UNIVERSITY OF CALIFORNIA

Los Angeles

A Comparative Analysis of Fine-Tuned Llama 3-8B and DistilBERT for News Classification

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Applied Statistics and Data Science

by

Chenghao Sun

2024

ABSTRACT OF THE THESIS

A Comparative Analysis of Fine-Tuned
Llama 3-8B and DistilBERT
for News Classification

by

Chenghao Sun
Master of Applied Statistics and Data Science
University of California, Los Angeles, 2024
Professor Hongquan Xu, Chair

This thesis presents a comparative analysis of Llama 3-8B and DistilBERT language models for news classification across 26 classes. Utilizing a balanced dataset, we employed Low-Rank Adaptation (LoRA) for fine-tuning Llama 3-8B and traditional fine-tuning for DistilBERT. The study aims to evaluate the performance, efficiency, and practical applicability of these models in categorizing news articles.

Our experiments reveal that Llama 3-8B consistently outperforms DistilBERT in overall accuracy, achieving around 70% compared to DistilBERT's 60%. However, both models demonstrate competitive capabilities and exhibit distinct strengths across different news categories. The analysis uncovers significant variability in category-specific performance across multiple experimental runs, emphasizing the importance of robust evaluation procedures in model assessment.

The thesis of Chenghao Sun is approved.

Nicolas Christou

George Michailidis

Hongquan Xu, Committee Chair

University of California, Los Angeles

2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

The landscape of natural language processing (NLP) was forever changed in 2017 with the introduction of the Transformer model in the paper Attention Is All You Need [VSP17]. This breakthrough architecture, relying solely on self-attention mechanisms, eliminated the need for recurrence in sequence modeling tasks. By leveraging attention, the Transformer provided significant improvements in both performance and efficiency, laying the foundation for today's large language models (LLMs) such as BERT, GPT, and their successors. The ability to handle long-range dependencies and parallelize training has made the Transformer a central framework in NLP.

Since then, Language model pretraining has been shown to be effective for improving many natural language processing tasks [DCL18]. Models such as BERT pioneered the notion of bidirectional contextual comprehension, while GPT's autoregressive mechanism facilitated remarkable text generating capabilities. Recently, sophisticated models such as LLaMA (Large Language Model Meta AI) have enhanced fine-tuning techniques like LoRA (Low-Rank Adaptation) and quantisation. LoRA posits that the modifications to parameter matrices during fine-tuning exhibit low-rank properties [CCW24]. This facilitates the training and utilisation of models with millions to billions of parameters. Developed as a condensed version of BERT, DistilBERT showcases the emphasis on constructing compact and efficient models that preserve a significant portion of the original performance.

The purpose of this thesis is to compare and contrast two well-known models: DistilBERT and Llama 3-8B. At the forefront of LLM development, Llama 3-8B efficiently handles complex NLP problems by leveraging advanced optimisation techniques with 8 billion parameters pretrained. In comparison, DistilBERT provides a more efficient option, maintaining a significant portion of BERT's computational capabilities while using fewer resources. The primary objective of my research is to assess the performance of these models in text categorisation tasks, while exploring the compromises they make in terms of accuracy, speed, and memory consumption. The findings of this comparison analysis will offer valuable understanding of the suitability of these models in practical situations when both computing efficiency and accuracy are of utmost importance.

CHAPTER 2

Exploratory Data Analysis

The dataset consists of approximately 210k news headlines, titles, contents from the HuffPost, spanning from 2012 to 2022[Mis22]. This dataset provides a valuable resource for benchmarking NLP models, particularly for tasks such as text classification.

The dataset contains news articles from a variety of categories. The initial distribution of news categories is highly imbalanced, with some categories being overrepresented while others contain relatively few examples. This imbalance must be addressed before training models to avoid skewed results.

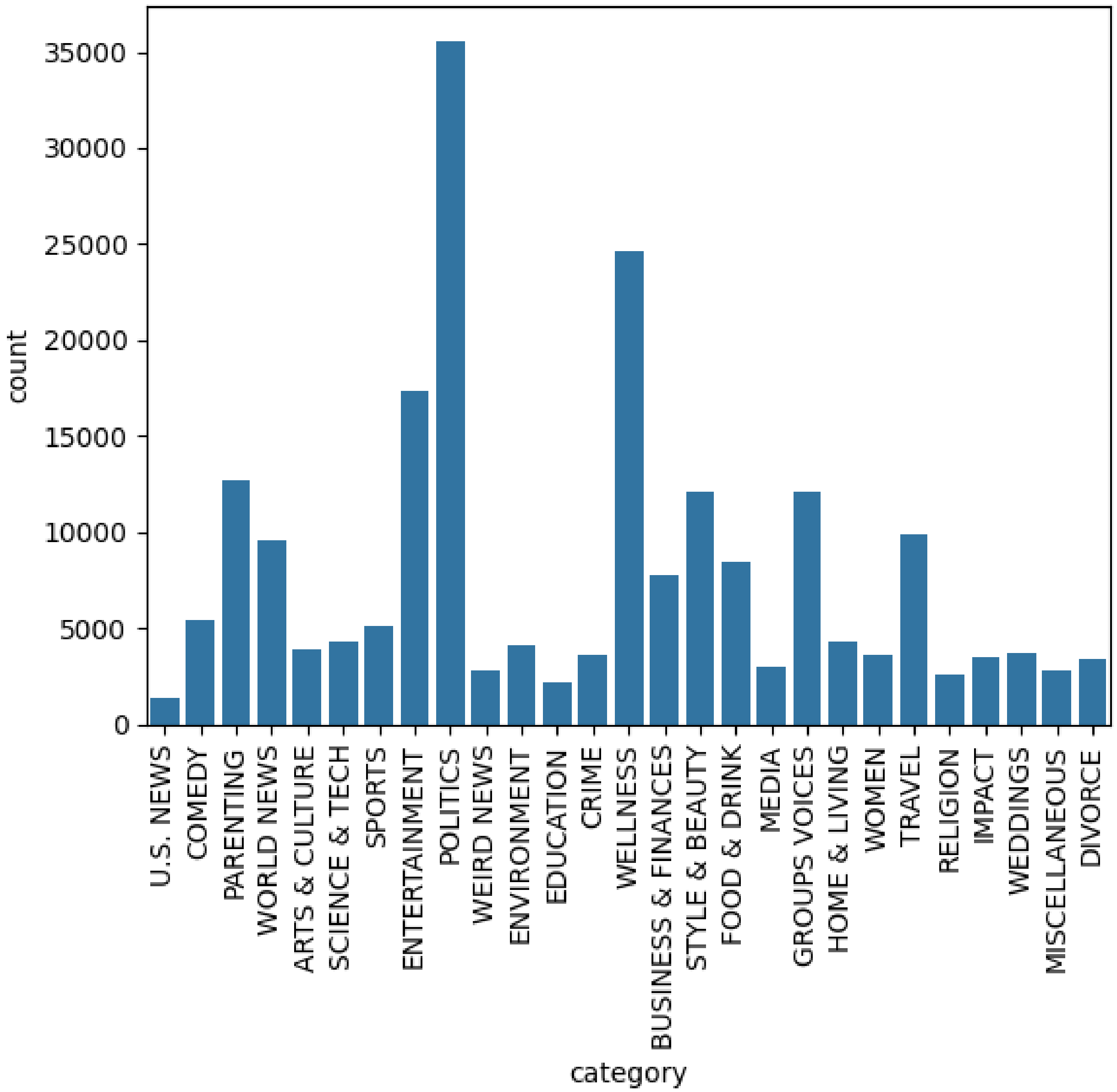

Figure 2.1: Distribution of News Categories in the Dataset

Figure 2.1 shows the distribution of news categories in the dataset. This figure reveals a notable disparity in the number of articles across various categories. In order to avoid imbalanced model, it is necessary to use balance techniques for fair model evaluation.

To mitigate the imbalance seen in the raw data, we used random sampling to achieve data balancing. After random sampling, the distribution of the news dataset. is more uniform, which is adequately represented during the training phase.

