CS7313: Advanced Artificial Intelligence

Project 4: Attention and Application of Transformer Models

This project has two parts:

1. **Attention:** You will implement the attention mechanism as described in the paper *"Attention is All You Need"* (2017). The paper is available here: https://arxiv.org/pdf/1706.03762.pdf.
2. **Application of Transformer Models:** You will use a pre-trained transformer model (provided by Hugging Face Co; usage details).

Attention:

You will implement the attention mechanism as described in the paper *"Attention is All You Need"* (2017). The paper is available here: https://arxiv.org/pdf/1706.03762.pdf.

IMPORTANT: Make a copy of the Colab notebook before working on it.

The attention mechanism allows a model to focus on different parts of the input sequence when producing an output. In particular, **scaled dot-product attention** computes attention scores based on the compatibility between queries and keys, and uses them to weight the values.

Given matrices:

- Q: the matrix of queries
- K: the matrix of keys
- V: the matrix of values

The attention output is computed as:

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

where:

- $QK^\top$ computes the dot products between queries and keys.
- $\sqrt{d_k}$ is a scaling factor (where d_k is the dimension of the keys) to prevent the dot products from growing too large.
- The softmax function normalizes the attention scores across each query.
- These scores are used to take a weighted sum of the values V.

Multi-Head Attention

Instead of performing a single attention function, **multi-head attention** runs multiple attention operations in parallel, allowing the model to jointly attend to information from different representation subspaces. It works as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O$$

where each head is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

and W_i^Q, W_i^K, W_i^V, and W^O are learnable weight matrices.

Application of Transformer Models

You will use a pre-trained transformer model (provided by Hugging Face Co; see usage details) in this project. Follow the provided Google Colab project link.

IMPORTANT: Make a copy of the Colab notebook before working on it.

You will apply transformer models to complete the following tasks:

- **Task 1:** Text generation from multiple prompts consisting of 1, 5, and 10 words.
- **Task 2:** Sentiment analysis of user reviews of products from a selected Amazon product category.
- **Task 3:** Predicting missing words (masked language modeling).

The datasets for Task 3 are available and attached to the relevant Harvey post. There are three files:

- `masked_nouns.csv`
- `masked_verbs.csv`
- `masked_adjs.csv`

Each file contains 500 data pairs in the format:

```
"masked text", "target word"
```

The masked word is denoted as `__MASKED__`. You must replace this with the appropriate token using `tokenizer.mask_token` as specified in the Hugging Face documentation.

Report and Grading Rubric

Attention: Submit a report of at least two pages (1" margins, 12 pt font, A4 paper) that includes the visualization of your implemented multi-head attention using BERT weights for different layers. Describe how the model's focus shifts between layers. Describe any meaningful insights in the attention maps and compare the attention maps for different heads.

Application of Transformer Models: For each task, submit a report of at least one page (1" margins, 12 pt font, A4 paper) that includes:

- Representative outputs summarizing your work
- Your analysis and takeaways
- Suggested applications of these tasks or other tasks you have tried using the models