About Masked Language Modelling with BERT and Attention Visualisation

This project is a Python-based natural language processing (NLP) application that makes use of a pre-trained BERT masked language model to predict missing words in text and to generate visualisations of self-attention patterns across the model's layers and heads. The objective of this project was to explore how transformer-based language models such as BERT interpret text input, and to investigate their internal mechanisms through attention heatmaps.

The programme is built using PyTorch for tensor computations and deep learning, Hugging Face Transformers to access the BERT model and its tokenizer, and PIL (Python Imaging Library) for visualisation and image generation. It demonstrates both the predictive power of BERT and how its attention mechanism distributes focus across input tokens.

The main user interaction is through text input containing a mask token ([MASK]) that represents the missing word. The programme predicts the top three likely replacements for the masked word and generates attention diagrams that illustrate how the model distributes "attention" between tokens at each layer and head.

Techniques and Features

This project combines a variety of machine learning and visualisation techniques:

- Pre-trained Language Model (BERT):
  The programme uses *bert-base-uncased*, a widely adopted BERT variant. Rather than training a model from scratch, it leverages transfer learning by making use of a model already trained on a large corpus of English texts.
- Masked Language Modelling:
  The [MASK] token is inserted into the input text by the user. BERT predicts the most likely replacements for this token based on surrounding context. The top three prediction candidates are returned, showcasing BERT's contextual understanding.
- Tokenisation and One-Hot Processing:
  The Hugging Face tokenizer is used to split text into tokens, map them to IDs, and feed them into the model in a format suitable for PyTorch processing.
- Attention Mechanism Visualisation:
  The results include attention matrices from every layer and head in BERT. These are converted into attention heatmaps, represented as grids where lighter colours indicate stronger attention. Tokens are displayed along both axes for interpretability.
- Image Generation with PIL:
  Attention visualisations are drawn as images, with labels for each token along rows and columns. The diagrams are automatically saved with filenames that indicate both the layer and attention head number (e.g., Attention_Layer1_Head3.png).

Challenges and Problem Solving

Some of the main challenges encountered during development included:

- **Handling Input with Mask Tokens:**
  A check had to be added to ensure that user input contained exactly one [MASK] token, otherwise the model could not make meaningful predictions. This issue was solved by writing a helper function that identifies the mask token index.
- **Generating Interpretable Visualisations:**
  While BERT outputs raw attention weights as tensors, translating them into clear and interpretable diagrams required careful scaling and the use of appropriate greyscale shading. The solution was to convert scores into values between 0 and 255, representing shades of grey.
- **Balancing Legibility and Layout:**
  Drawing all tokens as labels in both horizontal and rotated vertical positions was a challenge due to varying text lengths. This was solved by aligning text using bounding boxes and ensuring consistent grid sizing.

Through addressing these challenges, I gained deeper experience in handling raw transformer outputs and translating abstract model internals into visual, human-readable forms.

Reflection

This project allowed me to experiment directly with transformer models, one of the most influential architectures in modern AI. Working with BERT exposed me to not only masked language modelling, but also the inner workings of attention, which is often considered a “black box” feature. Creating visualisations made the model more transparent, showing how tokens relate to one another in context.

I also strengthened my skills working with the Hugging Face library, PyTorch tensor operations, and Python image manipulation. Debugging attention indices and testing different layer/head outputs gave me valuable hands-on insight into both NLP modelling and visualisation programming.

Future Work

There are several ways in which this project could be expanded:

- Interactive Web Interface: Build a simple front-end where users can type sentences, introduce masks, and see both predictions and attention heatmaps directly in the browser.
- Support for Multiple Masks: Extend the programme to handle cases with more than one [MASK] token.
- Coloured Heatmaps: Replace greyscale shading with colour gradients to enhance readability.
- Comparison Between Models: Include alternative transformer architectures (e.g. DistilBERT, RoBERTa) to compare prediction accuracy and attention behaviour.
- Case Studies: Apply the tool to real-world examples such as ambiguous sentences, idioms, or polysemous words, to analyse how BERT interprets them.

Conclusion

This project successfully demonstrates how a pre-trained transformer model can be applied to natural language understanding tasks, specifically masked language prediction. Beyond predictions, the programme provides a way to visualise the internal attention mechanisms that contribute to BERT's contextual reasoning. The combination of machine learning, data visualisation, and user interaction makes this both an analytical tool and an educational resource for understanding transformer models in practice.