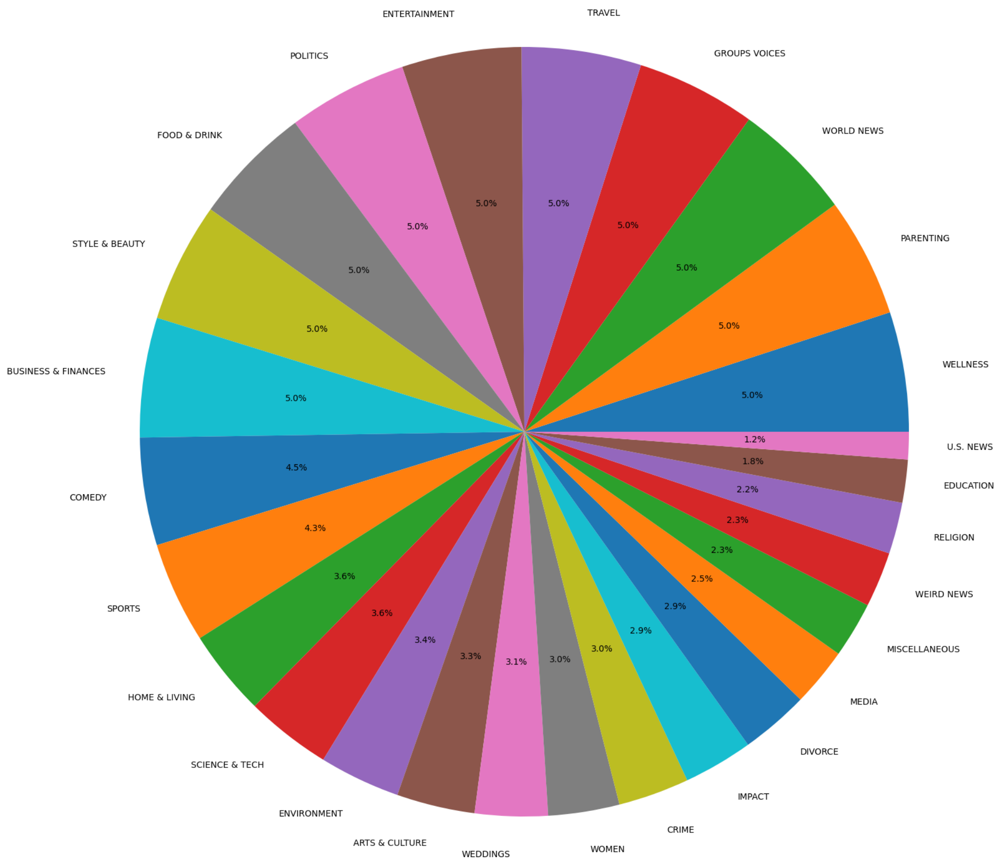

Figure 2.2: Pie Chart of News Categories After Data Balancing

Figure 2.2 shows the distribution of the dataset after the balancing process. The new dataset could improve the model's generalisation across all categories.

Moreover, The data's average length of titles across categories help identify potential differences in headlines. The structure of different categories' headline could be different.

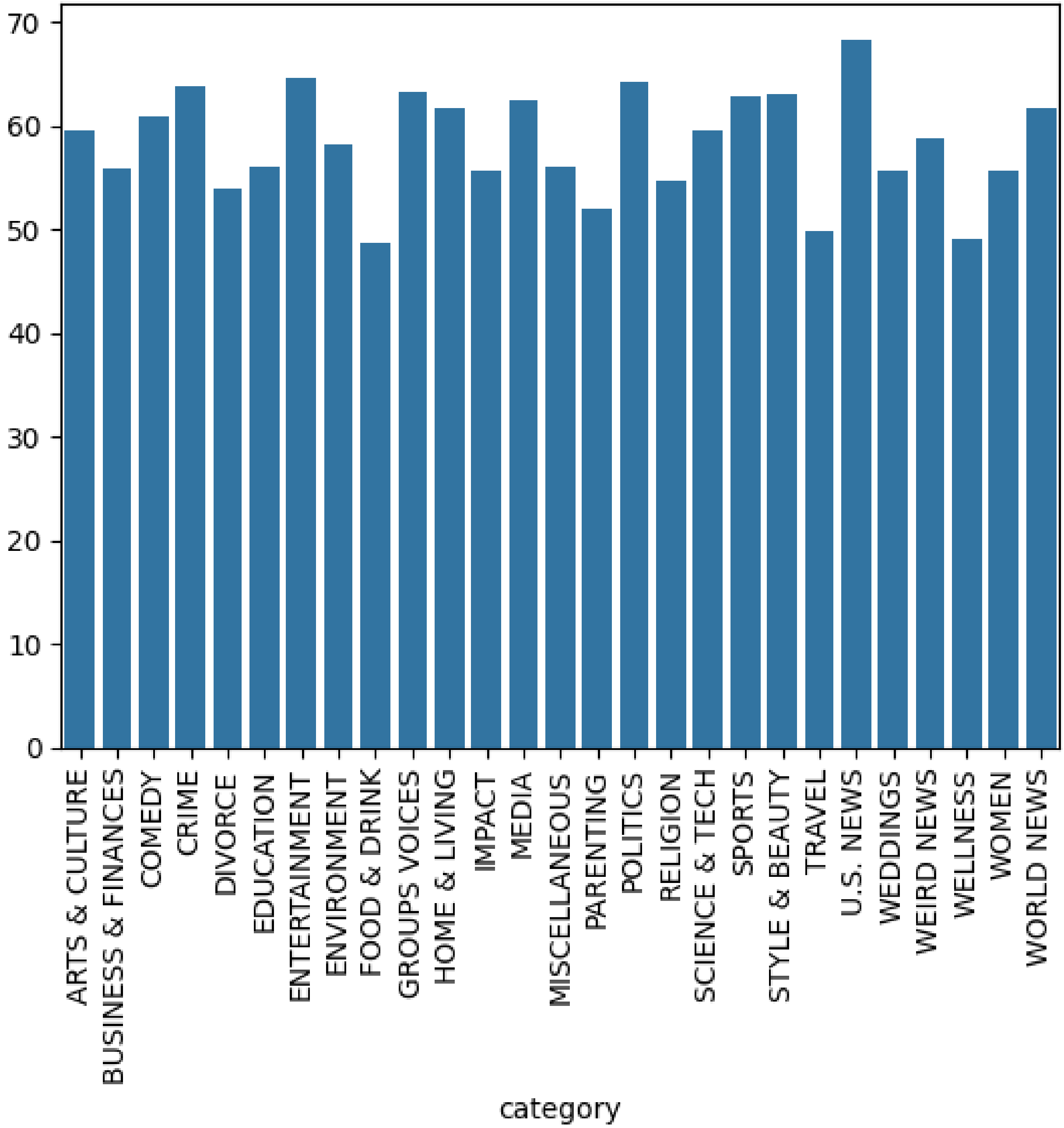

Figure 2.3: Average Length of Title for Each Category

Figure 2.3 is the average length of titles for each category that could provide further insight into this variation.

For each category of news data,the average paragraph length was analyzed to understand the hidden information associated with each class of news. It is possible that these features and depths could help model to assess the granularity of news articles in each category.

