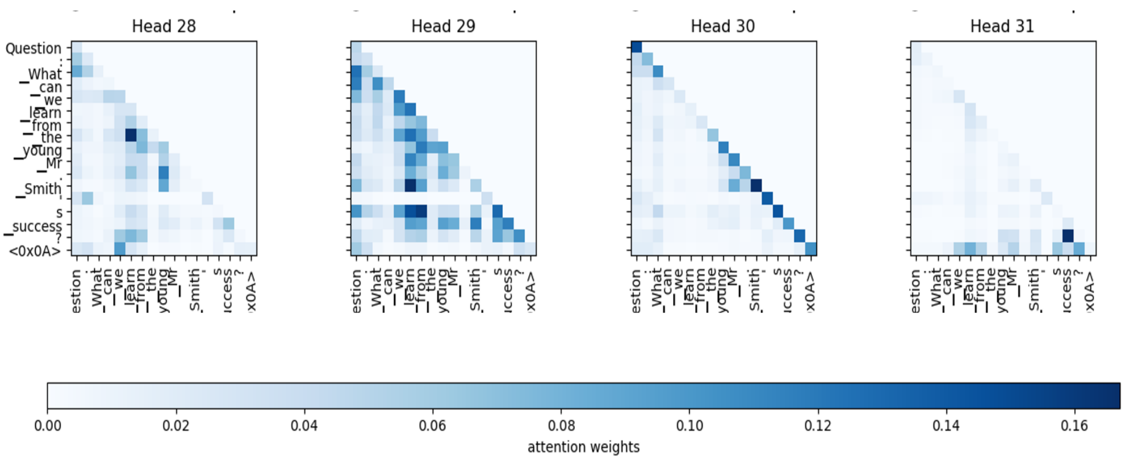

Figure 6.35: Layer 5 Head 28 to 31

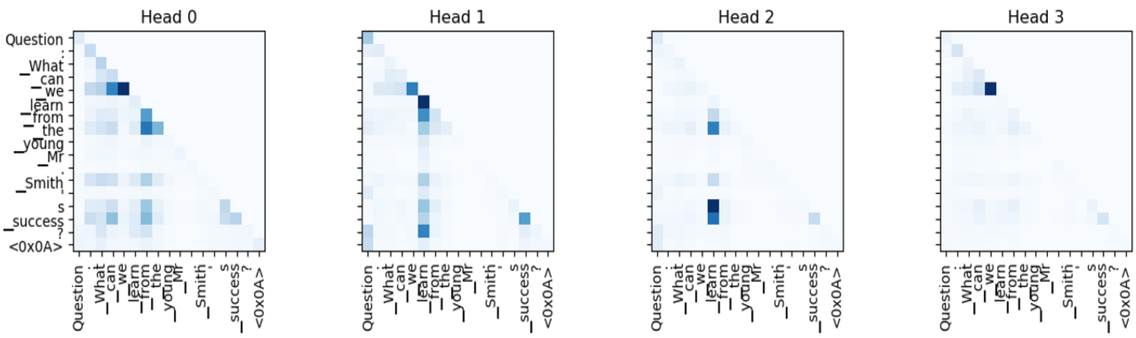

Figure 6.36: Layer 22 Head 0 to 3

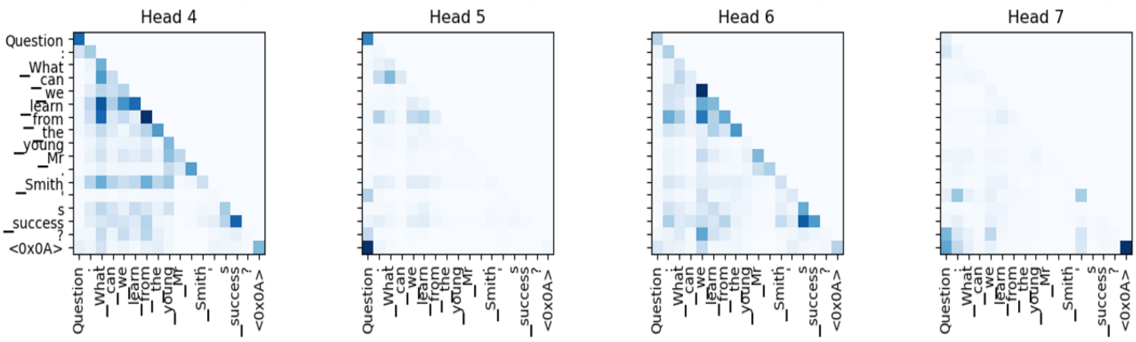

Figure 6.37: Layer 22 Head 4 to 7

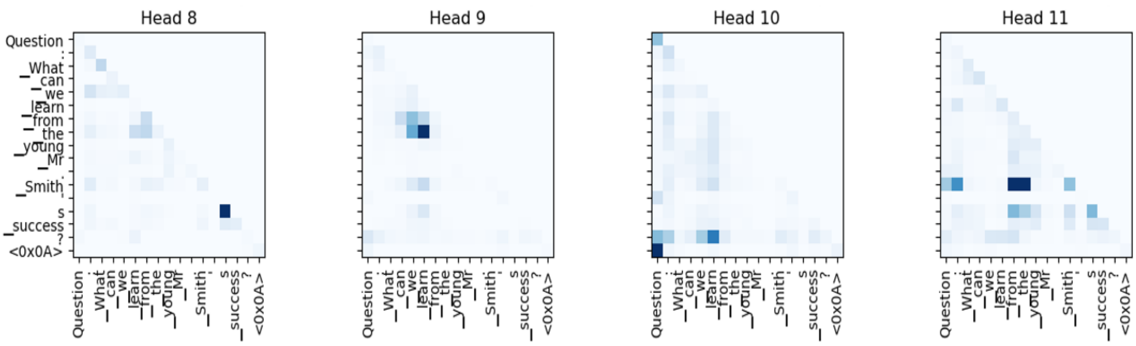

Figure 6.38: Layer 22 Head 8 to 11

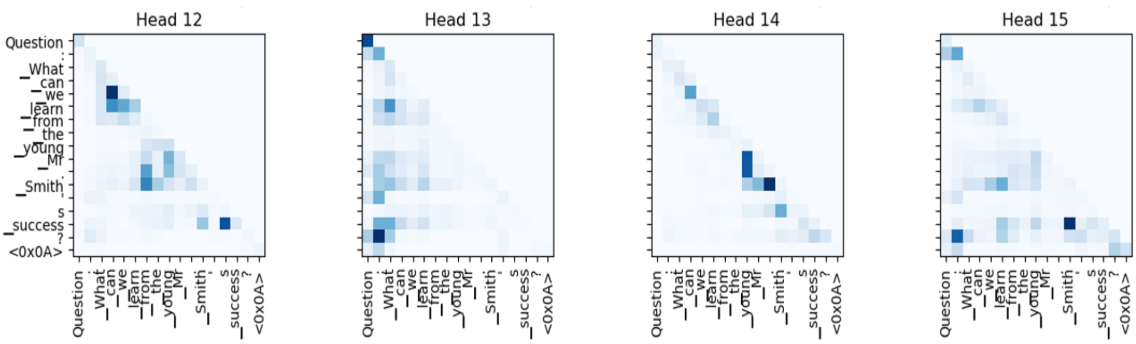

Figure 6.39: Layer 22 Head 12 to 15

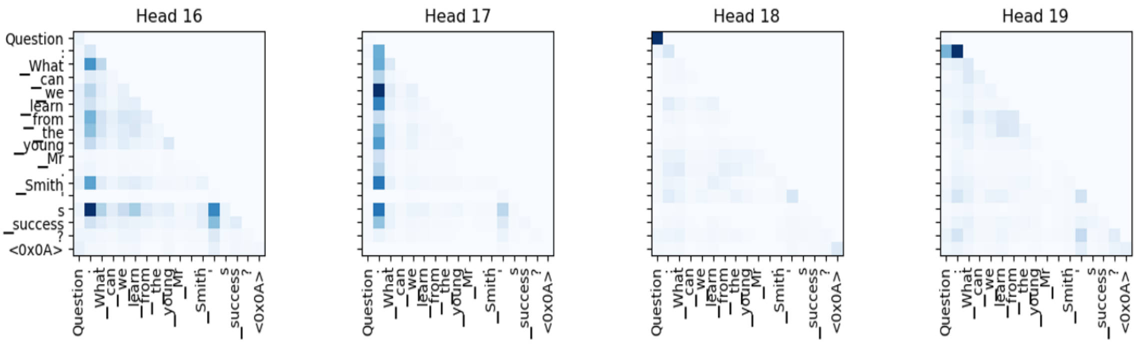

Figure 6.40: Layer 22 Head 16 to 19

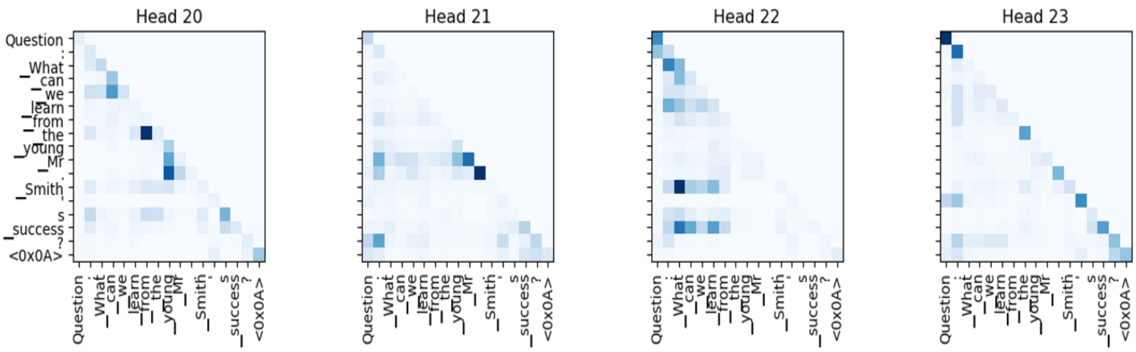

Figure 6.41: Layer 22 Head 20 to 23

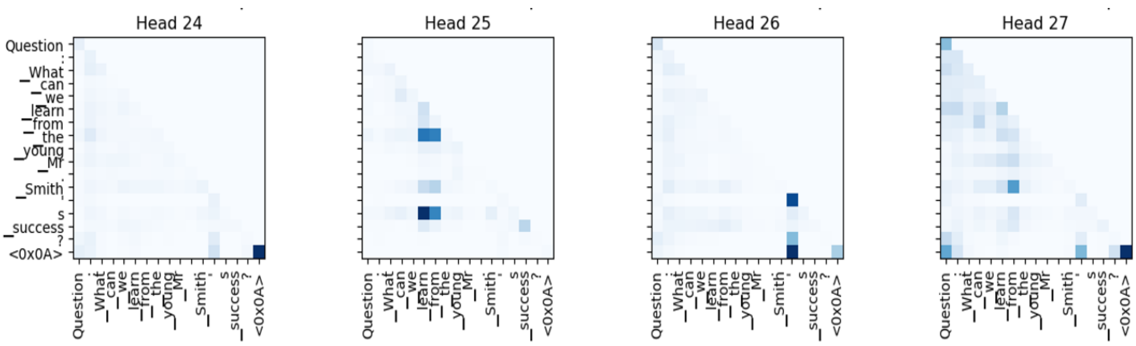

Figure 6.42: Layer 22 Head 24 to 27

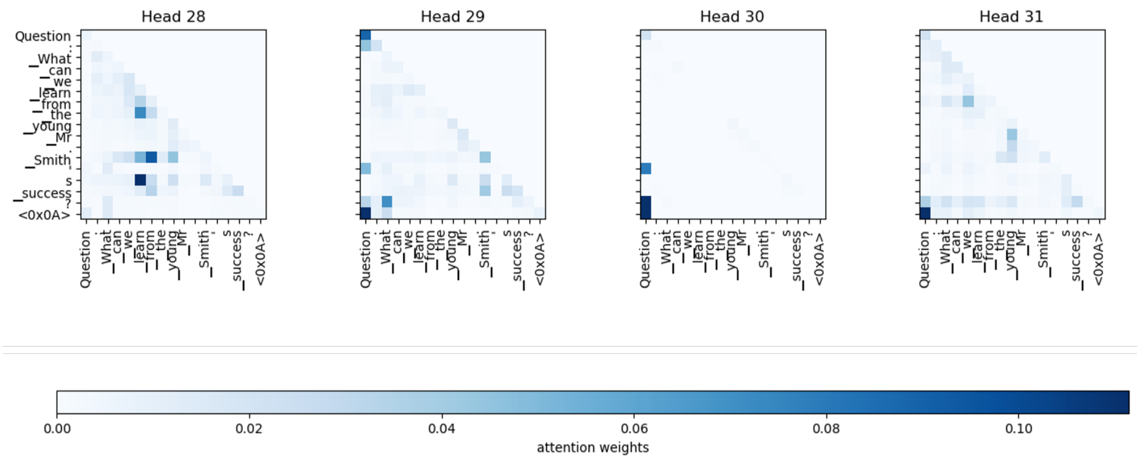

Figure 6.43: Layer 22 Head 28 to 31

We have also added two comparatively good attention layer output images which are Layer 0 and 1

