NLP Selection Camp: BERT Implementation

**Please note that this is not an actual Inter IIT Tech Problem statement. Solving this bootcamp gives you a significant advantage for the selection in the coming mid prep NLP problem statement in Inter IIT Tech Meet 14.0.*

Implement a simplified version of BERT from scratch using PyTorch or TensorFlow. Train it on a small English corpus using masked language modeling (MLM) and next sentence prediction (NSP). Demonstrate its learned representations by showing masked token predictions and NSP accuracy.

BERT is a general-purpose language representation model that achieved state-of-the-art results on 11 different NLP tasks upon its release . These tasks ranged from question answering to natural language inference, demonstrating the wide applicability of BERT's learned representations. The introduction of BERT "took the NLP community by storm" with its strong empirical performance and has become a staple method in NLP ever since . In fact, BERT's Transformer-based architecture and transfer learning approach have made it a ubiquitous baseline in NLP experiments by 2020 , and it's considered a foundational advancement that many subsequent models (and Inter-IIT Tech problems) build upon.

Bootcamp Tasks

Read the BERT paper thoroughly- https://arxiv.org/abs/1810.04805.

Ensure you understand the main goals of the model, the architecture of BERT, and the training objectives (Masked Language Modeling and Next Sentence Prediction). It may help to also review background materials on the Transformer architecture and prior models like ELMo/GPT for context. Key sections to focus on are the model architecture and the training procedure.

Model Implementation (from scratch): Implement an **encoder-only Transformer** in PyTorch (or similar). Include:

- Multi-layer Transformer encoder with self-attention and feed-forward networks (you can choose a smaller number of layers/hidden size than the full BERT_base to suit time and computing resources).
- Input embeddings that sum token embeddings, position embeddings, and segment embeddings (as in BERT, segment embeddings distinguish sentence A vs B for the NSP task)
- Special tokens [CLS] (classification token at start) and [SEP] (separator token at end of sentence or between sentence pairs).
- Output heads:
    - **MLM head** – predict masked tokens.
    - **NSP head** – binary classifier on [CLS] for next-sentence prediction.

      Use standard layers (e.g. nn.Linear) but **no prebuilt BERT**.

      Tokenization via existing WordPiece libraries is allowed.

Training Procedure: Train jointly on-

1. **Masked Language Modelling:** For each input sequence, randomly mask out a small percentage of tokens (e.g. 15%) and have the model predict the masked tokens' identities. Use the strategy described in the paper: e.g. 80% of the time replace with [MASK] , 10% with a random word, 10% leave unchanged (to bias against the model just learning the mask token.
2. **Next sentence prediction:** Create input sequence pairs for training. Some pairs are positive examples where the second sentence truly follows the first from the original text, and others are negative examples where the second sentence is chosen randomly from the corpus (not actually the next sentence).

   Both losses (MLM and NSP) should be combined (e.g. summed) to train the model jointly, as in the paper . You will need to iterate through the corpus to generate these training examples. Tip: It's often easiest to prepare the corpus as segmented sentences. Take consecutive sentence pairs as positive examples, and for each positive pair create a negative pair by pairing the first sentence with a random second sentence from the corpus.

Dataset

Use **WikiText-2** from HuggingFace (wikitext, wikitext-2-v1). WikiText-2 is much smaller than the full Wikipedia corpus BERT was trained on, but it is sufficient for a one-week exercise and will allow your model to learn basic patterns of English. The dataset comes in a training split (~2 million tokens) and a validation split for evaluation. Use a provided vocab or HuggingFace tokenizer. Segment into sentence pairs for NSP. Mask 15% tokens for MLM as per BERT's masking strategy.

Resources that might help

- **BERT Paper:** BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al., 2018)
- **Transformer Paper:** Attention Is All You Need (Vaswani et al., 2017)
- *The Illustrated Transformer* by Jay Alammar – visual guide to how transformers work. [The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time.]
- *The Illustrated BERT, ELMo, and co.* – intuition and visuals for contextual embeddings. [The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning) – Jay Alammar – Visualizing machine learning one concept at a time.]
- *BERT Explained: State of the art language model for NLP* (Ruder Blog). [BERT Explained: SOTA Language Model For NLP [Updated]]
- [Fine-Tuning BERT: A Practical Guide | by why amit | Medium]

These resources cover background theory, visualization guides, and practical tutorials that will help you understand and implement BERT efficiently.

Submission Guidelines

- Submit only code + README.
- No large model checkpoints required.
- **Submission link (open with LDAP ID)**- https://forms.gle/TBykDdAqxco3LGfD6
- **Deadline: 8th November EOD**.

For any queries, contact- Lakshaditya Singh (96679 48002)

Solving this bootcamp gives you a significant advantage for the selection in the coming mid prep NLP problem statement in Inter IIT Tech Meet 14.0. It's completely fine if anyone is able to do only a sub part of the asked deliverables. We just expect genuine efforts from your end.