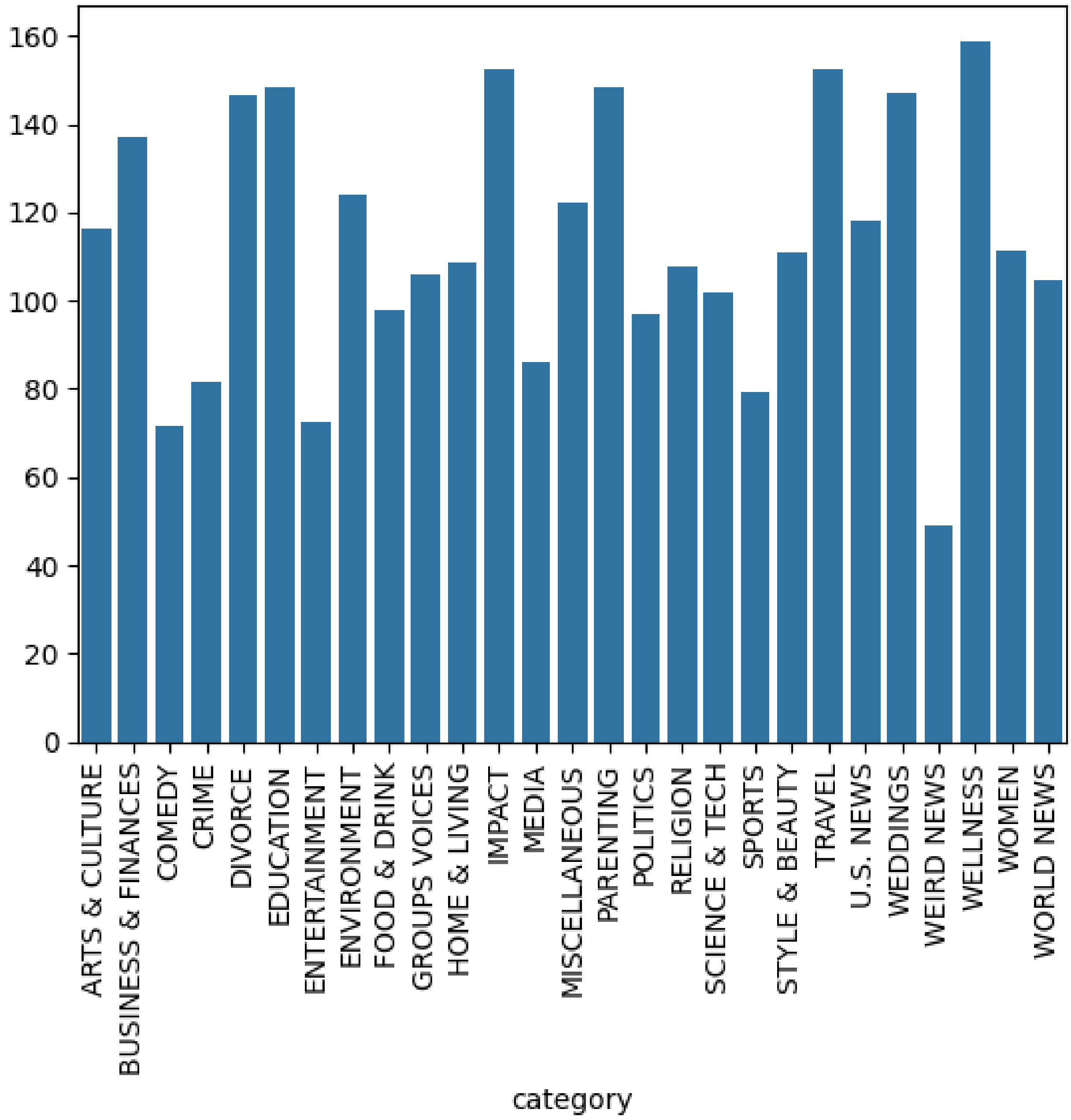

Figure 2.4: Average Length of Paragraph for Each Category

Figure 2.4 illustrates the average length of paragraphs for each category. Similar to the average title length, the paragraph length of news articles may exhibit different patterns or styles across categories.

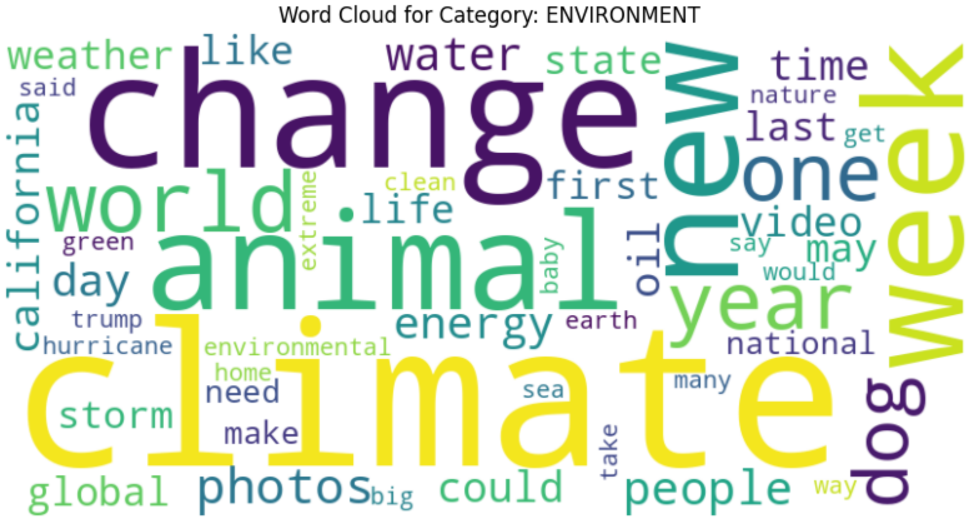

Figure 2.5: Word Cloud for Category Environment

Figure 2.5 is a visual representation of word frequency and relevance, offering an intuitive representation of the important features in the Environment category. Obviously, the climate change, animal are highly related with the category Environment.

CHAPTER 3

Methodology

3.1 Overview of the Transformer Architecture

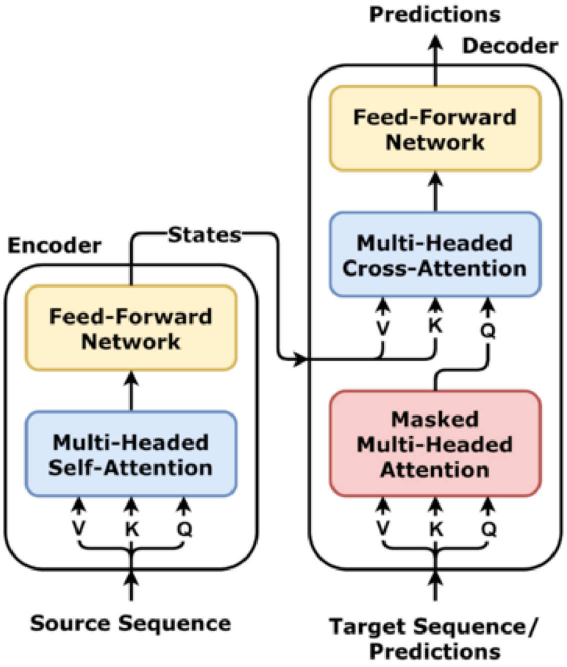
Predictions
Decoder
Feed-Forward Network
Multi-Headed Cross-Attention
V
K
Q
Masked Multi-Headed Attention
V
K
Q
Target Sequence/ Predictions
Encoder
States
Feed-Forward Network
Multi-Headed Self-Attention
V
K
Q
Source Sequence

Figure 3.1: Transformer Model Structure [Dvg]

As the plot shows, Google's Transformer model's two main components are the encoder and the decoder. This architecture is designed to process sequential data, such as text. With this structure, the model will not need to rely on recurrence (RNN) or convolution (CNN).

3.1.1 Encoder

The encoder, the Transformer's initial component, converts input sequences into rich, contextual representations. This representation encapsulates not just the surface-level information of each token, but also its relationship to other tokens in the sequence. The encoder consists of several key elements:

- **Input Embedding Layer**: This is the initial layer of transforms. In this layer, each input token (typically a word or subword) transfers into a dense vector representation. Word embedding is a process of representing words as numerical vectors that capture their semantic and syntactic properties [GRN24].
- **Positional Encoding**: In this layer, transformer processes all tokens in parallel, which could potentially lead to a loss of sequence order information. Positional encodings are added to the input embeddings to reduce information loss. These encodings inject information about the position of each token in the sequence, allowing the model to consider the order of words.
- **Multiple Identical Layers**: The heart of the encoder consists of a stack of identical layers. Each of these layers contains two main sub-components:
  - **Multi-head Self-Attention Mechanism**: This is perhaps the most innovative aspect of the Transformer. The self-attention mechanism allows each position in the input sequence to attend to all positions in the same sequence. This means that when processing a particular word, the model can directly consider the context provided by all other words, regardless of their distance in the sequence. The "multi-head" aspect allows the model to attend to different types of relationships in parallel.
  - **Position-wise Fully Connected Feed-Forward Network**: Following the attention mechanism, each position is processed independently through a feed-

forward neural network. This network typically consists of two linear transformations with a ReLU activation in between, allowing the model to introduce non-linearity and increase its representational power.

Each layer in the encoder processes its input through the self-attention mechanism and the feed-forward network, with residual connections and layer normalization applied after each sub-component. This structure allows the model to build up increasingly abstract and context-aware representations of the input sequence as it progresses through the layers.

3.1.2 Decoder

The decoder, the second major component of the Transformer, is tasked with generating the output sequence based on the encoder's representation and the previously generated outputs. Its structure is similar to the encoder's, but with some crucial differences that allow it to generate coherent and contextually appropriate sequences. The decoder includes:

- **Output Embedding Layer**: Similar to the input embeddings in the encoder, this layer transforms each output token into a dense vector representation.
- **Positional Encoding**: As in the encoder, positional encodings are added to provide information about the sequence order.
- **Multiple Identical Layers**: Like the encoder, the decoder consists of a stack of identical layers. However, each decoder layer contains three main sub-components:
    - **Masked Multi-head Self-Attention Mechanism**: This mechanism is similar to the self-attention in the encoder, but with an important distinction. To prevent the decoder from attending to future positions during training (which would amount to cheating), this self-attention is masked. This masking ensures that the prediction for position i can depend only on the known outputs at positions less than i.

- **Multi-head Attention over the Encoder's Output**: This attention mechanism allows the decoder to focus on relevant parts of the input sequence. It takes the output of the encoder as its Key and Value, and the output of the masked self-attention layer as its Query.
- **Position-wise Fully Connected Feed-Forward Network**: As in the encoder, this network processes each position independently, introducing non-linearity and increasing the model's capacity.

The decoder's structure allows it to generate each output token while considering both the relevant parts of the input (via attention over the encoder's output) and the previously generated output tokens (via masked self-attention).

3.1.3 Final Linear and Softmax Layer

After the decoder stack, the Transformer applies a final linear transformation followed by a softmax function. This step converts the decoder's output into a probability distribution over the vocabulary, allowing the model to predict the most likely next token in the sequence.