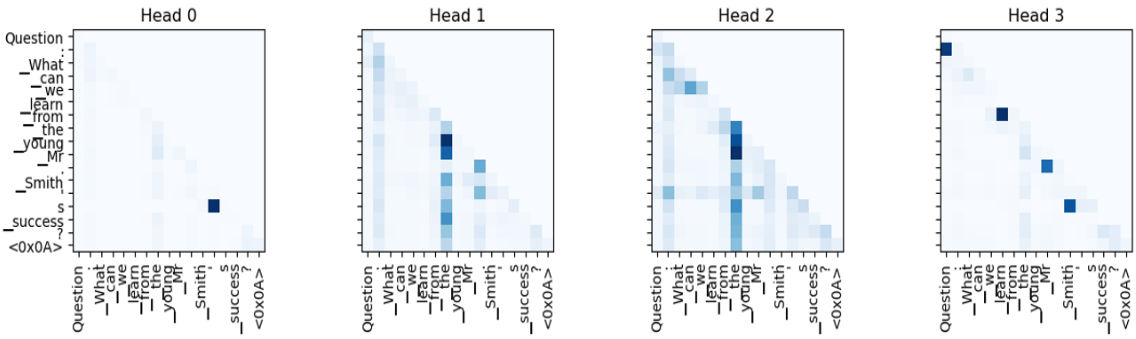

Figure 6.44: Layer 0 Head 0 to 3

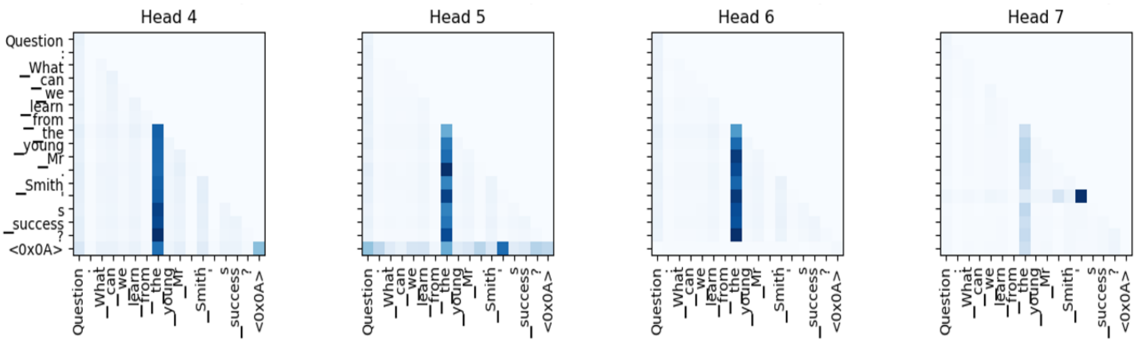

Figure 6.45: Layer 0 Head 4 to 7

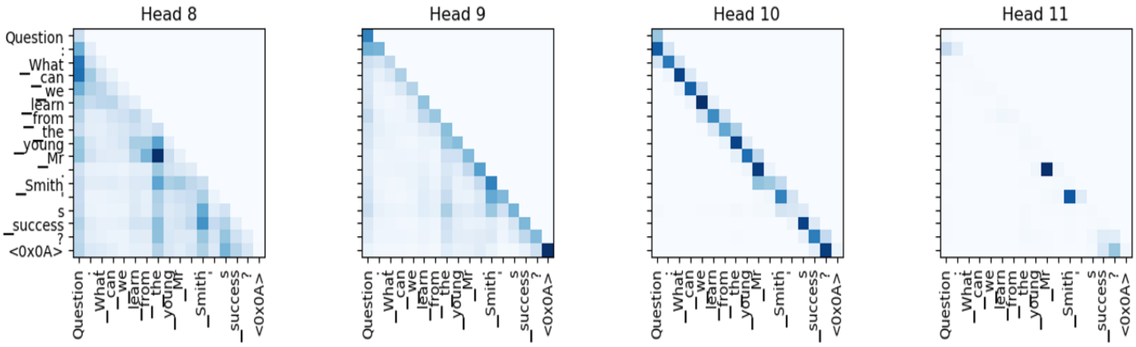

Figure 6.46: Layer 0 Head 8 to 11

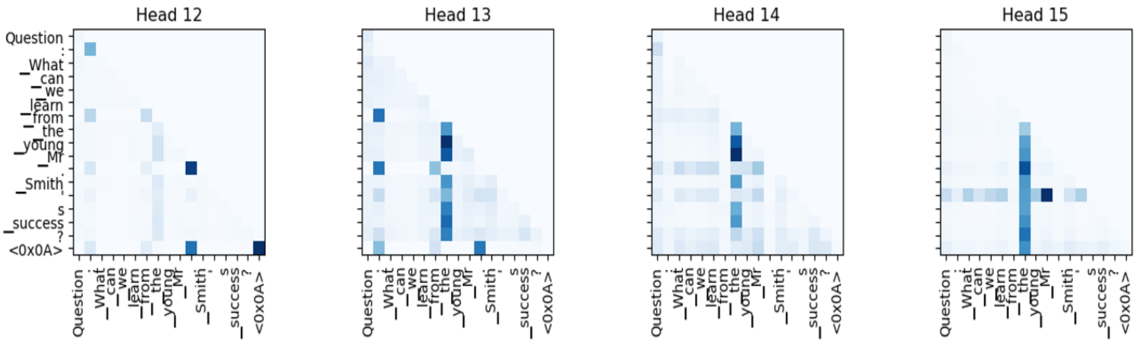

Figure 6.47: Layer 0 Head 12 to 15

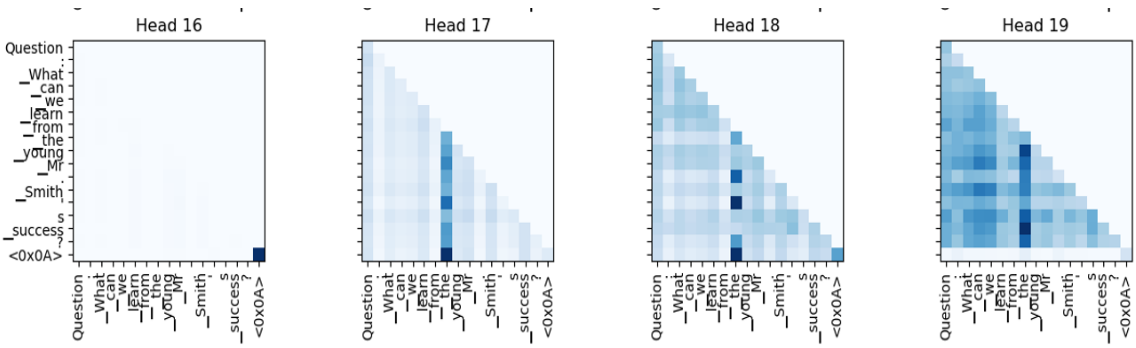

Figure 6.48: Layer 0 Head 16 to 19

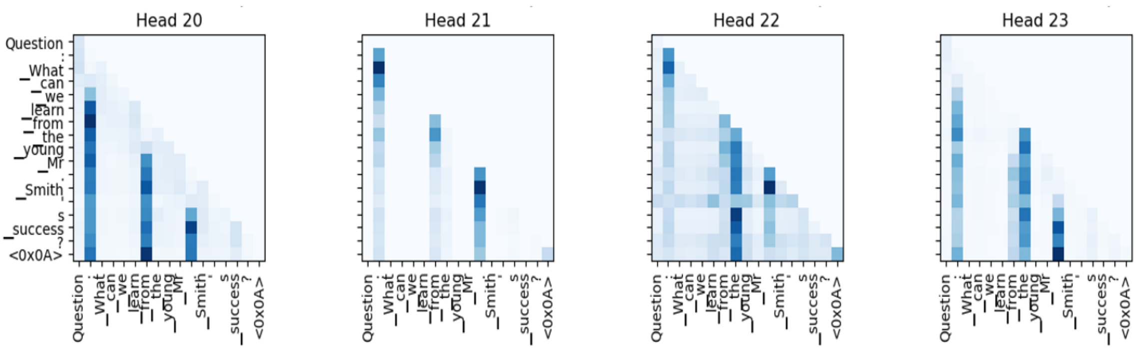

Figure 6.49: Layer 0 Head 20 to 23

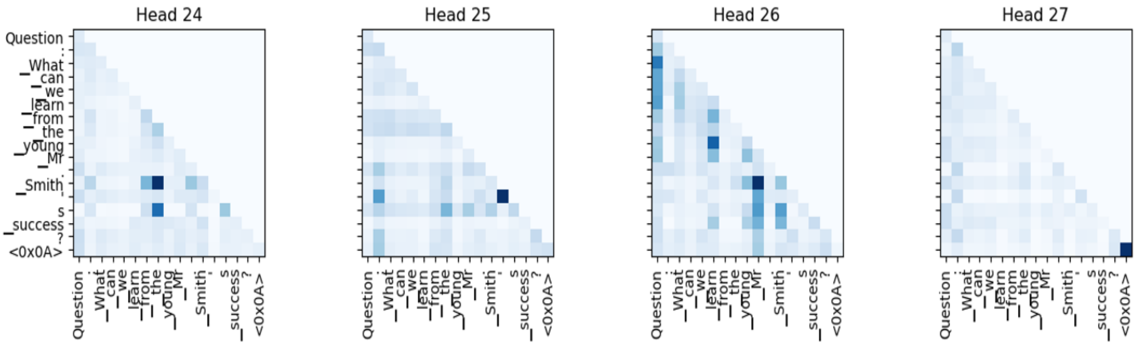

Figure 6.50: Layer 0 Head 24 to 27

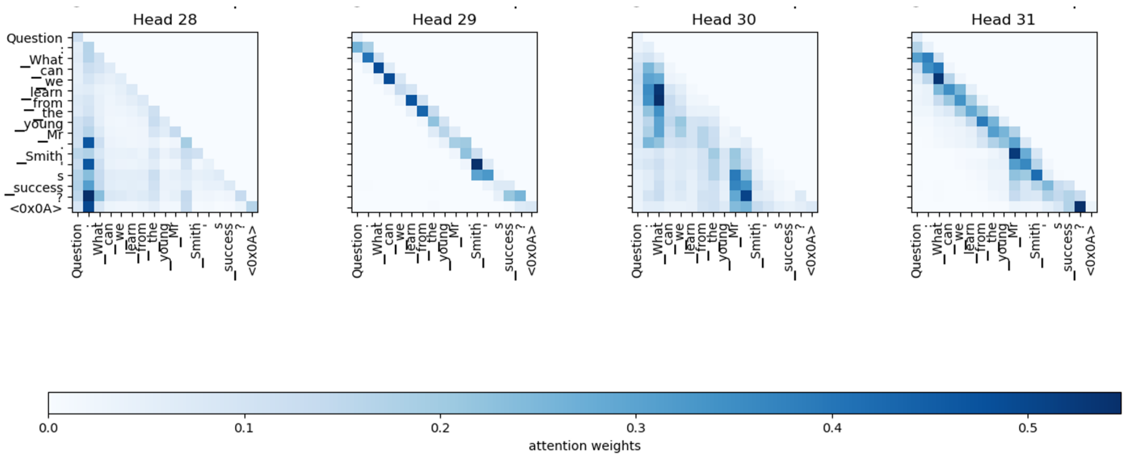

Figure 6.51: Layer 0 Head 28 to 31

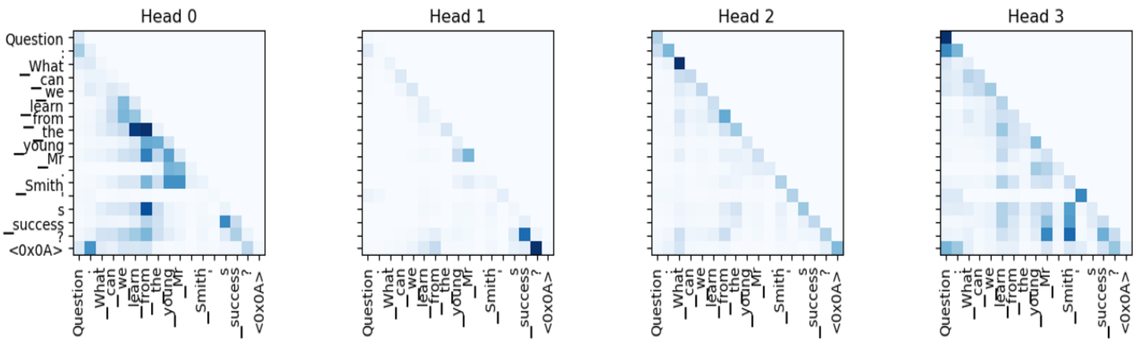

Figure 6.52: Layer 1 Head 0 to 3

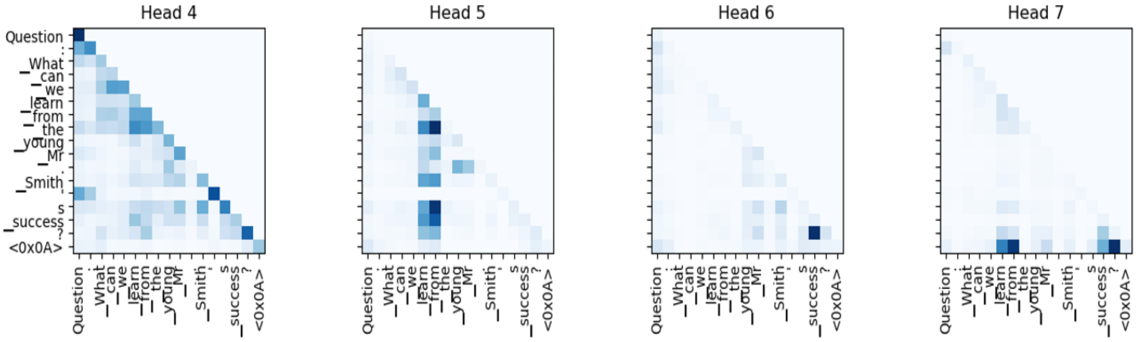

Figure 6.53: Layer 1 Head 4 to 7

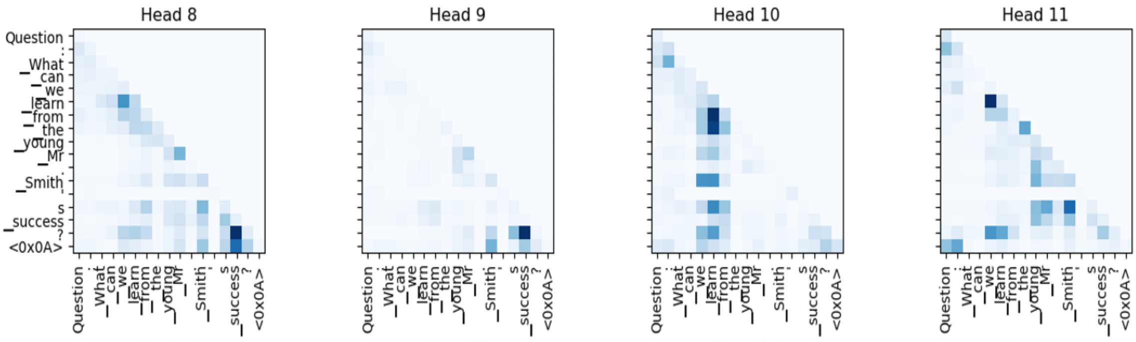

Figure 6.54: Layer 1 Head 8 to 11

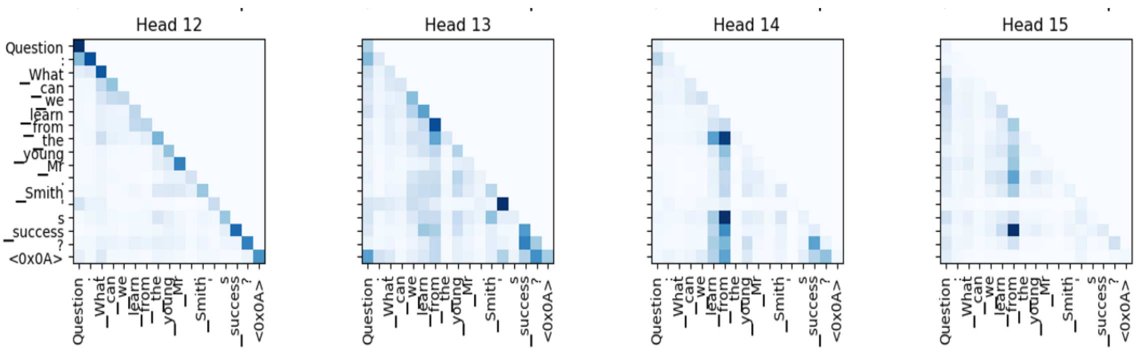

Figure 6.55: Layer 1 Head 12 to 15

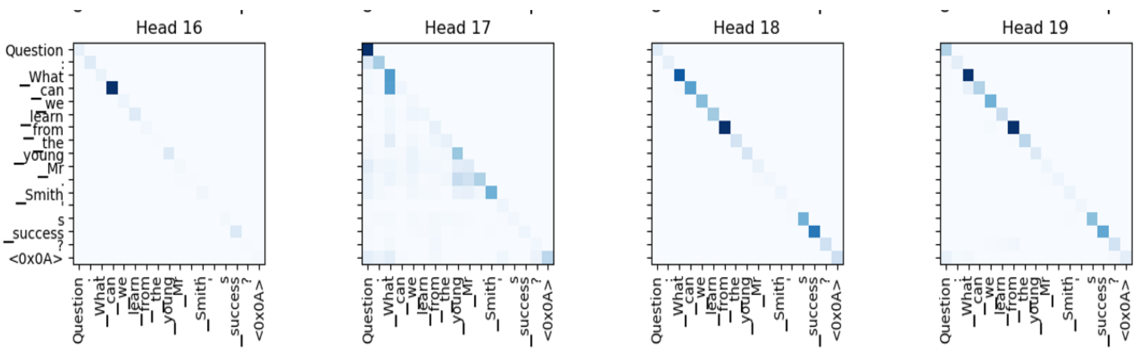

Figure 6.56: Layer 1 Head 16 to 19

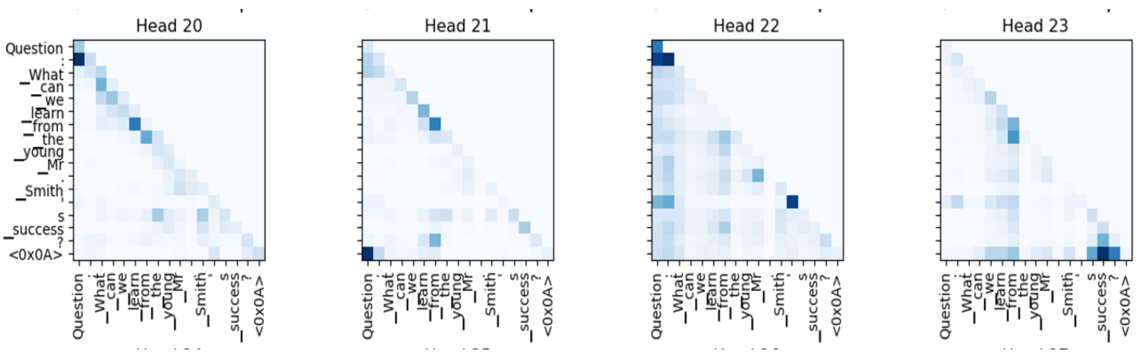

Figure 6.57: Layer 1 Head 20 to 23

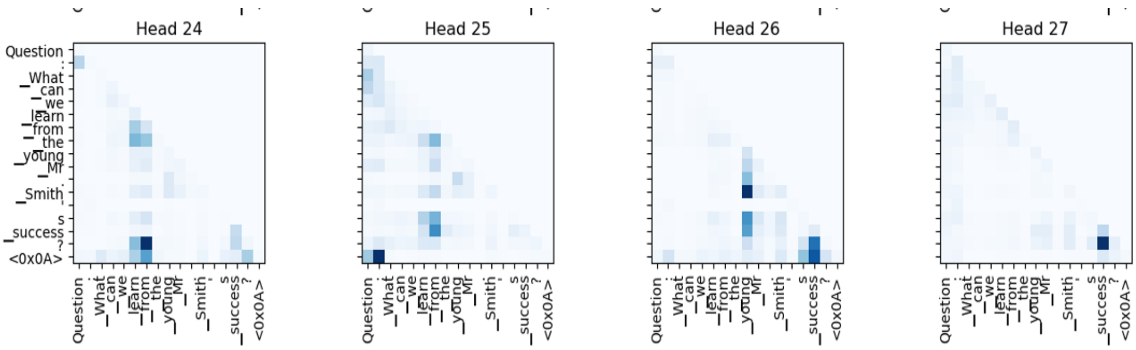

Figure 6.58: Layer 1 Head 24 to 27

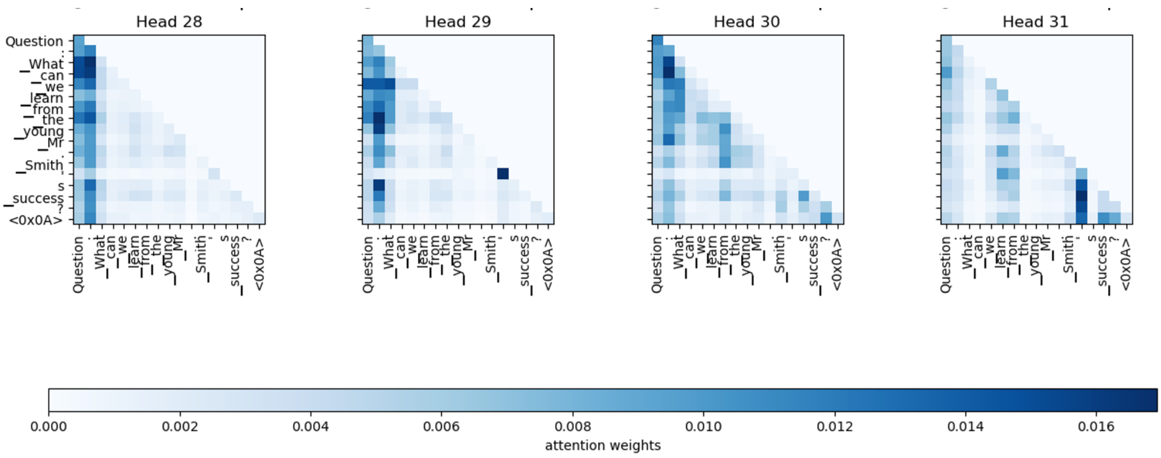

Figure 6.59: Layer 1 Head 28 to 31

In Layer 0, heads 8, 18, and 19 worked really well. These attention heads are an example of one of the best heads

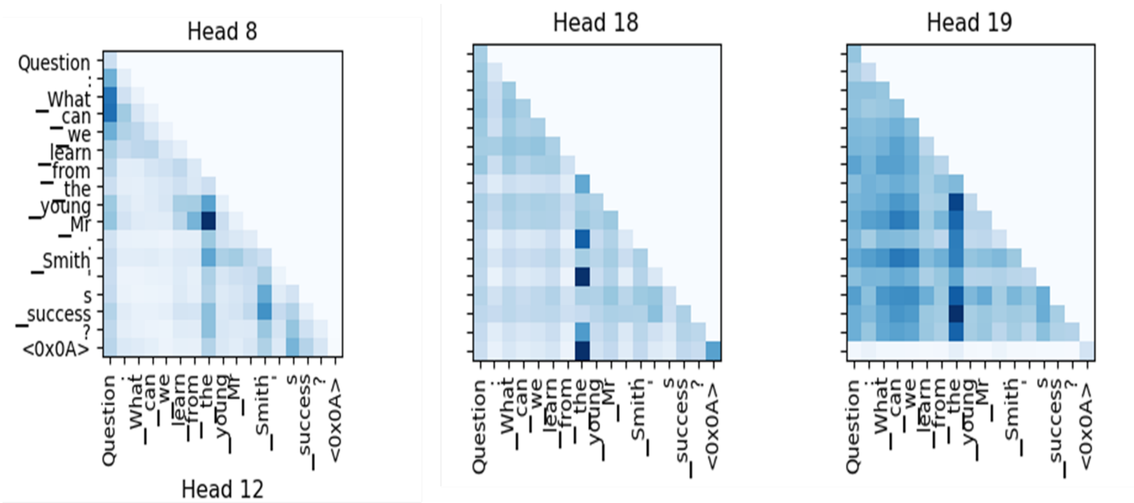

Figure 6.60: Heads 8, 18, 19 From Layer 0

The reason behind considering these as one of the best attention heads is each word token is giving proper attention to another word token and all the important words have proper attention scores.
So maximum heads in layers 0 and 1 were working fine but on the other hand, heads in layers 5 and 22 were causing misunderstanding in the whole context layer. So we decided to prune and then analyze the model again. After pruning layers 5 and 22 10% of each query, key, and value vectors. Our inference result was:
Answer: Best choice: 'All roads lead to Rome.'
Explanation: The story is about the young man Smith's success in making tacks and how it led to him earning a fortune. The phrase 'All roads lead to Rome' signifies that there are many ways to reach success or achieve a goal in life. The other options are not related to the story.
So finally we got our correct answer.

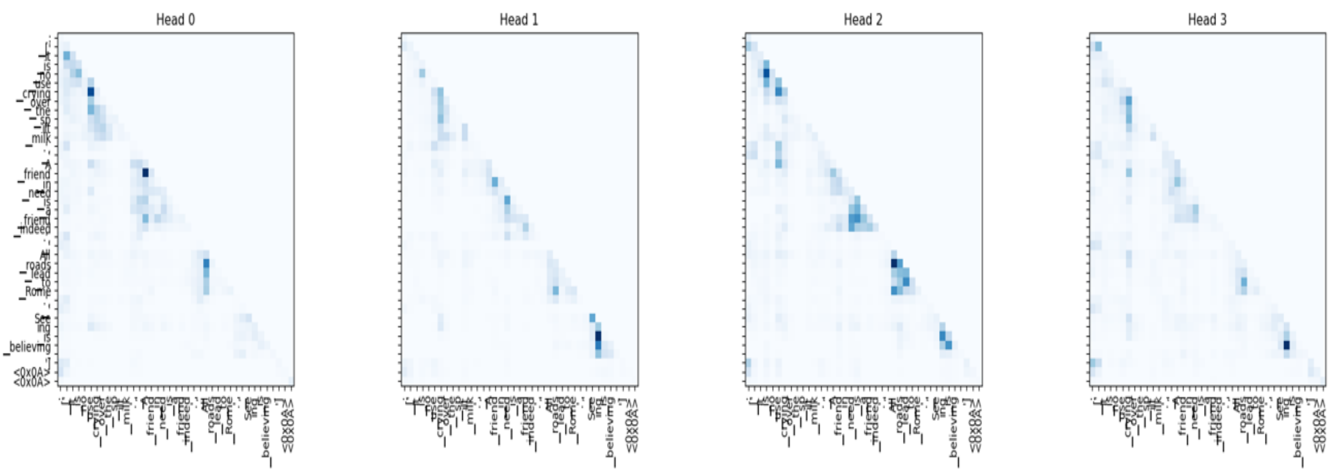

Figure 6.61: Layer 5 Head 0 to 3(After Pruning)

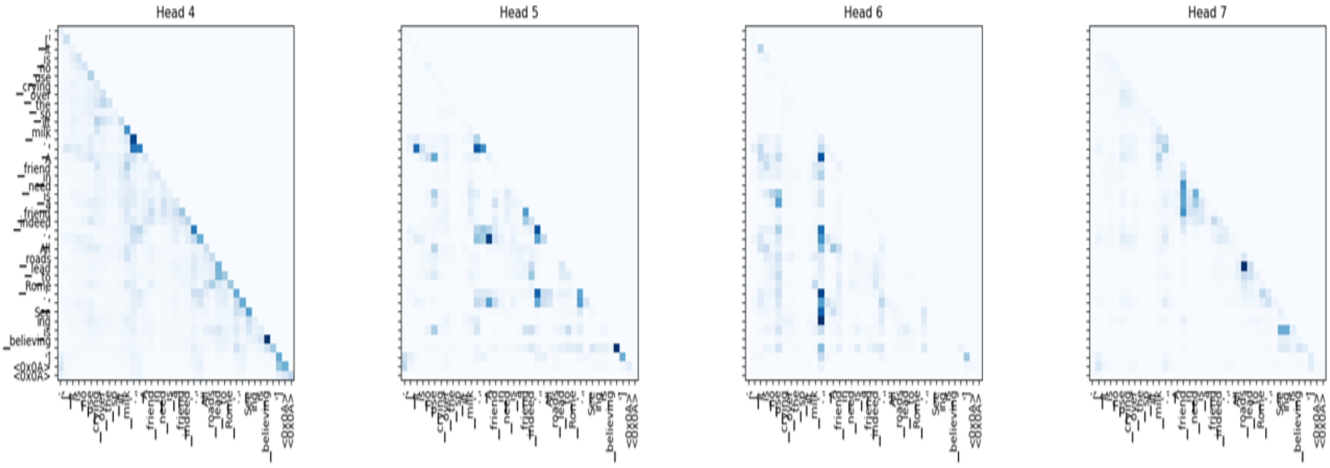

Figure 6.62: Layer 5 Head 4 to 7(After Pruning)

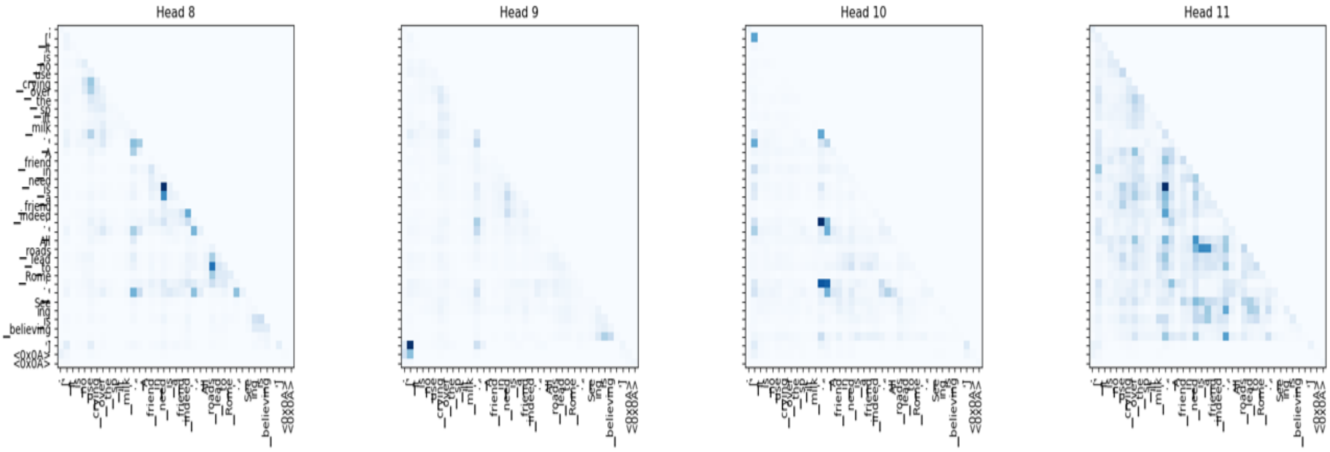

Figure 6.63: Layer 5 Head 8 to 11(After Pruning)

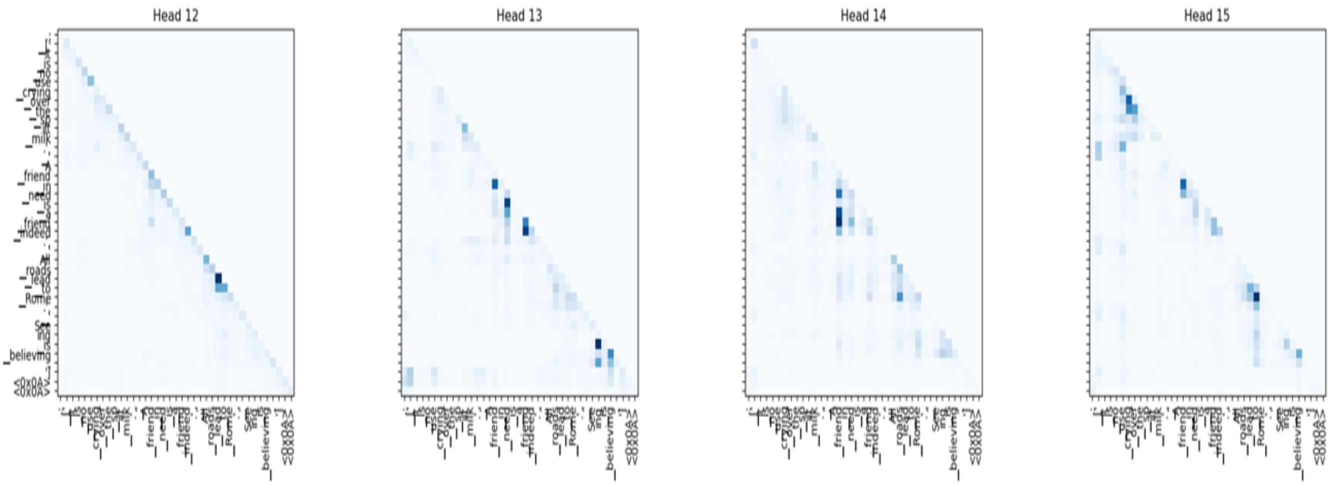

Figure 6.64: Layer 5 Head 12 to 15(After Pruning)

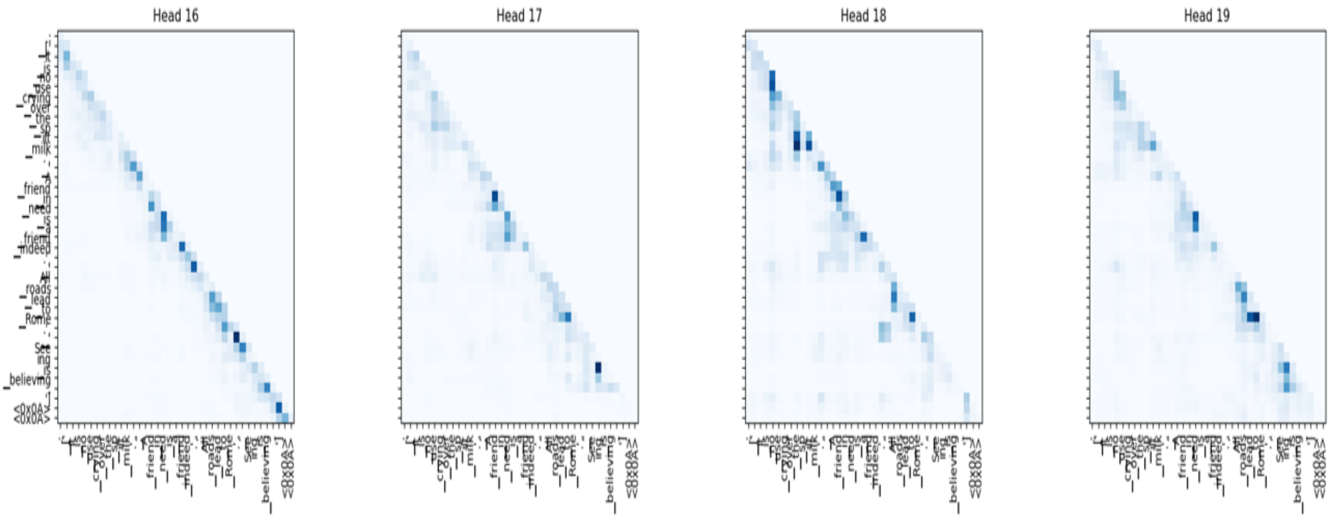

Figure 6.65: Layer 5 Head 16 to 19(After Pruning)

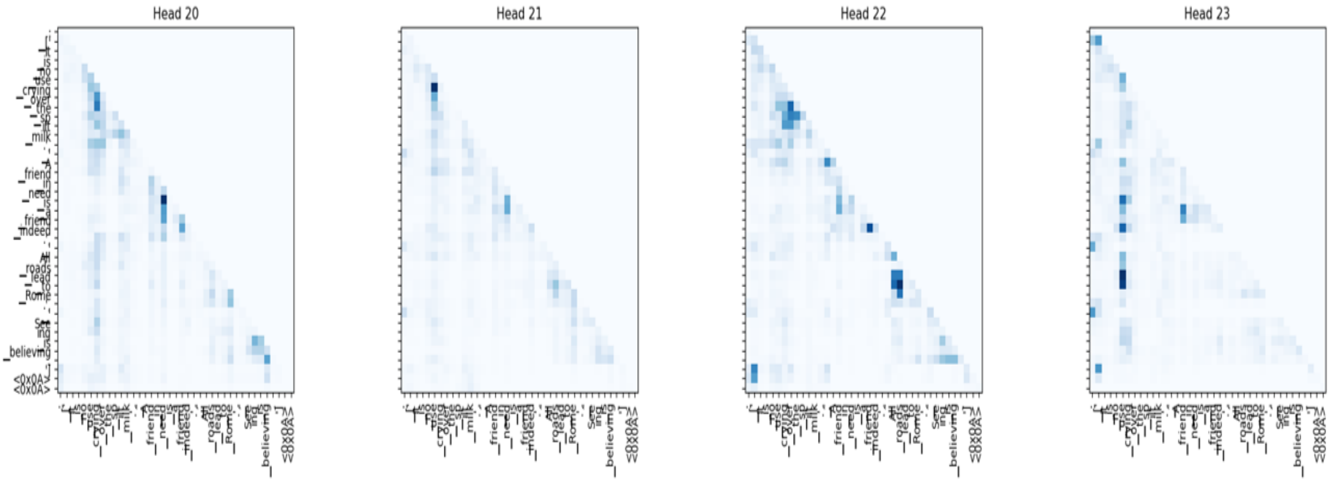

Figure 6.66: Layer 5 Head 20 to 23(After Pruning)

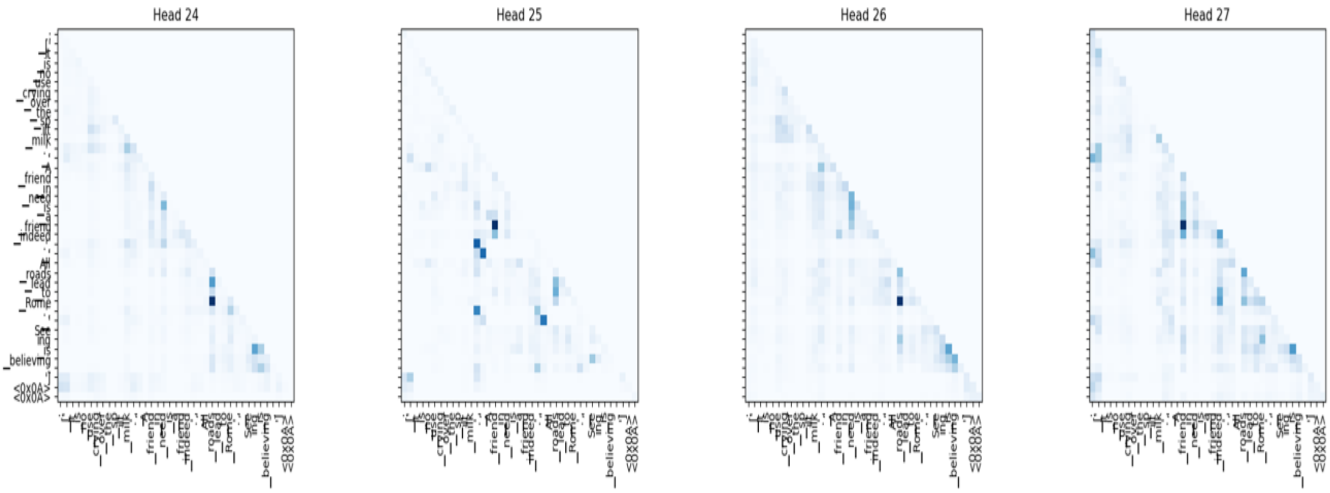

Figure 6.67: Layer 5 Head 24 to 27(After Pruning)