3.2 Detailed Multi-Head Attention Calculation

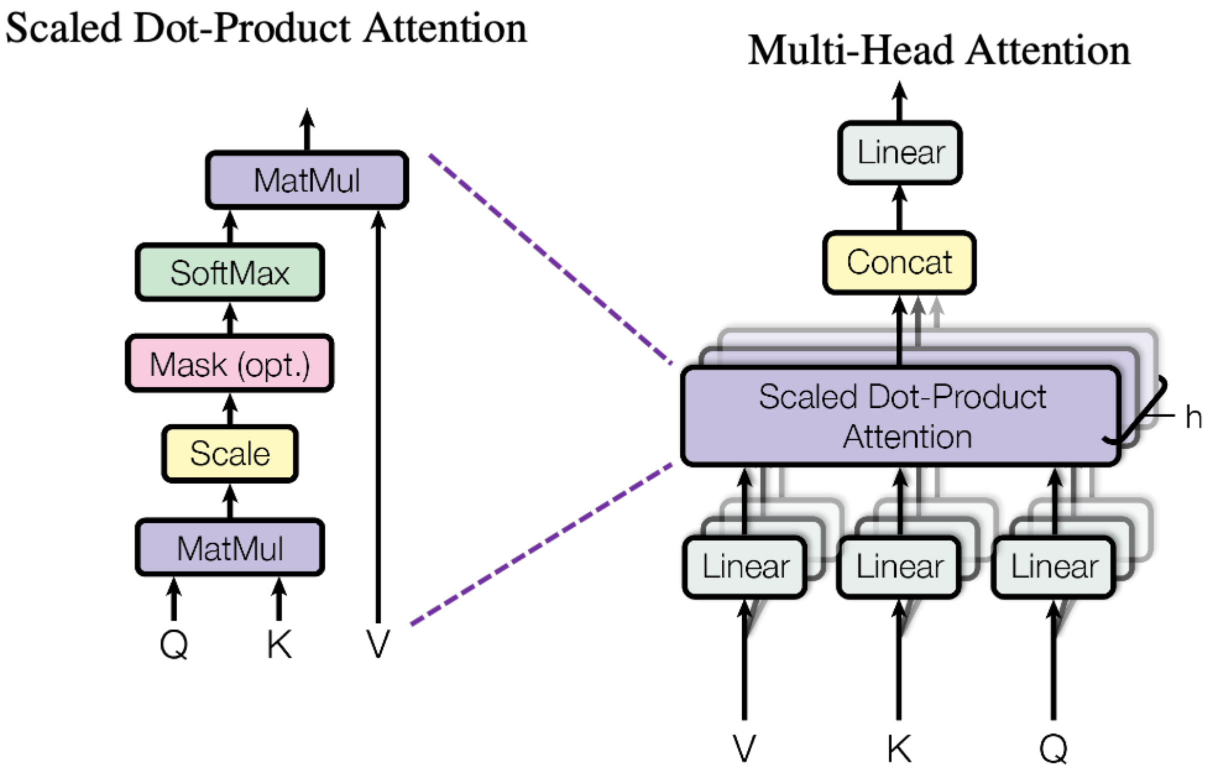

Figure 3.2: Multi-Head Attention Mechanism

The multi-head attention mechanism is the core innovation of the Transformer model. It allows the model to focus on different parts of the input sequence when producing each element of the output sequence.

Scaled Dot-Product Attention The fundamental building block of the attention mechanism is the scaled dot-product attention, calculated as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.1}$$

The multi-head attention extends the basic attention mechanism by allowing the model to attend to information from different representation subspaces at different positions. Here's a step-by-step breakdown of the calculation:

1. **Query (Q), Key (K), and Value (V)**: The input is transformed into three different representations: Q, K, and V. These three matrices have dimensions $n \times d$, where n is the sequence length (number of tokens), and d is the dimensionality of each token. In general, it could be think of the Query as a question, the Key as possible answers, and the Value as the information associated with each answer.

2. **Attention Scores**: The model compares each Query with all Keys to produce attention scores. The attention scores are calculated using the dot product of the Q and K matrices, scaled by the square root of the dimensionality d_k. These scores indicate the level of attentions that should be assigned to each part of the input when generating the output.

3. **Weighted Sum**: Then the attention mechanism calculates a weighted sum of the Values based on the attention scores. This weighted sum serves as the output for a single attention head.

4. **Multiple Heads**: The process of generating Queries, Keys, and Values is performed in parallel across multiple heads which allows the model to capture different types of relationships in the data.

5. **Combination**: Then, the individual outputs of all attention heads are merged to form the final output of the multi head attention mechanism.

By processing the input through multiple heads, the model is able to analyze various aspects of the data in parallel. Based on these, the model will result in a more refined and comprehensive understanding of the input.

This model can process various aspects of input data in parallel analysis attention heads attention mechanism. Based on these, the model will generate a more refined and comprehensive understanding of the input.

3.2.1 Position-wise Feed-Forward Networks

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{3.2}$$

The W_1 and W_2 are weight matrices, and b_1 and b_2 are bias vectors. The W_1 projects the input x from its original size to a larger intermediate space with a bias b_1. After this linear calculation, W_2 projects the output of ReLU back to the original dimensionality. The ReLU layer allows the model to introduce non-linearity and handle positional information separately. Similarly, a second bias term, b_2 is added after the calculation. Transformer applies this feedforward neural networks to each position after each attention layer.

3.2.2 Residual Connections and Layer Normalization

After sub-layer (attention or feed-forward), then the Transformer uses two techniques to help with training:

1. **Residual Connections**: The output of each sub layer is combined with its input to create a shortcut for information to flow directly through the network. By providing a direct path for information flow, it makes the model easier to learn.

2. **Layer Normalization**: Through standardized activation, this technology helps maintain training and supports deeper network architectures.

3.2.3 Final Output Generation

The outputs of decoder will be converted into word probabilities for the entire vocabulary at the end part. This is done using a linear transformation followed by a softmax function, which could ensure the probabilities sum to 1.

3.2.4 Training and Inference

In the training stage, the entire sequence is processed in parallel. To prevent attending to future positions, the decoder will set a suitable mask. While, during the inference process, the output token is generated one at a time through autoregression.

3.3 Llama 3 LoRA Fine-tuning

In 2024, Meta AI released Llama 3, specifically the 8-billion parameter variant (Llama 3-8B), which represents the latest advancement in the series of LLMs. Llama 3-8B balances computational efficiency with model size, positioning itself as a powerful yet manageable model for a wide range of natural language processing (NLP) tasks. Moreover, Llama 3 incorporates enhanced architectural design and training methodologies, resulting in improved performance across various language tasks [DJP24]. It strikes a middle ground between smaller models like DistilBERT and much larger models in the Llama 3 family, offering a strong performance across diverse tasks without the extreme resource demands of its larger counterparts.

One of the key challenges in utilizing large language models like Llama 3 family is the computational resource requirement for fine-tuning. Traditional fine-tuning methods involve updating all parameters of the model, which can be prohibitively expensive for models with billions of parameters. To address this issue, we employ Low-Rank Adaptation (LoRA), an efficient fine-tuning technique that significantly reduces the number of trainable parameters while maintaining performance.

3.3.1 LoRA Methodology

LoRA, introduced by Hu et al. (2021), In LoRA finetuning, for a given layer, only a low rank matrix called an adapter which is added to the pretrained weights, is trainable. [HGY24].

Instead of fine-tuning all parameters, LoRA freezes the pretrained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architecture.

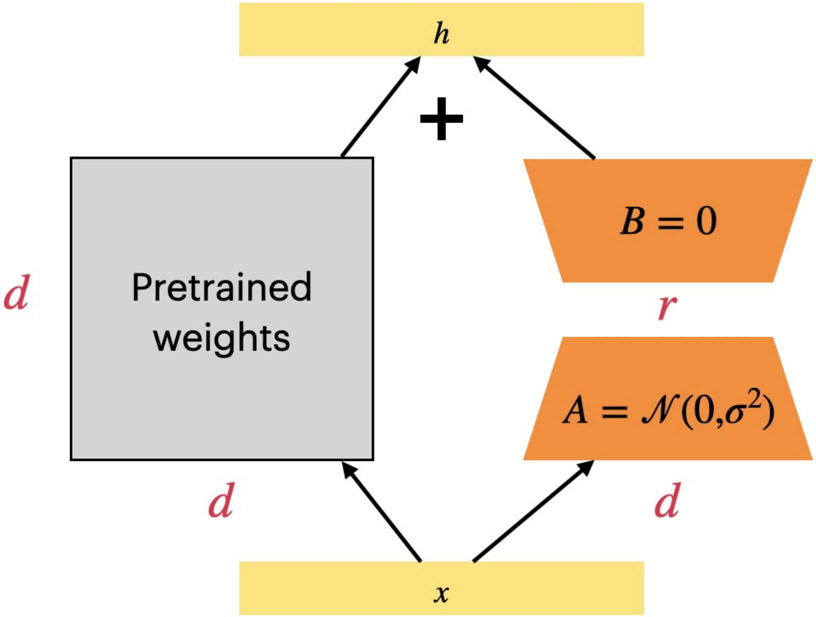

Figure 3.3: LoRA Fine-tuning Mechanism [HSW21]

As illustrated in Figure 3.3, LoRA introduces additional matrices B and A to the original weight matrix W. The computation in a LoRA-augmented layer can be expressed as:

$$\begin{aligned} h &= Wx + BAx \\ h &= \underbrace{(W + BA)}_{W_{\text{merged}}} x \end{aligned} \tag{3.3}$$

Where:

- $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ is the original pretrained weight matrix, where d_{out} is the output dimension and d_{in} is the input dimension.