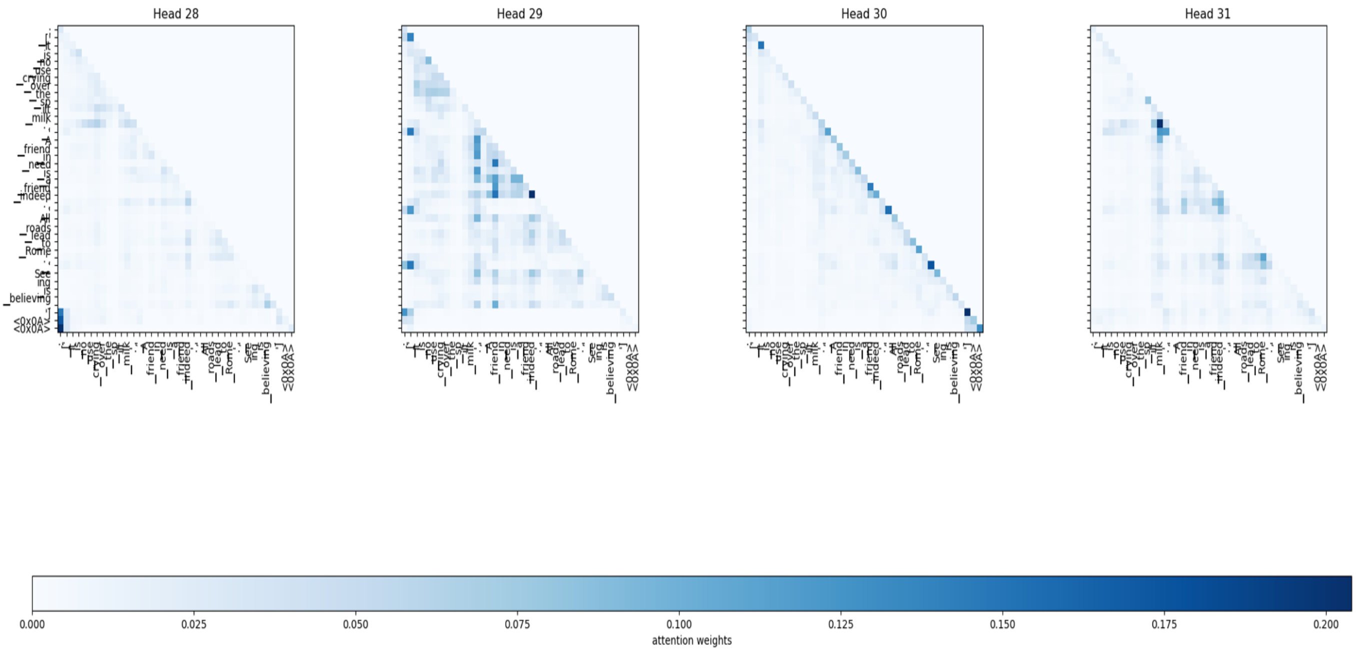

Figure 6.68: Layer 5 Head 28 to 31(After Pruning)

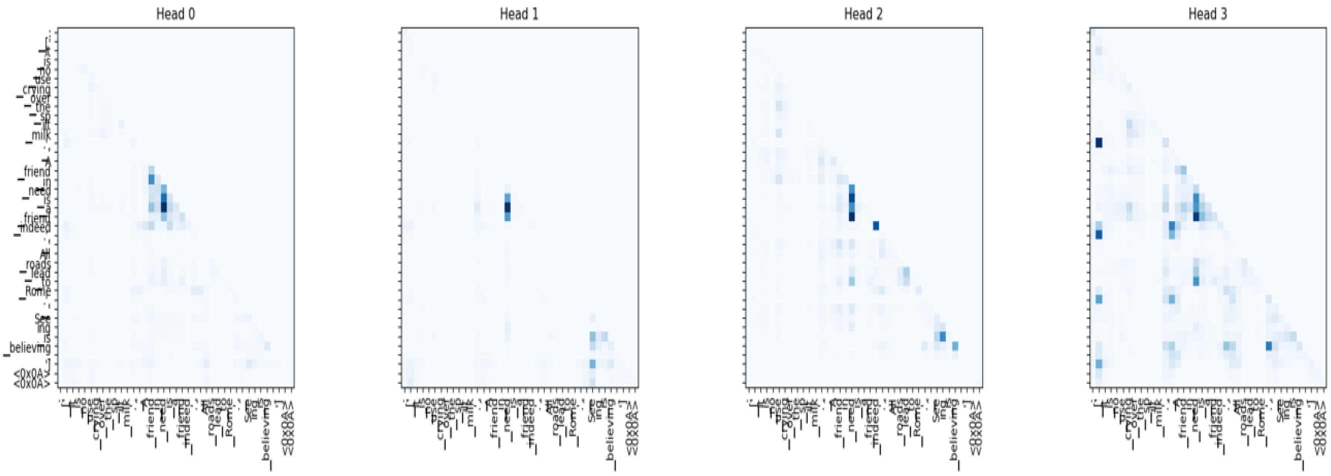

Figure 6.69: Layer 22 Head 0 to 3(After Pruning)

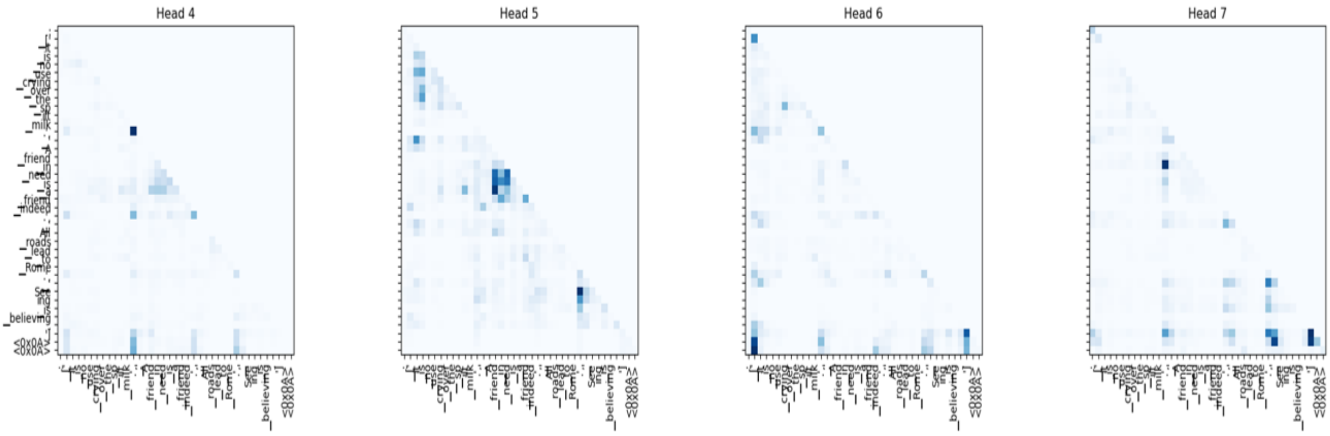

Figure 6.70: Layer 22 Head 4 to 7(After Pruning)

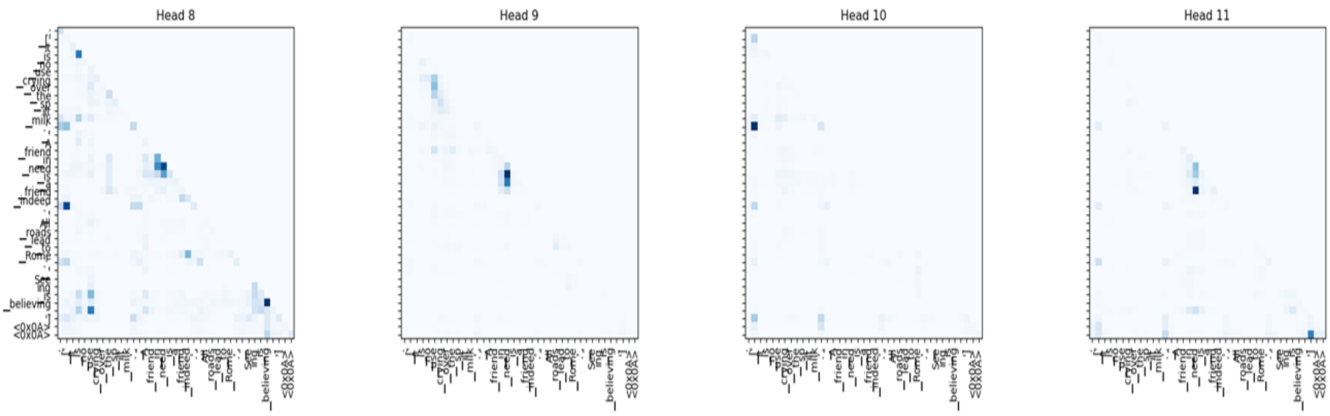

Figure 6.71: Layer 22 Head 8 to 11(After Pruning)

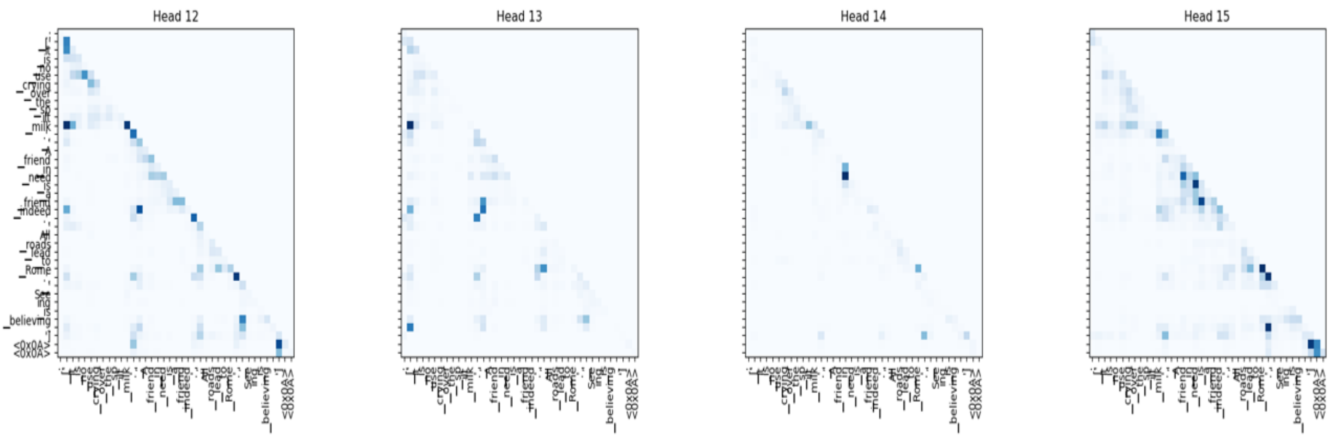

Figure 6.72: Layer 22 Head 12 to 15(After Pruning)

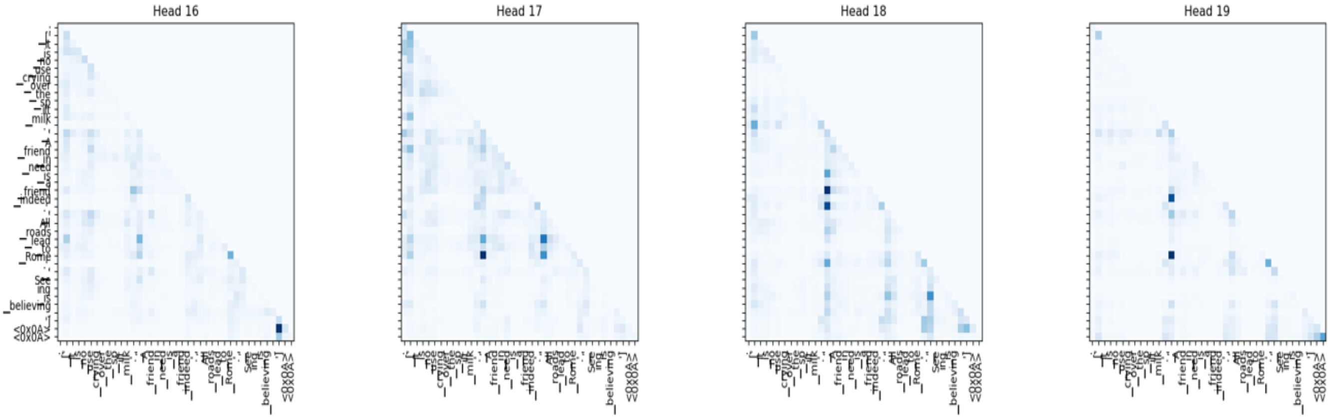

Figure 6.73: Layer 22 Head 16 to 19(After Pruning)

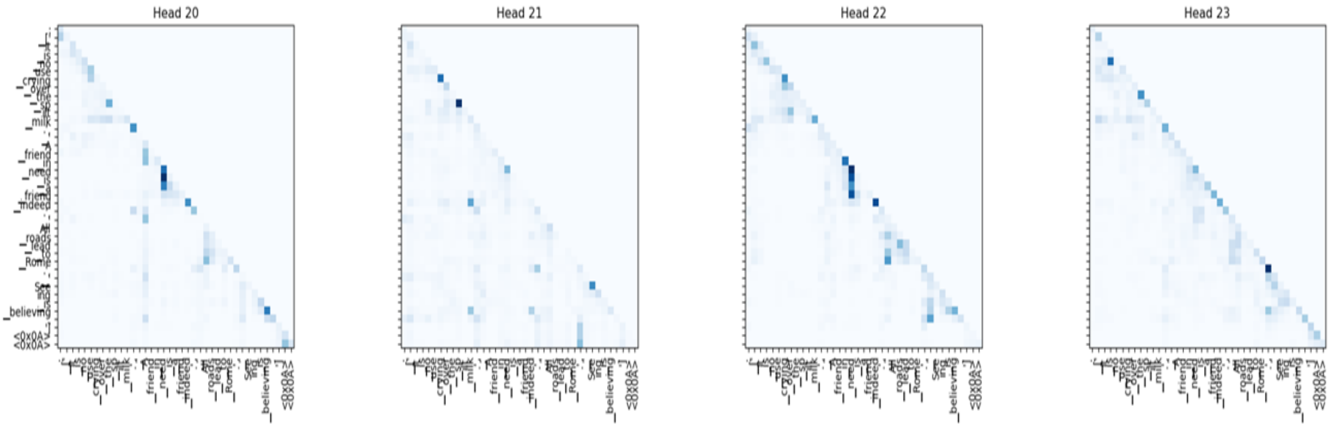

Figure 6.74: Layer 22 Head 20 to 23(After Pruning)

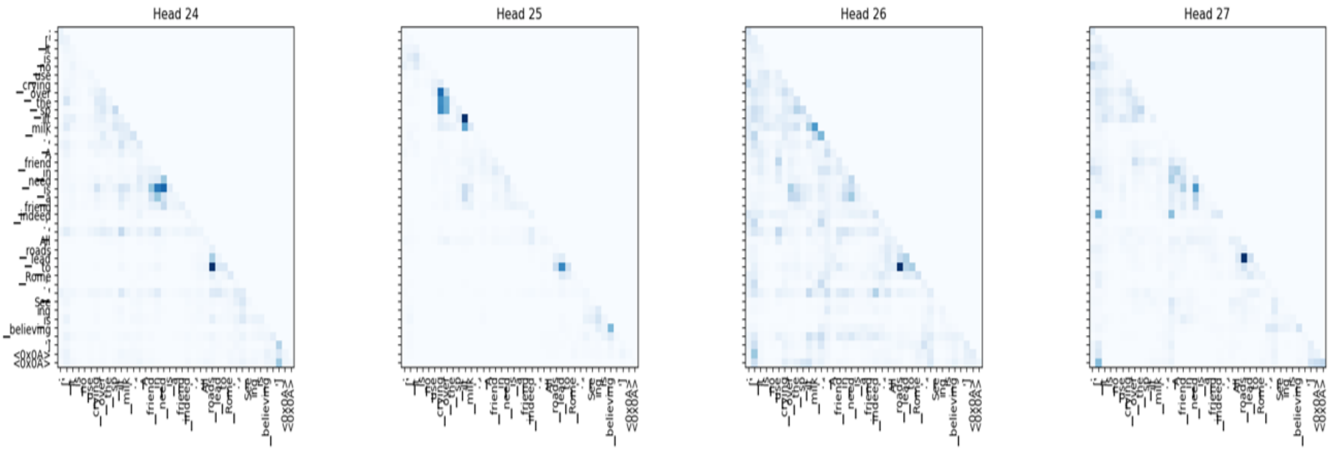

Figure 6.75: Layer 22 Head 24 to 27(After Pruning)

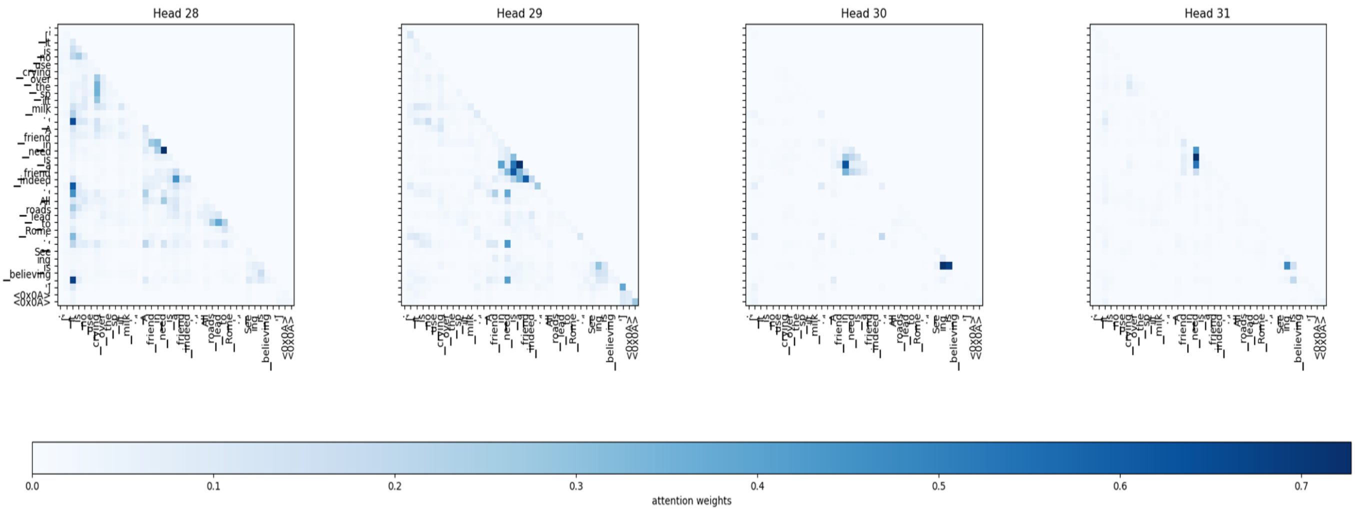

Figure 6.76: Layer 22 Head 28 to 31(After Pruning)