- $B \in \mathbb{R}^{d_{\text{out}} \times r}$ is a low-rank matrix that projects the low-rank space back to the output dimension. Here, r is the rank of the adaptation. By initializing B to zero, the original pretrained weights W remain unchanged at the atart of the fine-tuning. As the training process, B will learn to project the low-rank adaption onto the output dimision.

- $A \in \mathbb{R}^{r \times d_{\text{in}}}$ is a low-rank matrix that projects the input x into a low-dimensional space of rank r. A is initialized from a random normal distribution begins with small values. In this way, A start exploring different directions in the input space with a stable convergence in neural networks.

- $x \in \mathbb{R}^{d_{\text{in}}}$ is the input vector.

- $h \in \mathbb{R}^{d_{\text{out}}}$ is the output of the layer.

The key advantage of this approach is that only the matrices B and A are trained, which are significantly smaller than W, thus reducing the number of trainable parameters and the computational resources required for fine-tuning.

3.3.2 Fine-tuning Process for Llama 3-8B

In our implementation, we use Llama 3-8B, a compact version of Llama 3 family, optimized for efficiency while retaining much of the performance of the larger model. The fine-tuning process involves the following steps and hyperparameters:

1. **Data Preparation**: The news classification dataset is preprocessed and tokenized using the Llama 3 tokenizer.

2. **LoRA Configuration**: We apply LoRA to the attention layers of Llama 3-8B. The rank r of the LoRA rank matrices is set to 8, and the alpha is set to 16. Alpha and R controls the contribution of these low-rank updates which provide a good balance

between model capacity and efficiency.

$$W_{\text{merged}} = W + \frac{\alpha}{r}(BA)$$

3. **Learning Rate**: We use a learning rate of 5e-5, which is higher than typical all fine-tuning rates due to the smaller number of parameters being updated.

4. **Batch Size**: To make balance between memory constraints and training stability, the batch size will not exceed 16

5. **Number of Epochs**: The model is trained for 3 epochs, which we found sufficient for convergence on our dataset.

6. **Optimizer**: The optimizer AdamW with a weight decay of 0.01 to prevent overfitting.

7. **Warmup**: A linear learning rate warmup is applied for the first 10 of training steps to stabilize early training.

8. **Gradient Clipping**: To prevent exploding gradients, we apply gradient clipping with a max norm of 1.0.

9. **LR Scheduler**: Using cosine learning scheduler to adjust learning rate

This fine-tuning process allows us to adapt the powerful Llama 3-8B model to our specific news classification task efficiently. By using LoRA, we can achieve comparable performance to all fine-tuning while significantly reducing computational requirements and training time.

The resulting fine-tuned model combines the broad knowledge captured in the pretrained Llama 3-8B with task-specific adaptations for news classification, enabling accurate and efficient categorization of news articles across various topics.

3.3.3 Prompt Engineering for Llama 3-8B

In addition to the LoRA fine-tuning process, we employed specific prompt engineering techniques to optimize Llama 3-8B's performance on our news classification task. Prompt engineering involves carefully structuring the input to the model to elicit the desired output format and improve task performance. While emerging prompt optimizers can automatically refine many of these aspects[MPS24]. For our news classification task, we implemented the following prompt structure:

- **Instruction Prefix**: Each conversation with the model begins with an instruction that outlines the available category options. This instruction is formatted as follows:

  ```
  instruction: "Options: MISCELLANEOUS, EDUCATION, FOOD & DRINK,
  ARTS & CULTURE, SCIENCE & TECH, ENVIRONMENT, WELLNESS,
  GROUPS VOICES, BUSINESS & FINANCES, PARENTING, WORLD NEWS,
  STYLE & BEAUTY"
  ```

- **Input Prefix**: Before the actual text to be classified, we add the phrase:

  ```
  "Select one category for this text:"
  ```

  This prompts the model to focus on the task of category selection.

- **Output Prefix**: To standardize the model's output format, we instruct it to begin its response with

  ```
  "I select ####"
  ```

where #### is to separate the chosen category.

A typical prompt sequence for our fine-tuned Llama 3-8B model would look like this:

```
instruction: "Options: MISCELLANEOUS, EDUCATION, FOOD & DRINK,
ARTS & CULTURE, SCIENCE & TECH, ENVIRONMENT, WELLNESS,
GROUPS VOICES, BUSINESS & FINANCES, PARENTING, WORLD NEWS,
STYLE & BEAUTY"

Input: Select one category for this text: California Gov. Calls On......

Output: I select #### U.S. News
```

This prompt engineering approach offers several advantages:

1. **Uniform Output**: Prompt changes will provide a clear instruction and defined output format. This process will ensure the model consistently generates responses, making parsing and evaluation more efficient.

2. **Task Focus**: The input prefix helps direct the model's attention specifically to the classification task. It also may improve its focus and accuracy.

3. **Controlled Output**: By Listing the available categories within the instruction, risk of generating irrelevant results is minimized. Model's output will have standard format.

4. **Adaptability**: The prompt engineering can be easily adapted for various classification tasks or modified to include additional context.

In the fine-tuning process, the model is trained to match this prompt structure. It also allow the model to learn from the desired input-output pattern. This approach, combined with the LoRA fine-tuning technique, enables us to efficiently train Llama 3-8B for our specific news classification task while maintaining a structured and consistent interaction format.

3.4 DistilBERT

DistilBERT is a lighter version of the original BERT model, designed to maintain similar performance while significantly reducing model size and computational overhead. DistilBERT represents a significant advancement in the pursuit of more efficient and deployable language models [SDC19].

3.4.1 Overview of DistilBERT

DistilBERT is built using knowledge distillation, where a smaller model (student) learns to replicate the behavior of a larger, more complex model (teacher). In this case, BERT serves as the teacher model for DistilBERT. The key aspects of DistilBERT include:

- **Architecture**: DistilBERT maintains the general architecture of BERT but with fewer layers. While BERT typically has 12 or 24 layers, it has 6 layers instead of 12 for the base model and 2,048 hidden units instead of 7,680 [VSP23].

- **Model Size**: DistilBERT has 40% fewer parameters than BERT-base, resulting in a more compact model that requires less memory and computational power.

- **Speed**: The reduced size allows DistilBERT to perform inference 60% faster than BERT while retaining 97% of its language understanding capabilities.

- **Training Objective**: DistilBERT is trained using a combination of language modeling, distillation, and cosine-distance losses, allowing it to effectively learn from both the teacher model and the training data.

3.5 Comparison between DistilBERT and Llama 3-8B

While both DistilBERT and Llama 3-8B are advanced language models, they differ significantly in their design philosophy, architecture, and intended use cases. Here are some key differences:

1. **Model Size and Complexity**:
   - Llama 3-8B is a much larger model, with 8 billions of parameters, designed to capture a vast amount of knowledge and handle a wide range of tasks.
   - DistilBERT is intentionally compact, with significantly fewer parameters, focusing on efficiency and deployability.
2. **Architecture**:
   - Llama 3-8B uses a decoder-only transformer architecture, which is well-suited for generative tasks.
   - DistilBERT, like BERT, uses an encoder-only architecture, which is particularly effective for understanding and representation tasks.
3. **Pre-training Approach**:
   - Llama 3-8B is pre-trained on a vast corpus of internet text using a causal language modeling objective.
   - DistilBERT is trained using knowledge distillation from BERT, combining mimicry of the teacher model with direct learning from data.
4. **Tokenization**:
   - Llama 3-8B typically uses BPE (Byte-Pair Encoding) tokenization, which can handle a wide range of languages and symbols efficiently.

- DistilBERT uses WordPiece tokenization, inherited from BERT, which is effective for many languages but may be less flexible for very diverse or multilingual corpora.

5. **Fine-tuning Approach**:
    - Llama 3-8B, being larger, often benefits from parameter-efficient fine-tuning methods like LoRA, as discussed in the previous section.
    - DistilBERT, being smaller, can often be fine-tuned using traditional methods without requiring special techniques for efficiency.

6. **Intended Use Cases**:
    - Llama 3-8B excels in tasks requiring broad knowledge and generation capabilities, such as open-ended question answering, text generation, and complex reasoning.
    - DistilBERT a particularly well-suited for tasks requiring language understanding in resource-constrained environments, such as sentiment analysis, named entity recognition, and text classification.