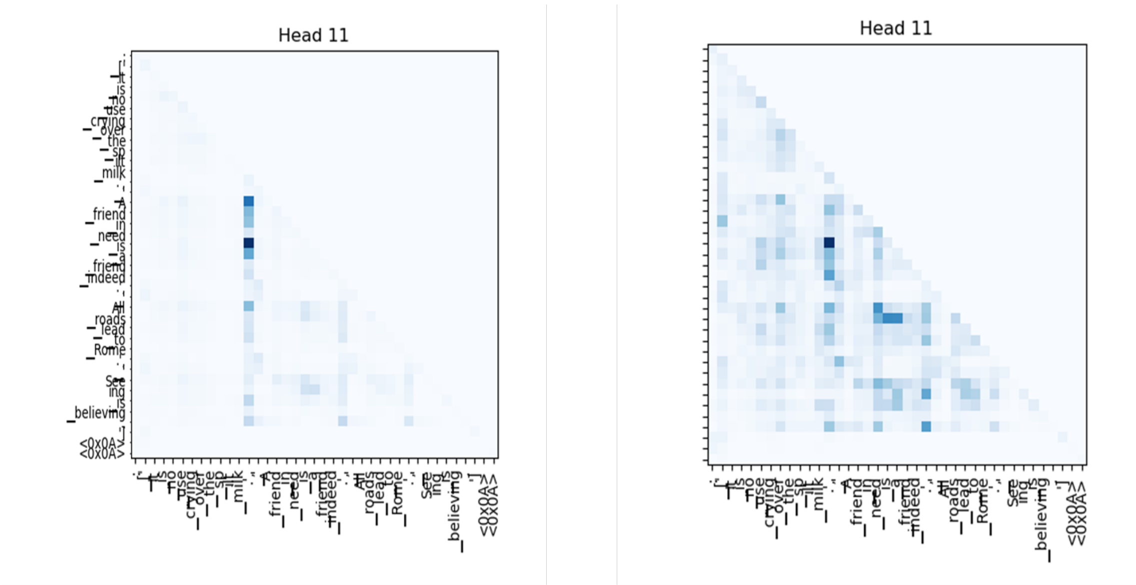

Figure 6.77: Layer 5 head 11 before(left) and after(right) pruning

So we can visualize that after pruning, some dead heads began to extract information from the context. Many attention heads got fixed after pruning and this is the reason we got the correct answer after pruning

After two simple tests, we decided to test after the verification results. So if any results show hallucination even after verification, we decided to find the reason and analyze both before and after pruning results and attention heads. So our third example is:

Prompt: Given an article, a question, and 4 choices, choose the best choice among the four. When choosing, make sure your choice is based on critically analyzing the article, question and the answer choices. Answer format: Just output the best choice. Don't write any explanation. Don't write anything extra. Just the answer choice.

Article: "I planted a seed. Finally grow fruits. Today is a great day. Pick off the star for you. Pick off the moon for you. Let it rise for you every day. Become candles burning myself. Just light you up, hey!... You are my little little apple. How much I love you, still no enough." This words are from the popular song You Are My Little Dear Apple. Bae Seul-Ki acted as the leading dancer in the MV of the song. She loves dancing. She became crazy about hip-hop when she was a school girl. Bai Seul-Ki was born on September 27, 1986. She is a South Korean singer and dancer. She is 168cm tall. She loves cooking. Her favourite food is spicy and salty. She like pink and red most. There are five members in her family—father, mother, two younger brothers and herself. She isn't married. After her father and mother broke up, she lived with her mother and new daddy. She enjoys being alone.

Original Question: We can infer from the passage that .

Options: ['this article is mainly about a song.', "Bai Seul-Ki can't cook", "Bai Seul-Ki doesn't have a good family life.", 'Bai Seul-Ki became interested in dancing after she left school.']
Baseline Answer: Option: ['this article is mainly about a song.', "Bai Seul-Ki can't cook", "Bai Seul-Ki doesn't have a good family life.", 'Bai Seul-Ki became interested in dancing after she left school.']
Verification Questions Answer Pairs: Q1: What is the main topic of the article?
A1: The main topic of the article is Bae Seul-Ki, a South Korean singer and dancer, who loves dancing and cooking, has a favorite food of spicy and salty, likes pink and red colors, and is single after her parents divorced.

Q2: Does Bai Seul-Ki have any siblings?
A2: Yes, Bai Seul-Ki has two younger brothers.

Q3: What is Bai Seul-Ki's favorite color?
A3: Bai Seul-Ki's favorite color is not mentioned in the given information.
First inference result was
Answer: Best choice: ['this article is mainly about a song.']
Alternative choice: None of the above
Explanation: The article is mainly about a song, and it mentions that Bae Seul-Ki appeared in the music video for that song. The article provides information about Bae Seul-Ki, but the focus is on the song. Therefore, the best choice is "this article is mainly about a song."
But the correct answer was "Bai Seul-Ki doesn't have a good family life."
We separately analyzed the attention heads of Questions, Options, and Baseline Answers to see if there were any defective heads. We found three layers common in three of them and those are 12, 27, and 29 Here are some samples of defective heads:

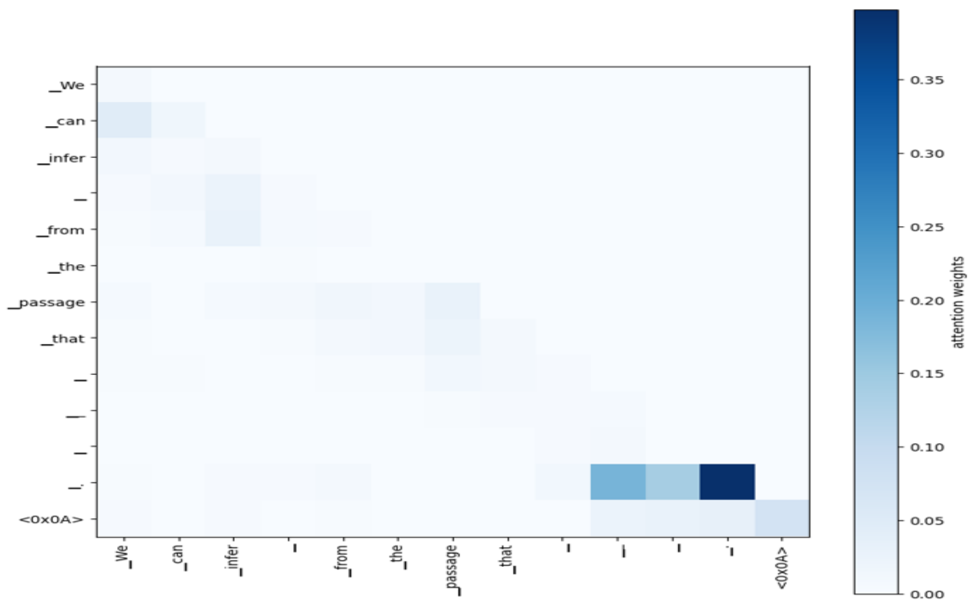

Figure 6.78: Layer 27 head 0

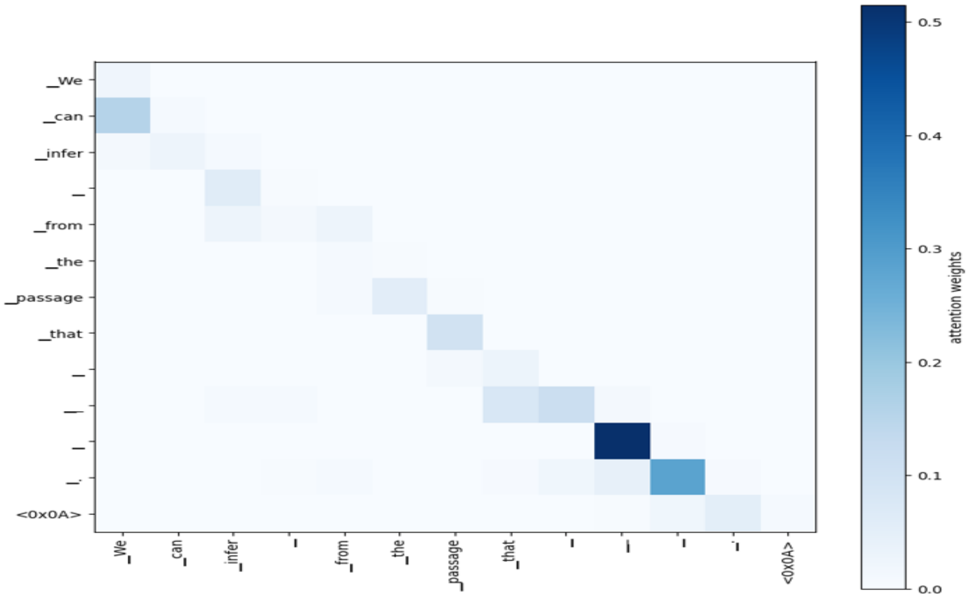

Figure 6.79: Layer 27 Head 1

We can see that the tokens were not focusing on any words before and they were just focusing on themselves which is not good

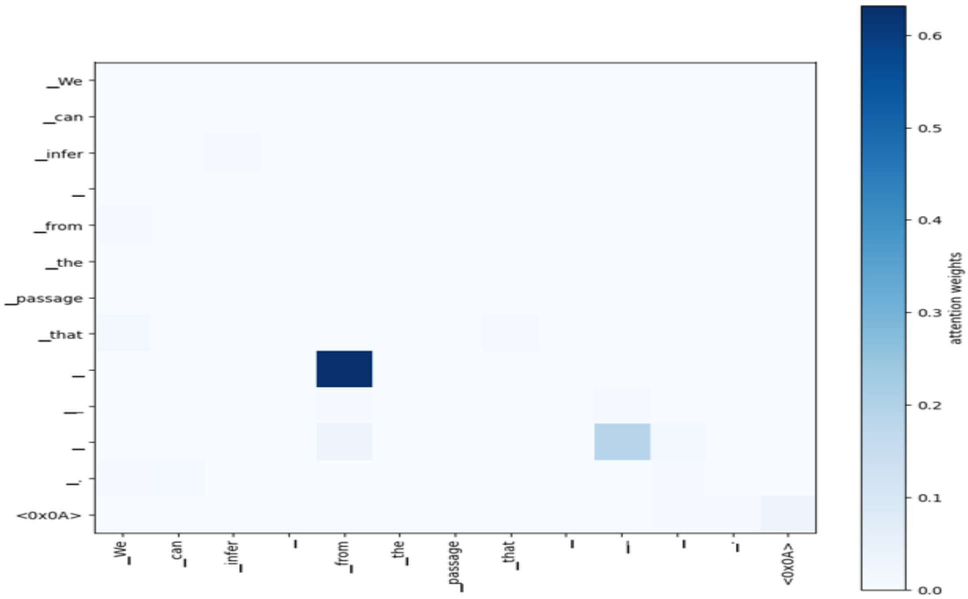

Figure 6.80: Layer 12 Head 4

This is an example of one of the worst attention heads.

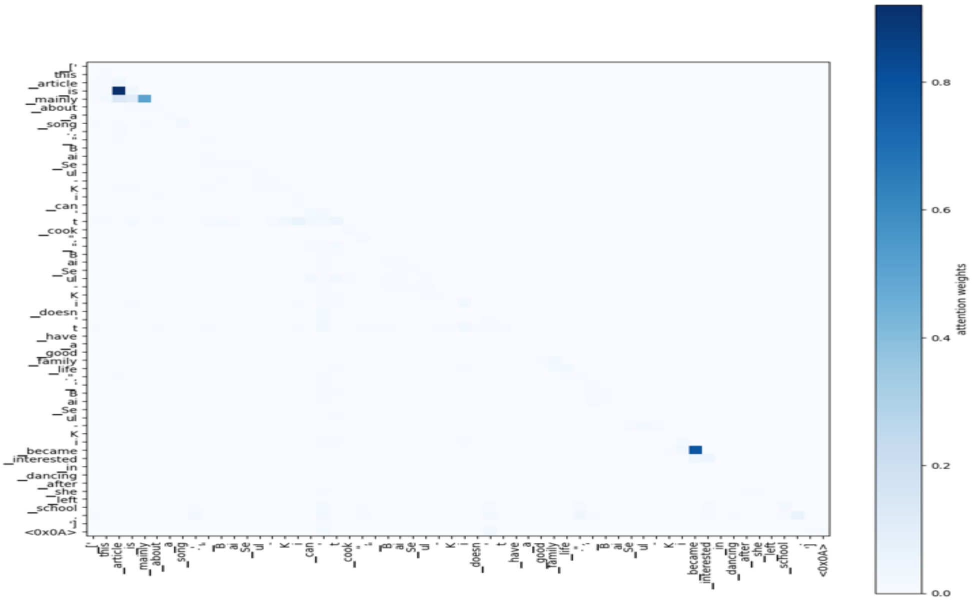

Figure 6.81: Layer 27 Head 19

It is the attention head of options and we can consider this also as one of the worst attention heads

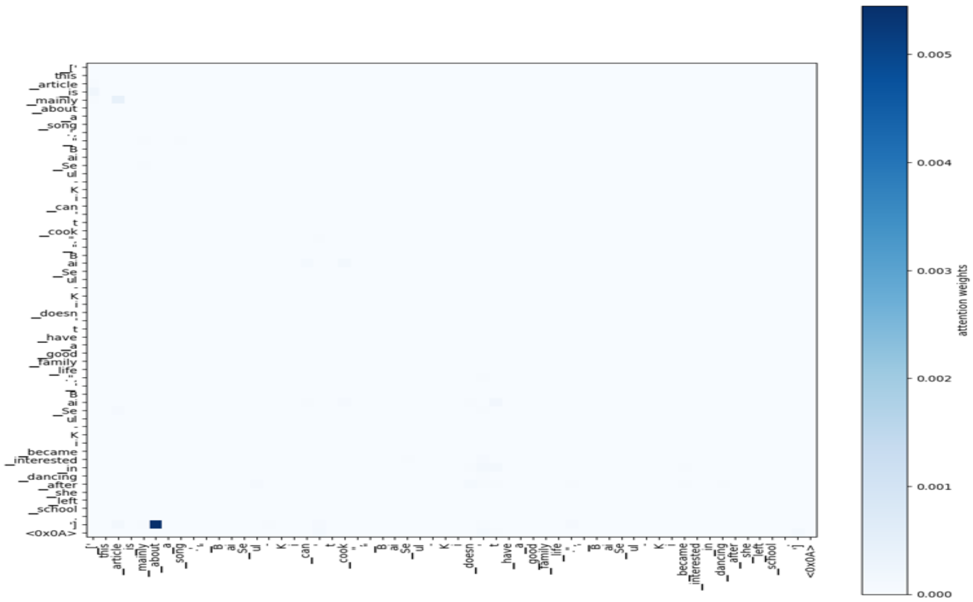

Figure 6.82: Layer 12 Head 5

Here is also one of the worst heads-on Options and there is almost 0 attention
But not all attention heads are bad. Here is an example of the best attention head

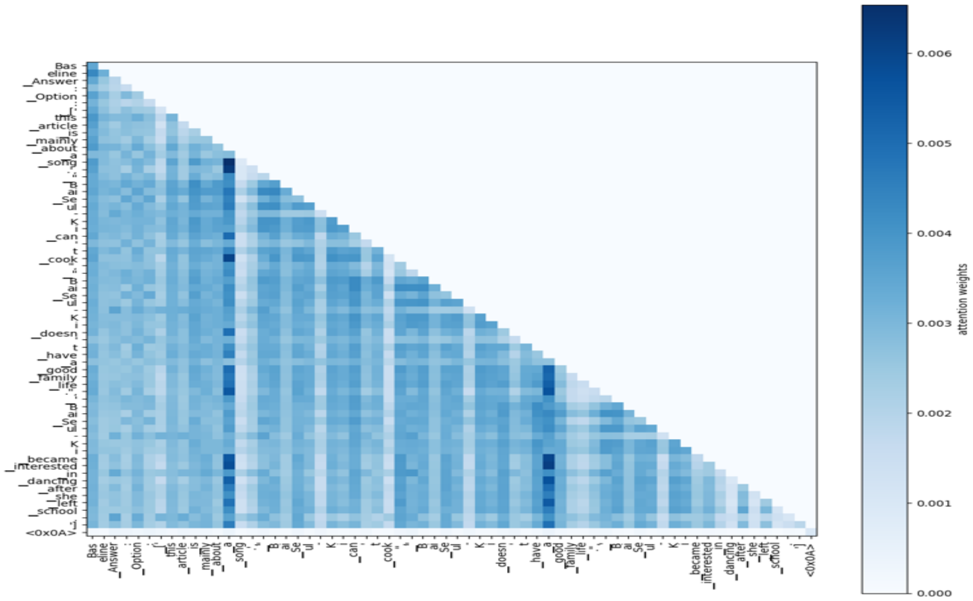

Figure 6.83: Layer 0 Head 19

This is an attention head of Baseline Responses. In this attention head, we can see that every token is focusing on themselves and before theirs. It is a good attention head which helps the model to understand the context

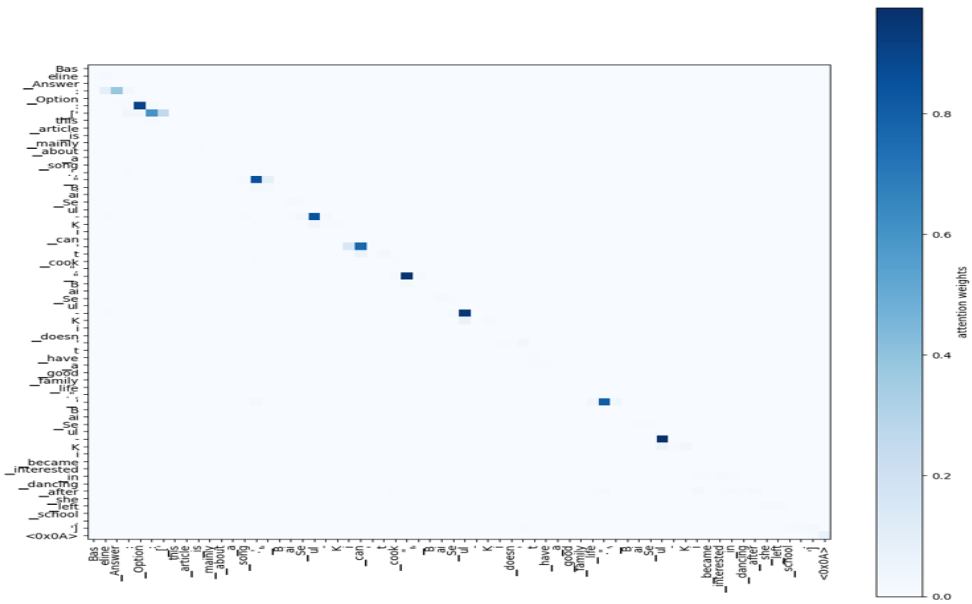

Figure 6.84: Layer 0 Head 19

But this is an attention to baseline response which is extremely bad. This head is just focusing on some of their own words and the rest is blank. This is the reason that most of the LLM models are still in hallucination even after verifications

So we have decided to prune layers 12, 27, and 29 for 5%. After pruning 5%, the inference result was

Answer: From the passage, we can infer that 'Bai Seul-Ki doesn't have a good family life.'