7. **Computational Requirements**:
    - Llama 3-8B requires significant computational resources for both training and prediction, often necessitating GPU or TPU acceleration.
    - DistilBERT can run efficiently on CPUs and require less powerful hardware. In this way, DistilBERT have more available situations.

In our news classification task, both models provide unique advantages. Firstly, the well structured and pretrained Llama 3-8B, shows certain level of knowledge and generative abilities, which clearly exceed the understanding capabilities of DistilBERT. On the other hand, DistilBERT, though more lightweight, is tailored for speed and high efficiency in classification tasks.

Our comparison of these models will shed light on the trade-offs between model complexity, computational efficiency, and classification accuracy. This comparison will provide valuable experience into the applicability of different model capacity for different practical NLP tasks.

CHAPTER 4

Experiment

4.1 Experimental Setup

For the full 26-category classification:

- The training data consisted of 26,000 rows, with 1,000 randomly selected articles from each of the 26 categories.
- The validation set contained approximately 13,000 articles, with around 500 randomly selected from each category.

For the 5-category and 10-category experiments:

- We selected the specified number of categories from the full set.
- The training and validation sets were created using the same proportion of articles per category as in the full experiment.

This approach ensures that each category is equally represented in both the training and validation sets across all experiments, mitigating potential biases due to class imbalance in the original dataset. The balanced subsets allow for a fair comparison of model performance across full categories and between different classification tasks.

By conducting experiments with varying numbers of categories (5, 10, and 26), we can assess how the models perform under different levels of task complexity, providing insights into their scalability and robustness.

4.2 Analysis of Model Training Processes

4.2.1 Llama 3-8B Training Dynamics for Full-Category Classification

Figure 4.1 illustrates the loss curve for Llama 3-8B over 250 training steps during the full 26-category classification task.

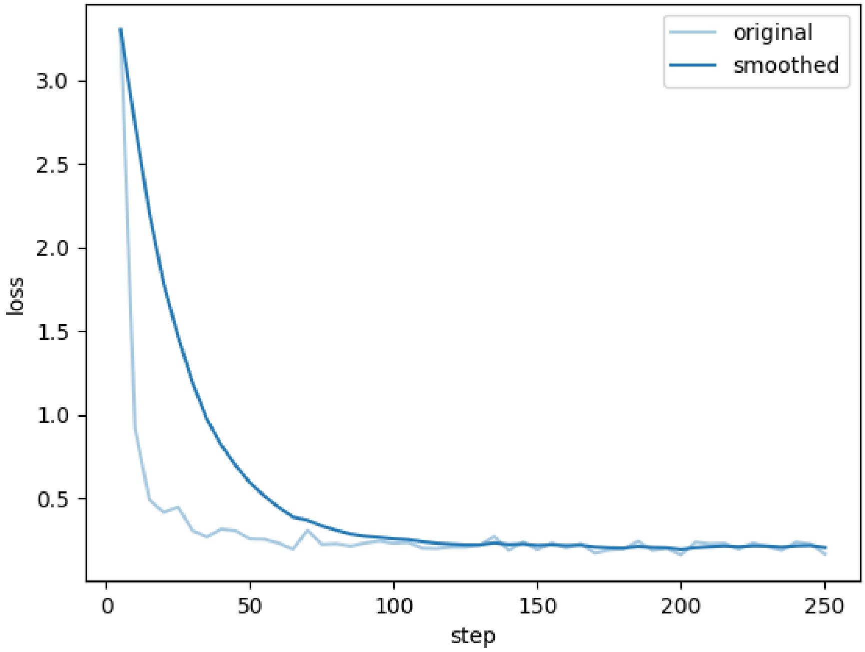

Figure 4.1: Llama 3-8B Training Loss Over Steps for Full-Category Classification

The Llama 3-8B model, when trained on all 26 categories, exhibited a rapid initial decrease in loss, followed by a more gradual decline and eventual stabilization:

- The loss started at approximately 3.5 and sharply decreased within the first 50 steps.
- After the initial rapid decline, the loss continued to decrease more gradually.
- The training process showed some fluctuations, as evident from the difference between the original and smoothed curves.

- Convergence was observed around step 150, with the loss stabilizing between 0.25 and 0.3.

This pattern suggests that Llama 3-8B quickly learned the most significant features of the data in the early stages of training, followed by fine-tuning and refinement in later steps, even when dealing with the complexity of all 26 categories.

4.2.2 BERT Training Progression for Full-Category Classification

Figure 4.2 shows the BERT model's loss over 10 epochs for the full 26-category classification task.

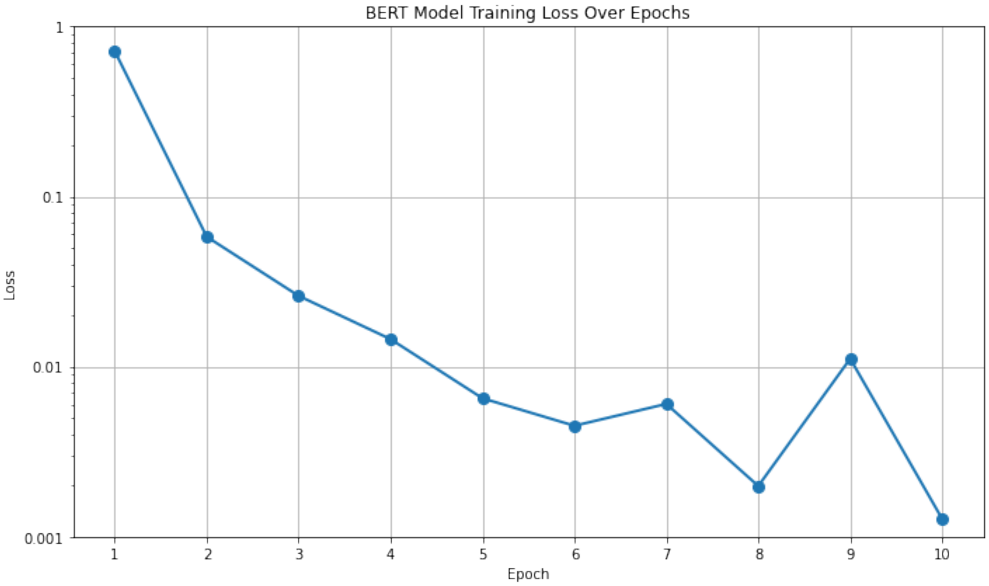

Figure 4.2: BERT Model Training Loss Over Epochs for Full-Category Classification

The BERT model demonstrated a consistent decrease in loss over the 10 epochs of training on all 26 categories:

- The initial loss was approximately 0.7, which decreased rapidly in the first few epochs.

- The rate of loss reduction slowed in later epochs, indicating diminishing returns as training progressed.

- Some fluctuations were observed, particularly a notable spike at epoch 9.

- The final loss at epoch 10 was approximately 0.001, suggesting strong convergence even for the complex full-category task.

The logarithmic scale of the loss axis highlights DistilBERT's ability to continually improve, even when dealing with very small loss values in later epochs and the complexity of classifying across all 26 categories.

4.2.3 Comparative Analysis

Both models showed effective learning, but with distinct characteristics:

- Initial Learning: Both models exhibited rapid initial learning, with Llama 3-8B showing a smoother decline compared to DistilBERT's more step-like reductions between epochs.

- Convergence Speed: Llama 3-8B appeared to reach a stable loss value earlier in its training process (around 60% of total steps), while BERT continued to improve over all 10 epochs.

- Final Performance: BERT achieved a lower final loss value, potentially indicating better fit to the training data. However, this should be validated against performance on a test set to ensure generalization.

- Stability: Llama 3-8B's smoothed curve suggested a more stable training process with fewer fluctuations compared to DistilBERT's epoch-to-epoch variations.

These differences in training dynamics may be attributed to the architectural differences between the two models, the nature of the pretraining tasks, or the specific datasets used for fine-tuning.

4.2.4 Implications for Model Selection

The analysis of training processes provides insights for model selection and deployment:

- Llama 3-8B's quicker stabilization might be advantageous in scenarios requiring faster training or deployment cycles.
- DistilBERT's continued improvement over epochs suggests it might benefit from extended training periods, potentially achieving higher accuracy at the cost of increased computational resources.
- The stability of Llama 3-8B's training process could be preferable in applications where consistent performance is critical.
- DistilBERT's lower final loss indicates it might offer superior performance in tasks closely aligned with its training objective, though this should be verified on held-out test data.

Further investigation into the models' performance on validation and test sets, as well as task-specific metrics, would provide a more comprehensive basis for model selection in practical applications.

4.3 Llama 3-8B Performance

We also evaluated the performance of the Llama 3-8B model on the same balanced news classification dataset across three different classification tasks: 5-category, 10-category, and full

26-category classification.This comprehensive approach allows us to assess how the models perform under varying levels of task complexity.