Therefore, the answer is: ['Bai Seul-Ki doesn't have a good family life.']

Here are all answer pairs for further reference:

[('What is the main topic of the article?', 'this article is mainly about a song.'),

('Does Bai Seul-Ki can't cook?', 'Bai Seul-Ki can't cook'),

('Does Bai Seul-Ki has good family life?', 'Bai Seul-Ki doesn't have a good family life.'),
('When did Bai Seul-Ki became interested in dancing after she left school?',
'Bai Seul-Ki became interested in dancing after she left school.')]

This is the correct answer matched with the targets. So layers 12, 27, and 29 were experiencing hallucinations in the model as the heads of these layers were unable to extract the real information. Here are those heads after pruning

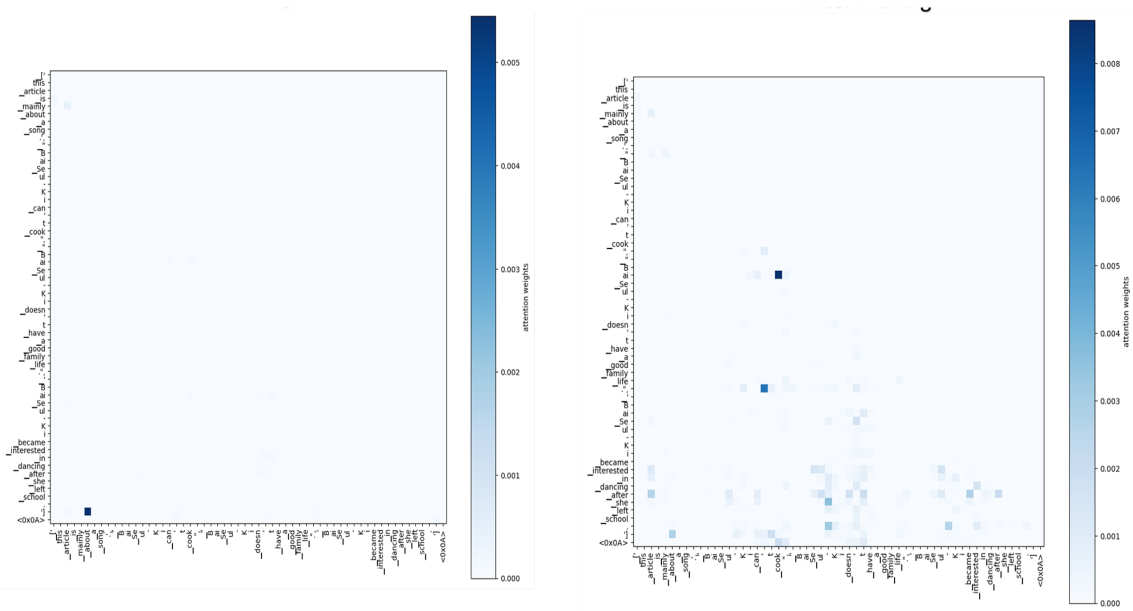

Figure 6.85: Layer 27 head 0 before(left) and after(right) pruning

Before pruning, tokes like passage, that was not impactful. But now, their score their score is higher than before. So it is helping

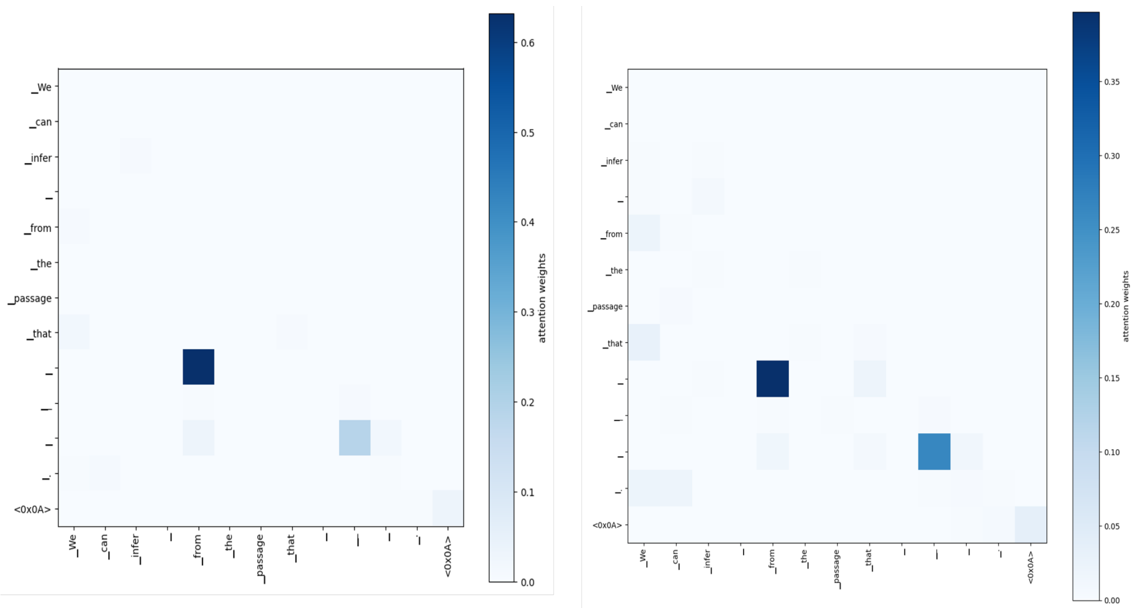

Figure 6.86: Layer 12 Head 4 before and after pruning

In this attention head, some of the tokens just came active. Otherwise, not that much changes

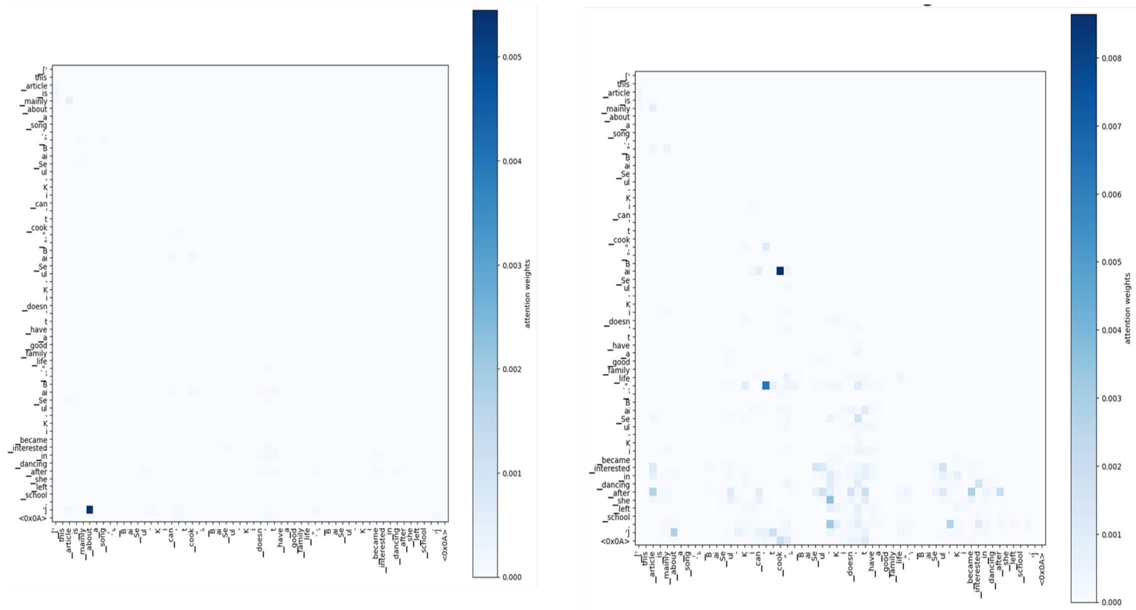

Figure 6.87: Layer 12 heads 5 before and after pruning

Before pruning, this was one of the worst heads. But now, it is much good.
So after pruning the weights of some layers, we can reduce hallucinations of an LLM model.
Here is another test after verification which is refers as test case 4

Prompt:
Given an article, a question, and 4 choices, choose the best choice among the four. When choosing, make sure your choice is based on critically analyzing the article, question and the answer choices. Answer format: Just output the best choice. Don't write any explanation. Don't write anything extra. Just the answer choice.

Article: Recently we had a discussion on the topic of My Chinese Dream. Inspired by President Xi's Chinese Dream, everyone talked happily about his understanding of the Chinese Dream. For me,I would like to be a teacher when I listen to my teacher carefully. I think I can be a teacher when I grow up. I can help many students learn things well. I can play with my students, too. So we are good friends.I would like to be a doctor when I see many doctors save their . Then I can help many people out of danger. I will be the happiest girl in the world. I would like to be a reporter when I watch TV every evening. We can get lots of important information from them. And I can learn a lot about China and the other countries around the world. They make the world smaller and also make us happy. I have lots of dreams. Hold fast to my dreams, no matter how big or small they are. The dreams may not be smooth and wide, even need some sacrifices, but hold on to the end, you will find there is no greater happiness than making your dream come true.

Original Question:
This article mainly tells us that .

Options:
['I would like to be a teacher', 'Dreams need sacrifices', 'MyChineseDream', 'Dreams are very important']

Baseline Answer: "Dreams need sacrifices"

Verification Questions Answer Pairs: Q1: What does the article primarily convey?
A1: The article primarily conveys that the author has various dreams for their future, including becoming a teacher, doctor, and reporter, and encourages others to hold onto their dreams despite potential challenges and sacrifices.

Q2: Is it possible for someone to achieve multiple dreams?
A2: Yes, it is possible for someone to achieve multiple dreams.

Q3: How do dreams contribute to one’s happiness?
A3: Dreams contribute to one’s happiness by providing motivation, purpose, and fulfillment in life. Achieving one’s dreams can bring a sense of accomplishment and satisfaction, leading to increased happiness and self-esteem.

After first inference, the answer came **Answer:** ”Dreams need sacrifices”

But the correct answer is MyChineseDream. So we decided to again analysis the attention heads again. After analyzing each layer for questions, options, and baseline, we have found defects in layers 7, 16, 18, 24, 25, 29, 30, 31.

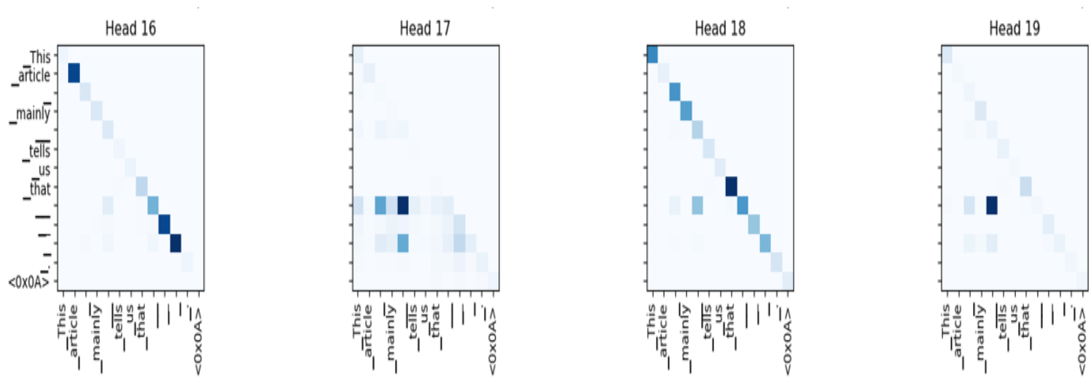

Figure 6.88: Layer 7

So we can see that heads 16, 18, 19 are in a bad phase which does not give good attention in their previous words which is one of the reasons for hallucination.

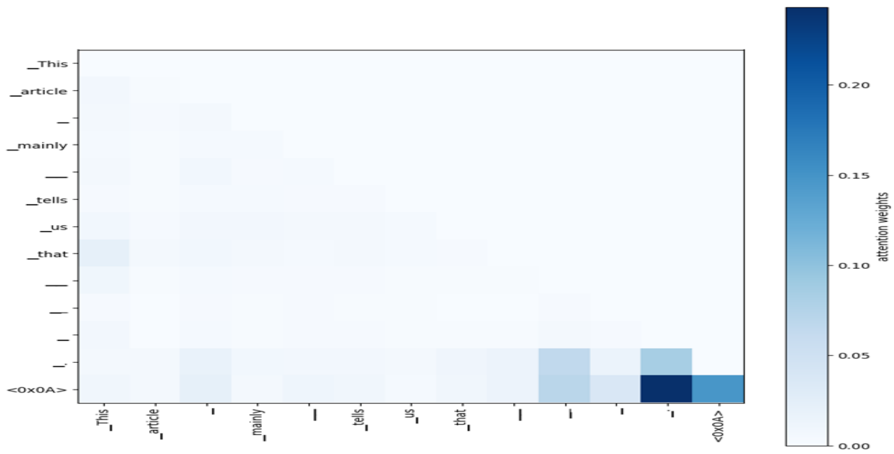

Figure 6.89: Layer 24 head 28

We can see that this attention head is also almost blank and some of the attention score is nearly 0.00 which is not good. And we do not need attention in “-” and “<0x0A>”. So this unnecessary attention and not giving attention on proper places are also causing hallucinations

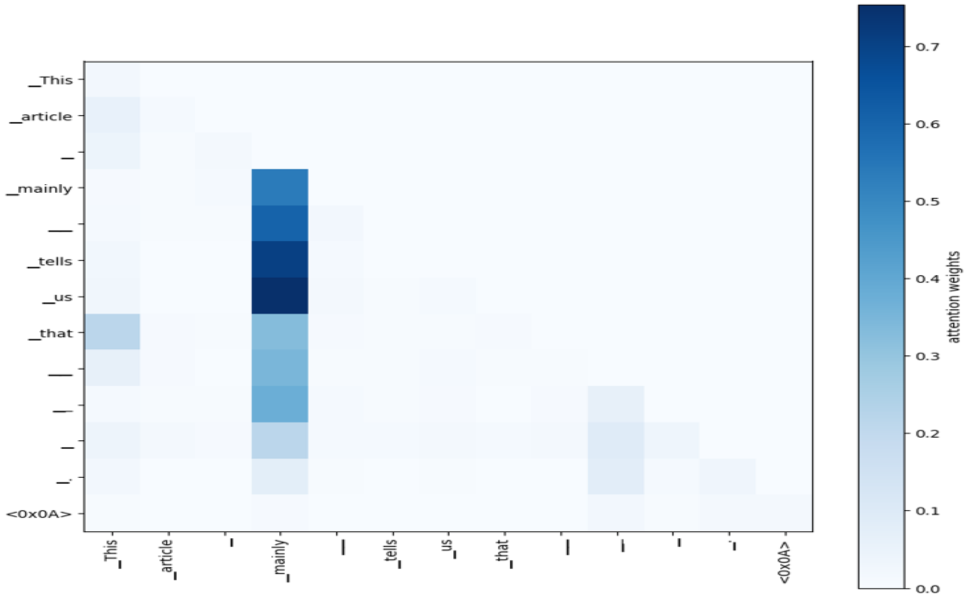

Figure 6.90: Layer 24 heads 7

Here we can see that words like mainly, tells, and us, are just focused on mainly for decoding which can cause hallucination as they cannot stuck to one words for decoding

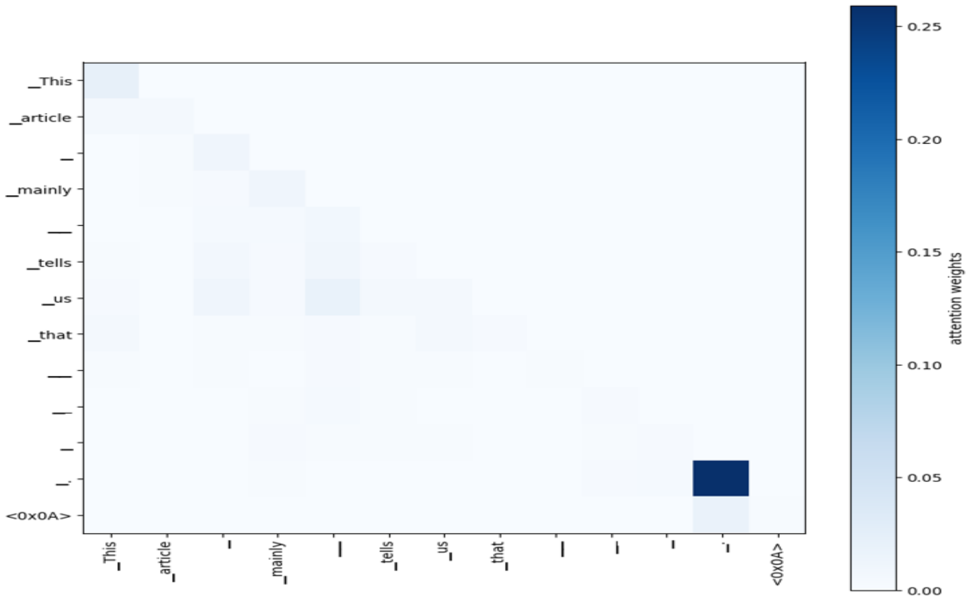

Figure 6.91: Layer 25 head 16

It is one of the worst attention heads which is almost blank for performing in decoding.

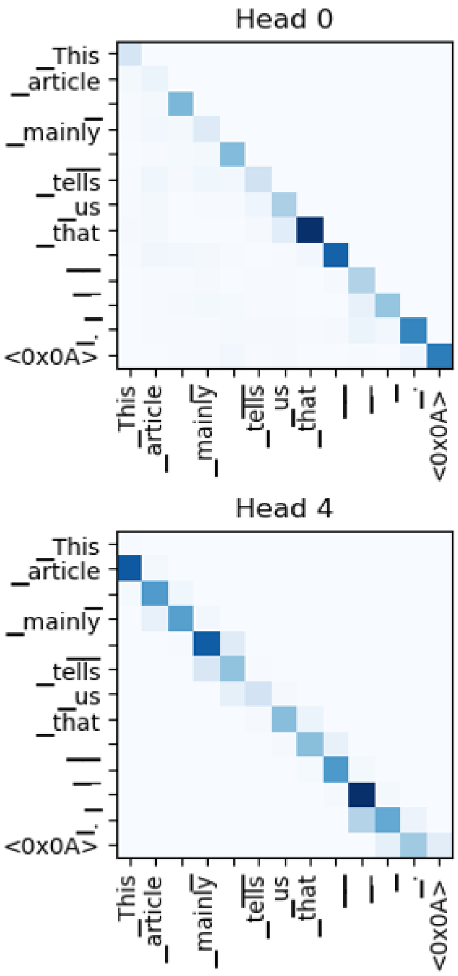

Figure 6.92: Layer 29

We can see that attention heads 4 and 0 were also unable to focus on their respective tokens

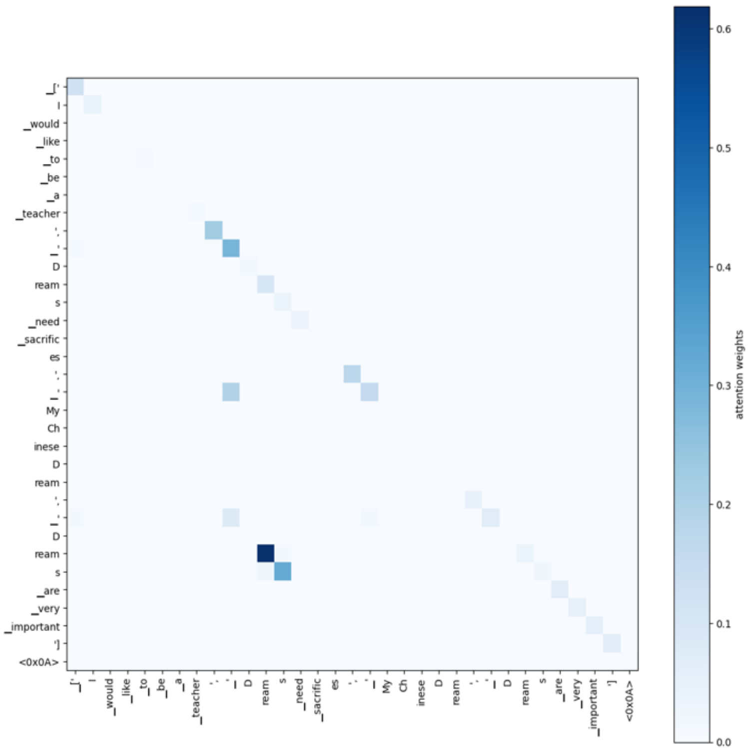

Figure 6.93: Layer 7 Head 18

It is also an example of the worst attention heads. Options should focus on their respective sentences of their options but here, they were not working

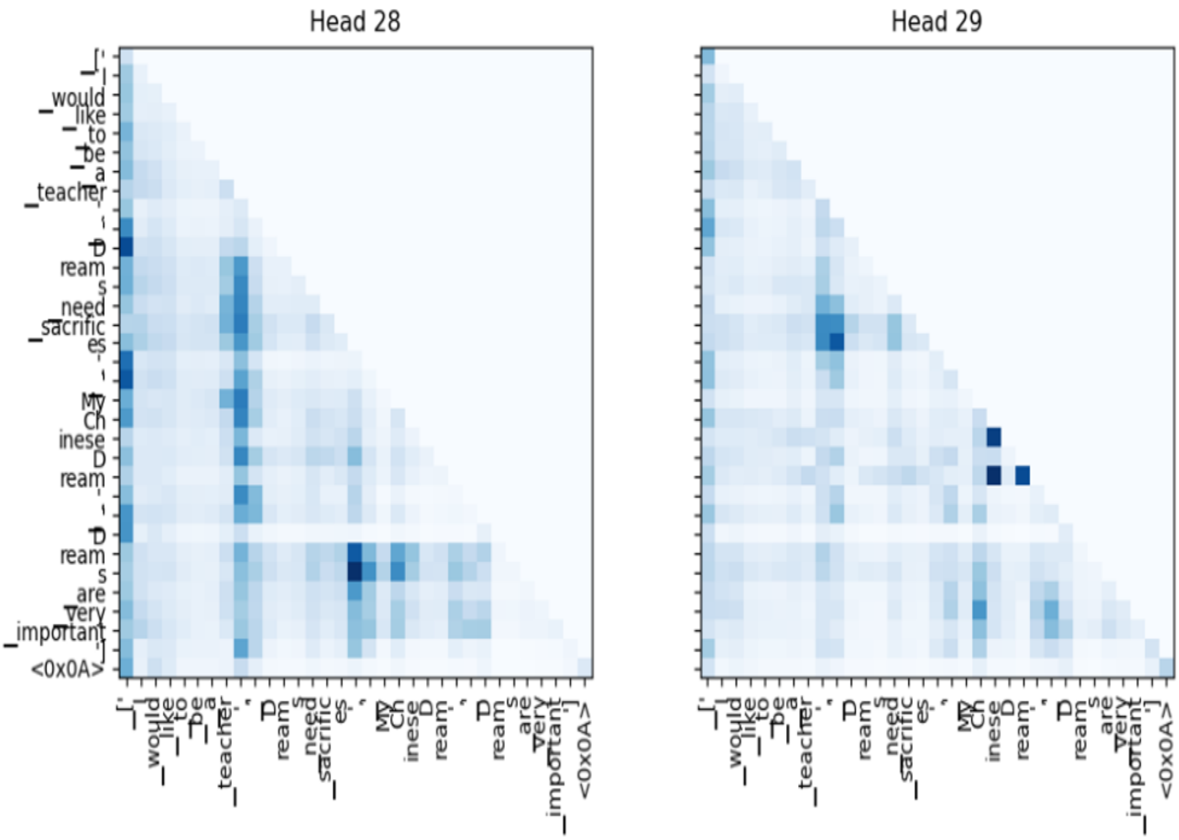

Figure 6.94: Layer 7

Here we can see that heads 16, 17, 18, 19 of layer 7 for baseline response are not quite good.

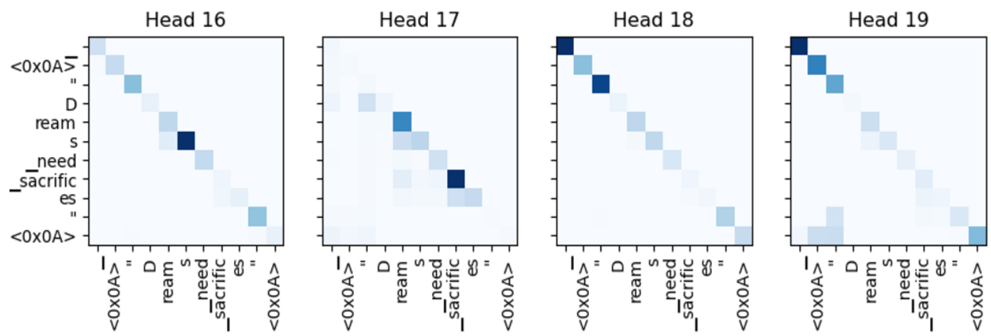

Figure 6.95: Layer 7

But on the other hand, we can see that head 28, 29, 30, 31 are quite good in the same layer So in the same layer, there were both good and bad attention heads which were trying to extract context and also create hallucinations. So if we prune some of the weights from those layers, then it can fix the issues

So we decided to prune 8% weights of layers 7, 16, 18, 24, 25, 29, 30, 31. After pruning, the inference result was:

After first inference, the answer came **Answer:** There isn't much in this article about Xi's Chinese Dream or China in general, so it wouldn't be accurate to choose "MyChineseDream" as the answer. The article does touch on the idea that dreams may require sacrifices, but it also emphasizes the importance of holding onto one's dreams in order to find happiness. Therefore, while the statement "Dreams need sacrifices" is true, it doesn't capture the full essence of the article. However, it is the most directly relevant answer to the original question ("What is the main idea of this article?") based on the provided article and choices. Therefore, the only option that accurately reflects the main idea of the article is "Dreams need sacrifices."

The actual answer was "MiChineseDream". But we can see that the model touched the option MyChineseDream but still could not give the correct answer. So even after pruning and almost touching the correct answer, the model began hallucinating

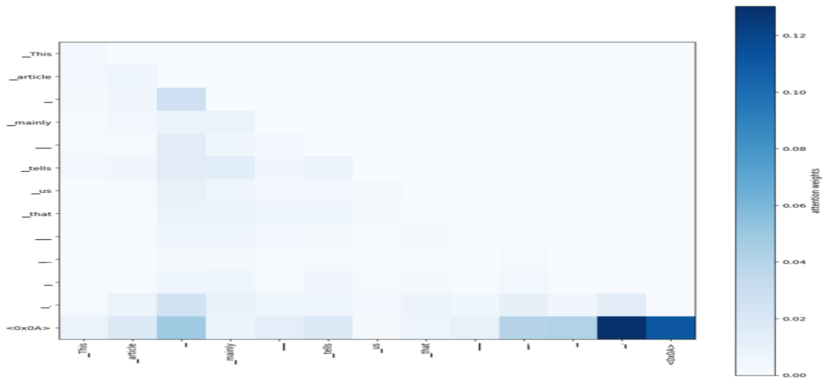

Figure 6.96: Layer 24 Head 28(After pruning)

This attention head became better than the previous one after pruning. So it is now extracting information better than the previous

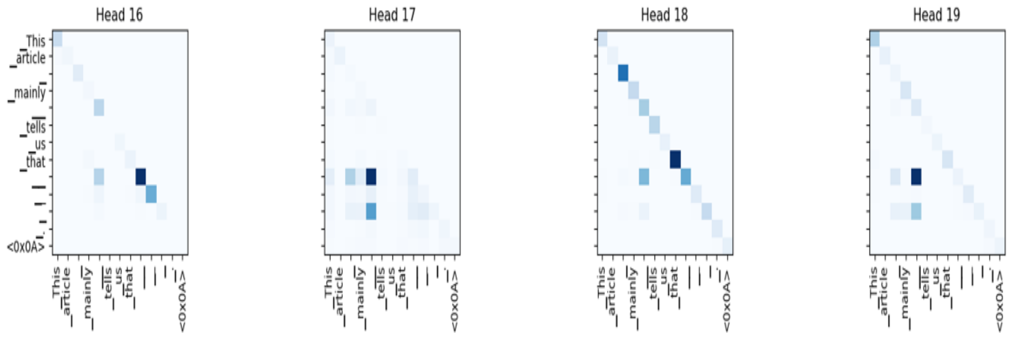

Figure 6.97: Layer 7(After Pruning)

Sometimes even after pruning, there are no such changes. And these 4 attention heads of layer 7 are a perfect example of that. Now there are such important heads in layer 7 so we cannot prune more than 8-10% there. So even after pruning, these attention heads still remain the same. Some scores had changed but not impactful

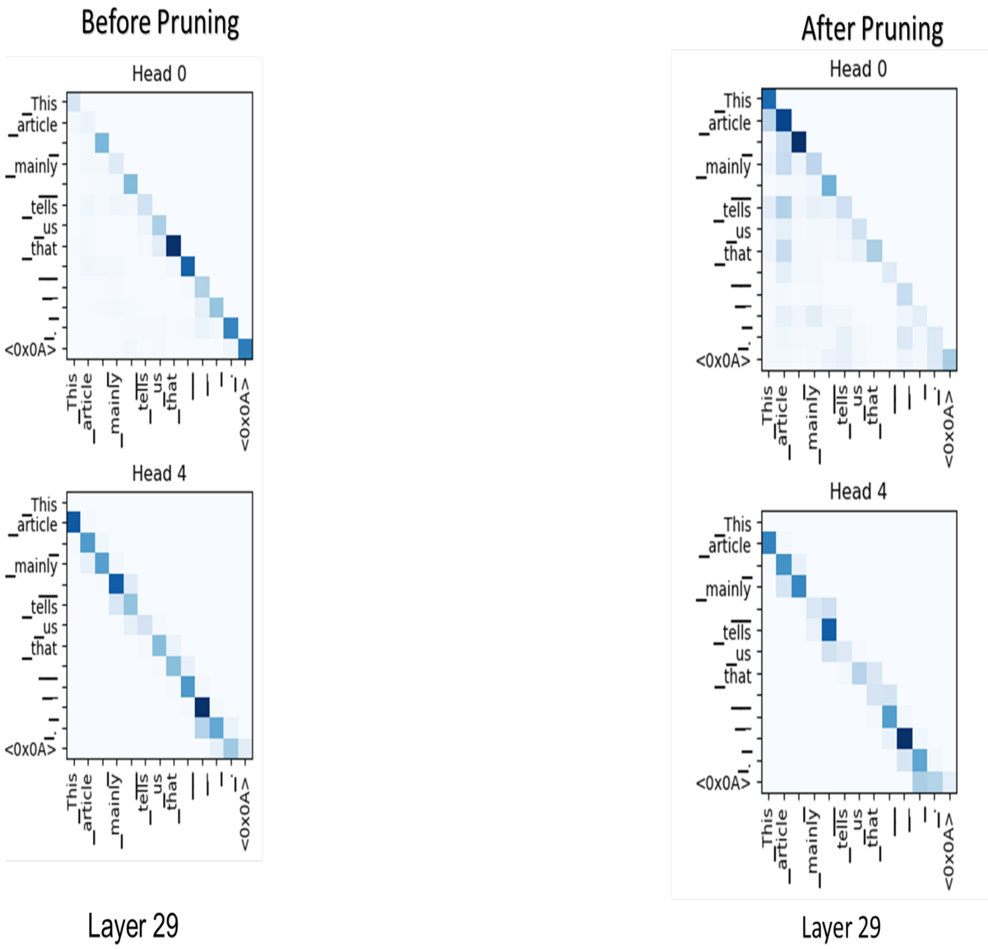

Figure 6.98: Layer 29(After Pruning)

We have visualized layer 29 again after pruning. Before pruning, the situation of both heads 0 and 4 was extremely bad. But after pruning, head 0 working fine, it is understanding the sentence. But head 4 remains the same. So one is changing and working better but on the other hand, another remains the same In this testing, we tried even to go beyond 10% but that ruined everything. So 8-10% was the best pruning measurements but still, in these measurements, we are unable to fix many of the attention heads. So we can conclude that even after pruning, the model can be hallucinated, giving wrong outputs. But we almost touched the output. So maybe pruning some useless weights, or training more on longer sequences can do better results for machine to understand human language

Chapter 7

Evaluation

Evaluating how accurately the models were performing was a cumbersome task. There were a couple of reasons for that. Since we're using an open-source small language model with much fewer parameters than the more advanced models, the output format of those models were pretty random, no matter how precisely we curated the prompts. As a result, it was difficult to directly compare the LLM generated answers to the answer format of the dataset. In addition to that, currently there is no approach that can calculate similarity between two pieces of texts. We still tried out a couple of different methods. Starting with a function called 'Cosine Similarity' which is used to calculate the distance vectors between two words. Even though the Cosine Similarity metric works pretty well when comparing single words. Comparing two pieces of text consisting of multiple words was not effective using this function. Hence we had to use a different approach. First, from the LLM response, we needed a way to figure out which answer is the answer choice the LLM is selecting. For this, we took our LLM response and calculated how similar it was compared to the answer choices. We used a classic Dynamic Programming Algorithm called 'Edit Distance' to essentially calculate how many operations it would take to convert the LLM response to one of the answer choices. The answer choice that had the least amount of distance was picked as the LLM response. This was done for both the baseline response and the final revised response. We then use Scikit-learns accuracy metrics to measure accuracy of those extracted responses. '

Chapter 8

Results

The dataset that we used for testing out our pipeline was the RACE dataset[45]. We evaluated the accuracy of the both the baseline responses and final responses using the approach described in the previous section. Since this is not a multiclass classification task and mostly a task on answer generation, we didn't use any metrices other than accuracy itself. evaluation metrics such as Precision, Recall and F1 score would have been irrelevant here. In terms of accuracy, the LLM was able to generate comparatively more correct answer choices after going through all the steps of our pipeline. the accuracy score was 61 percent for the baseline responses and 80 percent for the final responses. The LLM was able to correct its mistakes during the pipeline chain and made improvements in generating the correct result.