Table like 4.1 presents the detailed results for each category, as well as the overall performance metrics. Precision is the proportion of true positive predictions among all positive predictions made by the model. Recall is the proportion of true positives out of all actual positives. The F1-score is the harmonic mean of precision and recall, balancing both metrics to provide a single measure of model performance. The table also includes micro-average, which aggregates the contributions of all classes to compute a single precision, recall, or F1-score.

- 5-Category Classification: Table 4.1 presents the detailed results for each of the five categories.
- 10-Category Classification: Table 4.2 shows the performance across ten categories.
- Full 26-Category Classification: Table 4.3 presents the detailed results for each of the 26 categories, and Table 4.4 shows the aggregate scores.

4.3.1 5-Category Classification

For the 5-category classification task, Llama 3-8B achieved an impressive overall accuracy of 90.5%. This high performance indicates the model's strong capability in distinguishing between a limited number of news categories.

4.3.2 10-Category Classification

When expanding the task to 10 categories, Llama 3-8B maintained a robust performance with an overall accuracy of 83.7%. While this represents a decrease from the 5-category scenario, it's still a strong result considering the increased complexity of the task.

Table 4.2 presents the detailed classification report for the 10-category experiment:

Table 4.1: Llama 3-8B Classification Report for 5 Categories

Category	**Precision**	**Recall**	**F1-score**
ARTS & CULTURE	0.90	0.92	0.91
CRIME	0.95	0.97	0.96
IMPACT	0.83	0.89	0.86
RELIGION	0.96	0.81	0.88
SCIENCE & TECH	0.89	0.92	0.90
Macro Avg	0.90	0.90	0.90

4.3.3 Full-Category Classification

The Llama 3-8B model achieved an overall accuracy of 70.22% on the validation set. As shown in Table 4.3, the model's performance also varied across different categories. Categories like HOME & LIVING and WEDDINGS showed high F1-scores of 0.89 and 0.82 respectively, indicating strong performance. However, categories such as MISCELLANEOUS and U.S. NEWS had lower F1-scores of 0.39 and 0.42, suggesting areas for improvement.

The macro-average F1-score of 0.68 suggests that the model performs consistently across full categories when each category is given equal weight. The weighted average F1-score of 0.70 indicates slightly better performance when considering the support of each category.

4.4 DistilBERT Performance

We also evaluated the performance of the DistilBERT model on the same balanced news classification dataset across three classification tasks: 5-category, 10-category, and full 26-category classification. This comprehensive analysis allows us to assess DistilBERT's performance across different levels of task complexity, providing insights into its scalability and robustness in news classification tasks.

Table 4.2: Llama 3-8B Classification Report for 10 Categories

Category	**Precision**	**Recall**	**F1-score**
ARTS & CULTURE	0.85	0.85	0.85
CRIME	0.87	0.92	0.89
EDUCATION	0.73	0.85	0.79
ENVIRONMENT	0.81	0.81	0.81
FOOD & DRINK	0.90	0.92	0.91
IMPACT	0.71	0.65	0.68
RELIGION	0.87	0.81	0.84
SCIENCE & TECH	0.80	0.75	0.78
SPORTS	0.92	0.96	0.94
WELLNESS	0.76	0.73	0.75
Macro Avg	0.83	0.82	0.82

- 5-Category Classification: Table 4.5 presents the detailed results for each of the five categories.

- 10-Category Classification: Table 4.6 shows the performance across ten categories.

- Full 26-Category Classification: Table 4.7 presents the detailed results for each of the 26 categories, as well as the overall performance metrics.

4.4.1 5-Category Classification

For the 5-category classification task, DistilBERT achieved an overall accuracy of 78%. This performance demonstrates the model's strong capability in distinguishing between a limited number of news categories.

Table 4.3: Llama 3-8B Performance on News Classification Task

Category	**Precision**	**Recall**	**F1-Score**
ARTS & CULTURE	0.75	0.67	0.71
BUSINESS & FINANCES	0.61	0.53	0.57
COMEDY	0.65	0.59	0.62
CRIME	0.74	0.63	0.68
DIVORCE	0.85	0.89	0.87
EDUCATION	0.84	0.77	0.80
ENTERTAINMENT	0.57	0.64	0.60
ENVIRONMENT	0.65	0.65	0.65
FOOD & DRINK	0.80	0.93	0.86
GROUPS VOICES	0.64	0.67	0.65
HOME & LIVING	0.93	0.92	0.94
IMPACT	0.58	0.55	0.57
MEDIA	0.80	0.76	0.78
MISCELLANEOUS	0.46	0.39	0.42
PARENTING	0.68	0.74	0.71
POLITICS	0.62	0.56	0.59
RELIGION	0.71	0.81	0.75
SCIENCE & TECH	0.61	0.73	0.72
SPORTS	0.78	0.89	0.83
STYLE & BEAUTY	0.88	0.78	0.83
TRAVEL	0.91	0.83	0.87
U.S. NEWS	0.46	0.42	0.44
WEDDINGS	0.82	0.89	0.87
WELLNESS	0.67	0.61	0.64
WOMEN	0.47	0.62	0.54
WORLD NEWS	0.78	0.84	0.82

Table 4.4: Llama 3-8B Aggregate Scores

Metric	**Precision**	**Recall**	**F1-Score**
Macro Avg	0.68	0.68	0.68

Table 4.5: DistilBERT Classification Report for 5 Categories

Category	Precision	Recall	F1-score
ARTS & CULTURE	0.79	0.77	0.78
CRIME	0.82	0.80	0.81
IMPACT	0.75	0.78	0.76
RELIGION	0.80	0.76	0.78
SCIENCE & TECH	0.77	0.79	0.78
Macro Avg	0.79	0.78	0.78

4.4.2 10-Category Classification

When expanding the task to 10 categories, DistilBERT maintained a robust performance with an overall accuracy of 71%. While this represents a decrease from the 5-category scenario, it's still a strong result considering the increased complexity of the task.

4.4.3 Full 26-Category Classification

The DistilBERT model achieved an overall accuracy of 60% on the validation set. As shown in Table 4.7, the model's performance varied across different categories. Some categories, such as CRIME and U.S. NEWS, showed high F1-scores of 0.84 and 0.87 respectively, indicating strong performance. However, other categories like MEDIA and TRAVEL had lower F1-scores of 0.44 and 0.47, suggesting room for improvement in these areas.

The micro-average F1-score of 0.63 indicates that the model's overall performance is consistent across categories, considering the total number of true positives, false negatives, and false positives. The macro-average F1-score, also 0.63, suggests that the model performs

Table 4.6: DistilBERT Classification Report for 10 Categories

Category	Precision	Recall	F1-score
ARTS & CULTURE	0.74	0.72	0.73
CRIME	0.76	0.78	0.77
EDUCATION	0.68	0.70	0.69
ENVIRONMENT	0.72	0.70	0.71
FOOD & DRINK	0.75	0.77	0.76
IMPACT	0.67	0.65	0.66
RELIGION	0.73	0.71	0.72
SCIENCE & TECH	0.70	0.72	0.71
SPORTS	0.78	0.80	0.79
WELLNESS	0.69	0.67	0.68
Macro Avg	0.72	0.72	0.72

similarly across full categories when each category is given equal weight.

It's worth noting that the support for each category in the validation set is relatively balanced, ranging from 474 to 528 samples, which aligns with our intentional balancing of the dataset.

4.5 Comparison of DistilBERT and Llama 3-8B

Both DistilBERT and Llama 3-8B models demonstrated competitive performance across the 5-category, 10-category, and all 26-category news classification tasks. Llama 3-8B consistently outperformed DistilBERT in overall accuracy, but the performance gap varied depending on the number of categories. It's important to note that multiple experiments revealed some variability in results, particularly at the category level.

Table 4.7: DistilBERT Performance on News Classification Task

Category	Precision	Recall	F1-Score
ARTS & CULTURE	0.56	0.54	0.55
BUSINESS & FINANCES	0.72	0.43	0.54
COMEDY	0.61	0.70	0.65
CRIME	0.88	0.80	0.84
DIVORCE	0.78	0.56	0.66
EDUCATION	0.55	0.53	0.54
ENTERTAINMENT	0.64	0.64	0.64
ENVIRONMENT	0.89	0.74	0.81
FOOD & DRINK	0.57	0.49	0.53
GROUPS VOICES	0.80	0.77	0.79
HOME & LIVING	0.51	0.50	0.51
IMPACT	0.74	0.56	0.63
MEDIA	0.46	0.43	0.44
MISCELLANEOUS	0.60	0.60	0.60
PARENTING	0.46	0.66	0.54
POLITICS	0.81	0.69	0.74
RELIGION	0.62	0.52	0.57
SCIENCE & TECH	0.78	0.71	0.74
SPORTS	0.86	0.71	0.78
STYLE & BEAUTY	0.79	0.68	0.73
TRAVEL	0.64	0.37	0.47
U.S. NEWS	0.88	0.85	0.87
WEDDINGS	0.50	0.56	0.53
WELLNESS	0.44	0.57	0.50
WOMEN	0.73	0.69	0.71
WORLD NEWS	0.59	0.64	0.55

Table 4.8: DistilBERT Aggregate Scores

Metric	**Precision**	**Recall**	**F1-Score**
Macro Avg	0.66	0.61	0.63

4.5.1 Overall Performance

- **Accuracy:**
  - 5-Category Classification:
    - DistilBERT: 78% (Standard Error = 0.58%)
    - Llama 3-8B: 90.5% (Standard Error = 0.72%)
  - 10-Category Classification:
    - DistilBERT: 71% (Standard Error = 0.63%)
    - Llama 3-8B: 83.7% (Standard Error = 0.81%)
  - 26-Category Classification:
    - DistilBERT: 60.06% (Standard Error = 0.67%)
    - Llama 3-8B: 70.22% (Standard Error = 0.89%)

To ensure the robustness of the results, each experiment was repeated five times for both models on an NVIDIA V100 GPU. DistilBERT was trained for 10 epochs, while Llama 3-8B was trained for 3 epochs in each trial. Both models showed observable variability in category-level performance across different runs, with Llama 3-8B exhibiting slightly higher variability (standard deviation of 2-3% in overall accuracy) compared to DistilBERT (standard deviation of 1-2%).

- **Full-Category Training Time per Epoch:**
  - DistilBERT: **10 minutes per epoch** on average, resulting in a total training time of 100 minutes for 10 epochs.

  - Llama 3-8B: **2 hours per epoch** on average, resulting in a total training time of 6 hours for 3 epochs.

The significantly faster training time of DistilBERT makes it a more practical choice for quick iterations and tasks where speed and efficiency are prioritized. However, the longer training time of Llama 3-8B is justified by its more advanced architecture and larger model size, which allows it to capture more complex patterns in the data, reflected in its consistently higher accuracy across full classification tasks.

4.5.2 Performance Across Different Classification Tasks

1. **5-Category Classification:**

   - Both models performed exceptionally well, with Llama 3-8B achieving 90.5% accuracy and DistilBERT reaching 78%.
   - The performance gap between the two models was most pronounced in this simpler task, suggesting Llama 3-8B's superior ability to discriminate between a small number of categories.

2. **10-Category Classification:**

   - Performance decreased for both models as expected, with Llama 3-8B achieving 83.7% accuracy and DistilBERT reaching 71%.
   - The relative performance difference between the models remained consistent, with Llama 3-8B maintaining its advantage.

3. **26-Category Classification:**

   - The full category task proved most challenging, with Llama 3-8B achieving 70.22% accuracy and DistilBERT reaching 60.06%.

- The performance gap narrowed in this more complex task, suggesting that both models face similar challenges in highly granular classification.

4.5.3 Category-specific Performance

Both models exhibited varying performance across different categories, with some notable consistencies and differences:

1. **High-performance categories:**
   - DistilBERT consistently performed well in CRIME and U.S. NEWS (F1-scores of 0.84 ± 0.05 and 0.87 ± 0.04 respectively) in the 26-category task.
   - Llama 3-8B showed strong performance in HOME & LIVING and WEDDINGS (F1-scores of 0.94 ± 0.03 and 0.87 ± 0.05 respectively) in the 26-category task.
   - In the 5-category task, both models performed exceptionally well in the CRIME category (F1-scores above 0.90 for both).
2. **Low-performance categories:**
   - DistilBERT struggled with MEDIA and TRAVEL (F1-scores of 0.44 ± 0.03 and 0.47 ± 0.05) in the 26-category task.
   - Llama 3-8B faced challenges with MISCELLANEOUS and U.S. NEWS (F1-scores of 0.42 ± 0.06 and 0.44 ± 0.05) in the 26-category task.
   - In the 10-category task, both models showed relatively lower performance in the IMPACT category.
3. **Variability across experiments:**
   - Some categories showed higher variability across experiments. For example, the POLITICS category had F1-scores ranging from 0.65 to 0.83 for Llama 3-8B and 0.44 to 0.63 for DistilBERT across different runs in the 26-category task.

- The variability across different categories was more pronounced than the overall variability, suggesting that some categories benefit more from the models' architectural strengths.

4.5.4 Precision vs. Recall

In general, Llama 3-8B showed a better balance between precision and recall across most categories in full classification tasks, contributing to its consistently higher overall accuracy. Although DistilBERT showed more variance in precision and recall scores across categories and experiments, its efficiency in terms of training time and computational resources remains valuable, especially for tasks with fewer categories or when rapid deployment is necessary.

The performance of both models degraded as the number of categories increased, but Llama 3-8B maintained its edge in accuracy and F1-score across full tasks. This suggests that Llama 3-8B's larger size and more advanced architecture provide a significant advantage in capturing the nuances required for fine-grained classification, while DistilBERT offers a compelling trade-off between performance and efficiency, particularly in scenarios with fewer categories or limited computational resources.

CHAPTER 5

Conclusion and Discussion

Our examination of the DistilBERT and Llama 3-8B models for news classification has uncovered critical insights into their respective capabilities. First of all, Llama 3-8B consistently outperforms DistilBERT in terms of overall accuracy and macro-average F1-score. However, this advantage is not uniform across all news categories. Both models demonstrate competitive performance in specific classification tasks, suggesting that optimal model selection may depend on the particular application or focus areas within news categorization.

A key finding from our experiments is the significant variability in category-specific performance across multiple runs. Although overall metrics like accuracy and F1-score remain relatively stable, individual category results fluctuate notably. This variability underscores the importance of running multiple experiments to account for possible variations and ensuring that the chosen model is consistently reliable for critical categories.

When deciding between these models, trade-offs extend beyond mere accuracy. DistilBERT, being a smaller and more efficient model, offers clear advantages in terms of inference speed and lower computational resource demands. In contrast, Llama 3-8B provides higher overall accuracy but at the cost of increased computational demands. Therefore, the choice between the two involves balancing the need for computational efficiency against the requirement for precision, especially in high-stakes or complex classification contexts.

Finally, the variability observed in the category-specific results calls for further investigation. Variables such as dataset characteristics, model architectures, and training methodologies may contribute to these inconsistencies. Gaining a deeper understanding of these factors could lead to more effective model selection and optimization strategies. By combining the predictions from both Llama 3-8B and DistilBERT, an ensemble could take advantage of each model's specific strengths across different categories, potentially leading to improved overall performance and robustness in real-world applications. Further investigation into this approach could yield valuable improvements for news classification systems.

In conclusion, while Llama 3-8B demonstrates superior overall efficacy in news classification tasks, the decision to use Llama 3-8B or DistilBERT should be informed by specific application needs, computational limitations, and the relative importance of various news categories. The variability in results highlights the need for robust evaluation practices, and the exploration of ensemble methods could offer a promising path forward for enhancing performance in practical classification systems.

REFERENCES

[CCW24] Yupeng Chang, Yi Chang, and Yuan Wu. “Bias-Aware Low-Rank adaptation: Mitigating catastrophic inheritance of large language models.” *arXiv preprint arXiv:2408.04556*, 8 2024.

[DCL18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” *arXiv preprint arXiv:1810.04805*, 10 2018.

[DJP24] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and ... Al-Dahle. “The Llama 3 herd of models.” *arXiv preprint arXiv:2407.21783*, 7 2024.

[Dvg] Dvgodoy. “GitHub - dvgodoy/dl-visuals: Over 200 figures and diagrams of the most popular deep learning architectures and layers FREE TO USE in your blog posts, slides, presentations, or papers.” GitHub.

[GRN24] Hossein Salahshoor Gavalan, Mohmmad Naim Rastgoo, and Bahareh Nakisa. “A BERT-Based Summarization approach for depression detection.” *arXiv preprint arXiv:2409.08483*, 9 2024.

[HGY24] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. “The impact of initialization on LORA finetuning dynamics.” *arXiv preprint arXiv:2406.08447*, 6 2024.

[HSW21] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. “LORA: Low-Rank adaptation of Large Language Models.” *arXiv preprint arXiv:2106.09685*, 6 2021.

[Mis22] Rishabh Misra. “News Category Dataset.” Kaggle, 2022.

[MPS24] Qianou Ma, Weirui Peng, Hua Shen, Kenneth Koedinger, and Tongshuang Wu. “What you say = what you want? Teaching humans to articulate requirements for LLMs.” *arXiv preprint arXiv:2409.08775*, 9 2024.

[SDC19] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.” *arXiv preprint arXiv:1910.01108*, 10 2019.

[VSP17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention is All You Need.” *arXiv preprint arXiv:1706.03762*, 2017.

[VSP23] Sahana V, Nagamani H Shahapure, Rekha Pm, Nethravathi B, Pratiksha Khandelwal, Abhinav Anand, Pranjal Agrawal, and Vedant Srivastava. “The DistilBERT model: a promising approach to improve machine reading comprehension models.” *International Journal on Recent and Innovation Trends in Computing and Communication*, **11**(8):293–309, 9 2023.