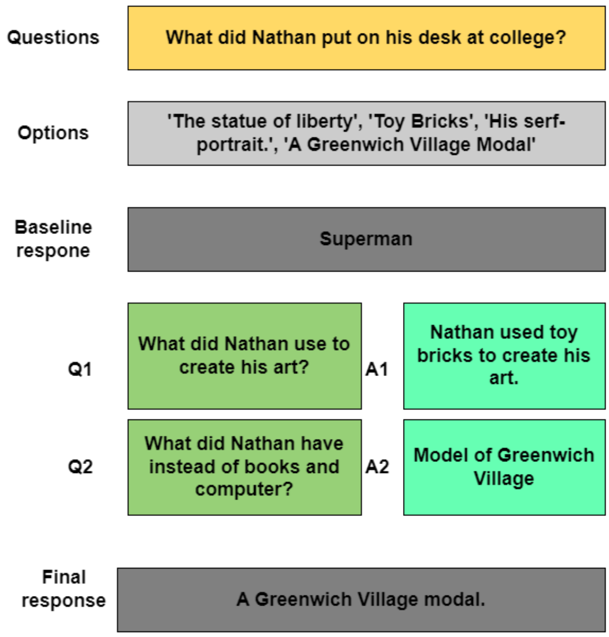

Figure 8.1: Correction during pipeline chain

After testing on middle level datasets, we have analyzed on 56 high level critical test sets of RACE Datasets[45]. We have interpreted and analyzed 4 test examples where 3 test examples work. In base example without pruning, we got 60% in baseline response and 40% in Final verification prompt examples. Then after first example analysis, we pruned layer 2 10%, and then after the inference on test sets, we found 60% in baseline and 57% in Final Response. In test example 2, we pruned layer 5, 22 10%, then got 50% accuracy in both baseline and final. In test example 3, after pruning 12, 27, 29 5%, we got 64% in baseline responses and 59% in final verification responses. In test example 4, after pruning 7, 16, 18, 24, 25, 29, 30, 31 for 8%, we got 42% in baseline and 53% in final verifications

	Layers	**Pruning**	**Baseline**	**Final**
Original	-	-	60%	40%
Test Example 1	2	10%	50%	57%
Test Example 2	5, 22	10%	50%	50%
Test Example 3	12, 27, 29	5%	64%	59%
Test Example 4	7, 16, 18, 24, 25, 29, 30, 31	8%	42%	53%

Table 8.1: Layers, Pruning, Baseline, and Final Results

Chapter 9

Research Limitations and Future Work

9.1 Limitations

As per limitations. Since our research required using good generative models with higher number of parameters, we needed a computer that has a GPU with 48GB video RAM or higher. But even the most highly configured computer in our lab had 24 GB of VRAM. Hence we had to use a smaller model Mistral 7b. Even then, we had to perform 4 bit quantization in order to load the model in our VRAM. In that process, we lost a significant amount of precision. In addition to that, we we able to access the High-configuration computer for maximum of two weeks in the semester. Hence we were limited about how many things we could try out.
First of all, there still doesn't exist a method to reliably compare an LLM generated response to and actual correct response. Even though for LLMs there exists certain evaluation metrics such as BLUE score, Perplexity Score etc. None of them fitted well for the task in our research. BLUE score is used in evaluating machine translation tasks which is fairly simple to evaluate. On the other hand, perplexity score can evaluate how well a piece of LLM generated response is based on how well each token in the response fit in the probability distributions of next word prediction task.
However, to run our pipeline on long form generation tasks as opposed to answering reading comprehension based multiple choice questions, we needed a strong evaluation metrics for that. Currently, there are two sort of reliable ways to do that. First one is 'Human Evaluation'. In this way, a human has to manually score both the baseline response and the revised response against the correct answer to evaluate the goodness of the generations. But this way is extremely time consuming and also the reliability of this method is questionable. The other approach is to evaluate those response using another LLM. But using another language model in this case was not feasible for our research. Because using the API for latest chatbots such as ChatGPT-4 requires payment based on daily data usage. Hence using an external LLM would have been extremely costly for us. Because of those hardwire limitations and unavailability of a proper benchmark to judge passage like LLM generations, we had solely focus on simpler task as Reading Comprehension based MCQ question answering.

9.2 Future Work

In the last month, a couple of papers came out. One of them proposed a method called 'PHD'[ref]. Which is a new benchmark that can be used for passage-level hallucination detection. The dataset for this benchmark was generated using ChatGPT was annotated by humans. The authors showed that their benchmark was able to detect hallucination in passages without any external resources. In future, we want to implement this benchmark for longform generation tasks. Additionally, we would try to run our pipeline on a better model, tentatively Mistral 8X7B which is a 45 billion parameter language model as opposed to Mistral7B which has only 45 parameters. Mistral 8X7B has a much longer context window than the other one and is much better at comprehension and

reasoning tasks. Running our pipeline on that model will help us yield a much better sense of how well our pipeline chain works in terms of mitigating hallucination.
Furthermore, we want to try model editing for Changning some specific behaviors of LLMs. That way we will be able to tweak some parts of the model for our specific generation tasks without negatively impacting the performance.
Additionally, for interpretation, we would want to extract and critically analyze the decoder layer of Mistral 8X7B which would give us a much clearer sense of what the model is doing in the intermediate steps of mitigating hallucination.

Chapter 10

Conclusion

Interpretability of Large Language models is always a difficult task and when it comes to the matter of hallucination, the trouble increases. When a model gets less sequence input, it works better but when it gets longer input, then the model does not perform well. We have seen LLM models like chatGPT that often model confused to understand the task even after verification
We have implemented a multi-stage verification method. But when we see that even verification did not work well, then we begin to interpret the self-attention layer and we visualize the matter. Some attention heads in the self-attention layers are often unable to extract the context information from inputs and this happens in the longer input sequences. After a critical analysis, we decided to prune some weights in the layer that had more dead heads. After pruning, ¾ tests work. So pruning worked in many cases where we were able to reduce the hallucinations and interpret the model's attention. But in the 4th case, many dead heads were not fixed after pruning
So we can conclude that the LLM model based on Transformers attention still has some problems in extracting information/context. The matter can be solved if it gets trained on longer data sequences or has a dropout in each query, key, and value vector. We hope one day, LLM will work finely.

Bibliography

[1] T. B. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," 2020. arXiv: 2005.14165 `[cs.CL]`.

[2] S. Dhuliawala *et al.*, "Chain-of-verification reduces hallucination in large language models," 2023. eprint: 2309.11495. [Online]. Available: https://arxiv.org/abs/2309.11495.

[3] A. Q. Jiang, A. Sablayrolles, A. Mensch, *et al.*, *Mistral 7b*, 2023. arXiv: 2310.06825 `[cs.CL]`.

[4] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://arxiv.org/abs/1706.03762.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. [Online]. Available: https://aclanthology.org/N19-1423.

[6] X. Wang, Y. Yan, L. Huang, X. Zheng, and X. Huang, "Hallucination detection for generative large language models by bayesian sequential estimation," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023, p. 1537. [Online]. Available: https://aclanthology.org/2023.emnlp-main.949.pdf.

[7] L. Huang, W. Yu, W. Ma, *et al.*, *A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions*, 2023. arXiv: 2311.05232 `[cs.CL]`.

[8] A. T. Kalai and S. S. Vempala, "Calibrated language models must hallucinate," 2023. eprint: 2311.14648. [Online]. Available: https://arxiv.org/abs/2311.14648.

[9] N. M. Guerreiro *et al.*, "Hallucinations in large multilingual translation models," vol. 11, 2023, pp. 1500–1517. DOI: 10.1162/tacl_a_00615.

[10] J. Wei *et al.*, "Chain-of-thought prompting elicits reasoning in large language models chain-of-thought prompting," 2023. [Online]. Available: https://arxiv.org/pdf/2201.11903.pdf.

[11] M. I. Martino and C. Truong, “Knowledge graph augmentation with neural embeddings,” in *Proceedings of the European Semantic Web Conference (ESWC)*, Accessed: January 2024, 2023. [Online]. Available: https://2023.eswc-conferences.org/wp-content/uploads/2023/05/paper_Martino_2023_Knowledge.pdf.

[12] Y. Wu *et al.*, “Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models,” 2024. [Online]. Available: https://arxiv.org/pdf/2401.00396.pdf.

[13] Y. Ding, Y. Liu, H. Luan, and M. Sun, “Visualizing and understanding neural machine translation,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1150–1159. DOI: 10.18653/v1/P17-1106. [Online]. Available: https://aclanthology.org/P17-1106.

[14] O. r. Bojar, R. Chatterjee, C. Federmann, *et al.*, “Findings of the 2016 conference on machine translation,” in *Proceedings of the First Conference on Machine Translation*, Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 131–198. [Online]. Available: http://www.aclweb.org/anthology/W/W16/W16-2301.

[15] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, “Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 5797–5808. DOI: 10.18653/v1/P19-1580. [Online]. Available: https://aclanthology.org/P19-1580.

[16] J. Vig and Y. Belinkov, “Analyzing the structure of attention in a transformer language model,” in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 63–76. DOI: 10.18653/v1/W19-4808. [Online]. Available: https://aclanthology.org/W19-4808.

[17] J. Vig, “A multiscale visualization of attention in the transformer model,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 37–42. DOI: 10.18653/v1/P19-3007. [Online]. Available: https://aclanthology.org/P19-3007.

[18] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, “What does BERT look at? an analysis of BERT’s attention,” in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 276–286. DOI: 10.18653/v1/W19-4828. [Online]. Available: https://aclanthology.org/W19-4828.

[19] S. Jain and B. C. Wallace, “Attention is not explanation,” vol. abs/1902.10186, 2019. arXiv: 1902.10186. [Online]. Available: http://arxiv.org/abs/1902.10186.

[20] S. Wiegreffe and Y. Pinter, “Attention is not not explanation,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 11–20. DOI: 10.18653/v1/D19-1002. [Online]. Available: https://aclanthology.org/D19-1002.

[21] J. Bastings and K. Filippova, “The elephant in the interpretability room: Why use attention as explanation when we have saliency methods?” In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, Online: Association for Computational Linguistics, Nov. 2020, pp. 149–155. DOI: 10.18653/v1/2020.blackboxnlp-1.14. [Online]. Available: https://aclanthology.org/2020.blackboxnlp-1.14.

[22] Y.-H. H. Tsai, M. Ma, M. Yang, R. Salakhutdinov, and L.-P. Morency, “Multimodal routing: Improving local and global interpretability of multimodal language analysis,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, Nov. 2020, pp. 1823–1833. DOI: 10.18653/v1/2020.emnlp-main.143. [Online]. Available: https://aclanthology.org/2020.emnlp-main.143.

[23] Y. Hao, L. Dong, F. Wei, and K. Xu, “Self-attention attribution: Interpreting information interactions inside transformer,” 14, vol. 35, May 2021, pp. 12 963–12 971. DOI: 10.1609/aaai.v35i14.17533. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17533.

[24] E. Voita, R. Sennrich, and I. Titov, “Analyzing the source and target contributions to predictions in neural machine translation,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online: Association for Computational Linguistics, Aug. 2021, pp. 1126–1140. DOI: 10.18653/v1/2021.acl-long.91. [Online]. Available: https://aclanthology.org/2021.acl-long.91.

[25] W. Ma, K. Zhang, R. Lou, L. Wang, and S. Vosoughi, “Contributions of transformer attention heads in multi- and cross-lingual tasks,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online: Association for Computational Linguistics, Aug. 2021, pp. 1956–1966. DOI: 10.18653/v1/2021.acl-long.152. [Online]. Available: https://aclanthology.org/2021.acl-long.152.

[26] C. Meister, S. Lazov, I. Augenstein, and R. Cotterell, “Is sparse attention more interpretable?” In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Online: Association for Computational Linguistics, Aug. 2021, pp. 122–129. DOI: 10.18653/v1/2021.acl-short.17. [Online]. Available: https://aclanthology.org/2021.acl-short.17.

[27] Y. Bian, J. Huang, X. Cai, J. Yuan, and K. Church, “On attention redundancy: A comprehensive study,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online: Association for Computational Linguistics, Jun. 2021, pp. 930–945. DOI: 10.18653/v1/2021.naacl-main.72. [Online]. Available: https://aclanthology.org/2021.naacl-main.72.

[28] M. Rigotti, C. Miksovic, I. Giurgiu, T. Gschwind, and P. Scotton, “Attention-based interpretability with concept transformers,” Oct. 2021. [Online]. Available: https://openreview.net/forum?id=kAa9eDS0RdO.

[29] B. Alastruey, J. Ferrando, G. I. Gállego, and M. R. Costa-jussà, “On the locality of attention in direct speech translation,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 402–412. DOI: 10.18653/v1/2022.acl-srw.32. [Online]. Available: https://aclanthology.org/2022.acl-srw.32.

[30] C. M. Schuster and S. Hegelich, “From BERT‘s Point of View: Revealing the Prevailing Contextual Differences,” in *Findings of the Association for Computational Linguistics: ACL 2022*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1120–1138. DOI: 10.18653/v1/2022.findings-acl.89. [Online]. Available: https://aclanthology.org/2022.findings-acl.89.

[31] D. Liu, D. Greene, and R. Dong, “A novel perspective to look at attention: Bi-level attention-based explainable topic modeling for news classification,” in *Findings of the Association for Computational Linguistics: ACL 2022*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2280–2290. DOI: 10.18653/v1/2022.findings-acl.178. [Online]. Available: https://aclanthology.org/2022.findings-acl.178.

[32] F. Yin, Z. Shi, C.-J. Hsieh, and K.-W. Chang, “On the sensitivity and stability of model interpretations in NLP,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2631–2647. DOI: 10.18653/v1/2022.acl-long.188. [Online]. Available: https://aclanthology.org/2022.acl-long.188.

[33] M. Geva, A. Caciularu, K. Wang, and Y. Goldberg, “Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space,” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 30–45. [Online]. Available: https://aclanthology.org/2022.emnlp-main.3.

[34] A. Modarressi, M. Fayyaz, Y. Yaghoobzadeh, and M. T. Pilehvar, “GlobEnc: Quantifying global token attribution by incorporating the whole encoder layer in transformers,” in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 258–271. DOI: 10.18653/v1/2022.naacl-main.19. [Online]. Available: https://aclanthology.org/2022.naacl-main.19.

[35] H. Sajjad, N. Durrani, F. Dalvi, F. Alam, A. Khan, and J. Xu, "Analyzing encoded concepts in transformer language models," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 3082–3101. DOI: 10.18653/v1/2022.naacl-main.225. [Online]. Available: https://aclanthology.org/2022.naacl-main.225.

[36] J. Ferrando, G. I. Gállego, B. Alastruey, C. Escolano, and M. R. Costa-jussà, "Towards opening the black box of neural machine translation: Source and target interpretations of the transformer," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 8756–8769. [Online]. Available: https://aclanthology.org/2022.emnlp-main.599.

[37] D. Dai, L. Dong, Y. Hao, Z. Sui, B. Chang, and F. Wei, "Knowledge neurons in pretrained transformers," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8493–8502. DOI: 10.18653/v1/2022.acl-long.581. [Online]. Available: https://aclanthology.org/2022.acl-long.581.

[38] K. Yin and G. Neubig, "Interpreting language models with contrastive explanations," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 184–198. [Online]. Available: https://aclanthology.org/2022.emnlp-main.14.

[39] R. Li, W. Xiao, L. Xing, L. Wang, G. Murray, and G. Carenini, "Human guided exploitation of interpretable attention patterns in summarization and topic segmentation," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 10 189–10 204. [Online]. Available: https://aclanthology.org/2022.emnlp-main.694.

[40] P. Huber and G. Carenini, "Towards understanding large-scale discourse structures in pre-trained and fine-tuned language models," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 2376–2394. DOI: 10.18653/v1/2022.naacl-main.170. [Online]. Available: https://aclanthology.org/2022.naacl-main.170.

[41] X. Zhang, Y. Shen, Z. Huang, J. Zhou, W. Rong, and Z. Xiong, "Mixture of attention heads: Selecting attention heads per token," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 4150–4162. [Online]. Available: https://aclanthology.org/2022.emnlp-main.278.

[42] K. Deb, X. Zhang, and K. Duh, "Post-hoc interpretation of transformer hyperparameters with explainable boosting machines," in *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 51–61. [Online]. Available: https://aclanthology.org/2022.blackboxnlp-1.5.

[43] X. Wang, K. Wen, Z. Zhang, L. Hou, Z. Liu, and J. Li, "Finding skill neurons in pre-trained transformer-based language models," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 11 132–11 152. [Online]. Available: https://aclanthology.org/2022.emnlp-main.765.

[44] M. Hassid, H. Peng, D. Rotem, *et al.*, "How much does attention actually attend? questioning the importance of attention in pretrained transformers," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 1403–1416. [Online]. Available: https://aclanthology.org/2022.findings-emnlp.101.

[45] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "Race: Large-scale reading comprehension dataset from examinations," *arXiv preprint arXiv:1704.04683*, 2017.

[46] H. W. Chung, L. Hou, S. Longpre, *et al.*, *Scaling instruction-finetuned language models*, 2022. arXiv: 2210.11416 `[cs.LG]`.

[47] H. Touvron, T. Lavril, G. Izacard, *et al.*, *Llama: Open and efficient foundation language models*, 2023. arXiv: 2302.13971 `[cs.CL]`.

[48] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, *Activation functions in deep learning: A comprehensive survey and benchmark*, 2022. arXiv: 2109.14545 `[cs.LG]`.

[49] B. Zhang and R. Sennrich, *Root mean square layer normalization*, 2019. arXiv: 1910.07467 `[cs.LG]`.

[50] I. Beltagy, M. E. Peters, and A. Cohan, *Longformer: The long-document transformer*, 2020. arXiv: 2004.05150 `[cs.CL]`.

[51] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai, *Gqa: Training generalized multi-query transformer models from multi-head checkpoints*, 2023. arXiv: 2305.13245 `[cs.CL]`.

[52] R. Pope, S. Douglas, A. Chowdhery, *et al.*, *Efficiently scaling transformer inference*, 2022. arXiv: 2211.05102 `[cs.LG]`.