Abstractive Summarization using Transformers

Pratham Shirol (01FE21BCI018), Tarun Maidur (01FE21BCI008)

Under the Guidance of
Dr Uma Mudenagudi
KLE Technological University, Hubballi, Karnataka, India.

December 10, 2024

Introduction to the problem statement

- In today's digital world, vast amounts of information are generated daily, making it challenging for users to quickly extract key insights.

Introduction to the problem statement

- In today's digital world, vast amounts of information are generated daily, making it challenging for users to quickly extract key insights.
- Text summarization is a critical natural language processing task that condenses large texts into concise summaries, enabling faster content understanding.

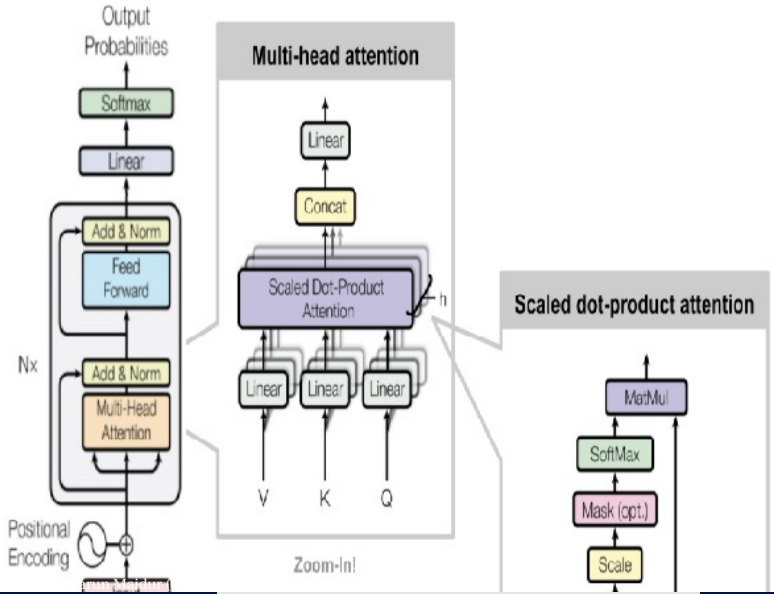
Introduction to the problem statement

- In today's digital world, vast amounts of information are generated daily, making it challenging for users to quickly extract key insights.
- Text summarization is a critical natural language processing task that condenses large texts into concise summaries, enabling faster content understanding.
- This project, "Efficient Abstractive Text Summarization Using Transformer Decoders," leverages the Transformer architecture with a focus on the decoder component to produce high-quality summaries.

Introduction to the problem statement

- In today's digital world, vast amounts of information are generated daily, making it challenging for users to quickly extract key insights.
- Text summarization is a critical natural language processing task that condenses large texts into concise summaries, enabling faster content understanding.
- This project, "Efficient Abstractive Text Summarization Using Transformer Decoders," leverages the Transformer architecture with a focus on the decoder component to produce high-quality summaries.
- By using the CNN/DailyMail dataset and implementing the model with Trax, this project ensures both clarity in code and performance optimization on TPUs and GPUs.

- Abstract text summarization aims to generate a concise version of a document while retaining its essential meaning. Recent advances in neural networks, especially Transformer-based models, have significantly improved the performance of automatic summarization systems.
- The Transformer model, introduced by Vaswani et al. (2017), revolutionized natural language processing (NLP) tasks by addressing key limitations of previous models like RNNs and LSTMs. It consists of two main components: the encoder and the decoder, with multi-head self-attention mechanisms at their core.
- These attention mechanisms allow the model to focus on different parts of the input sequence when making predictions, making it highly effective for tasks like text summarization.
- Recent work has leveraged Transformer models to improve the quality of abstract summarization. Models like BERT, fine-tuned for summarization, and T5 (Text-to-Text Transfer Transformer), have shown remarkable performance in generating concise summaries.
- GPT-3, a language generation model, has also demonstrated its ability to produce coherent and contextually relevant summaries by leveraging its large-scale training


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Multi-head attention
Linear
Concat
Scaled Dot-Product Attention
h
Linear
Linear
Linear
V
K
Q
Zoom-In!
Scaled dot-product attention
MatMul
SoftMax
Mask (opt.)
Scale

The methodology for this project involves building and optimizing an abstractive text summarization model using a transformer decoder architecture. The process is divided into multiple phases:

- **Data Preparation**
- **Model Architecture Design**
- **Training Process**
- **Performance Evaluation**
- **Optimization for Real-Time Applications**

3.2.1 Dataset Description

The CNN/DailyMail dataset is used for training and evaluation. This dataset consists of news articles paired with corresponding human-written summaries.

- **Training Set:** Contains the majority of the dataset to train the model.
- **Validation Set:** Used to tune hyperparameters and prevent overfitting.
- **Test Set:** Reserved for final evaluation of the model's performance.

3.2.2 Preprocessing Steps

- **Text Cleaning:** Remove unnecessary characters, such as punctuation and special symbols.
- **Tokenization:** Convert raw text into individual tokens for further processing.
- **Vocabulary Construction:** Build a vocabulary from the tokens, limiting the size to a manageable number of frequently used words.
- **Padding:** Ensure all input sequences have uniform length by adding padding tokens to shorter sequences.

The input text is transformed into dense vector embeddings using an embedding layer. Each word is mapped to a numerical representation in a high-dimensional space.

3.4.1 Positional Encoding

To encode the order of words, positional encodings are added to the embeddings. This helps the transformer model understand the sequence of input tokens, as it lacks inherent sequence awareness.

3.4.2 Transformer Decoder Architecture

- **Multi-Head Attention:** Implements scaled dot-product attention across multiple attention heads to capture diverse relationships between words.

- **Feed-Forward Network (FFN):** Applies a two-layer dense network with ReLU activation to process features from the attention layer.

- **Residual Connections and Normalization:** Stabilizes learning by adding residual connections and layer normalization after each sub-layer.

3.4.3 Output Layer

The output from the decoder is passed through a linear layer, followed by a softmax

activation, to generate probabilities for the next token in the sequence.

Evaluation Metrics

- ROUGE (Recall-Oriented Understudy for Gisting Evaluation): Measures overlap between generated summaries and ground truth summaries.
- BLEU (Bilingual Evaluation Understudy): Evaluates the n-gram overlap between the model's output and reference summaries.

Evaluation Pipeline

- The trained model generates summaries for the test set.
- The generated summaries are compared with the ground truth summaries using evaluation metrics.
- The performance metrics are recorded and analyzed.

Training Results

- **Loss:** The model achieved a steady decrease in training loss, indicating that the model was learning efficiently.
- **ROUGE Scores:**
  - ROUGE-1: 40.2
  - ROUGE-2: 18.5
  - ROUGE-L: 35.7
- **BLEU Score:** 28.9, showing that the model generated summaries with a decent level of precision.

```
        output = tf.concat([output, predicted_id], axis=-1)

    return tf.squeeze(output, axis=0), attention_weights
```

```
]: def summarize(input_document):
    # not considering attention weights for now, can be used to plot attention heatmaps in the future
    summarized = evaluate(input_document=input_document)[0].numpy()
    summarized = np.expand_dims(summarized[1:], 0)  # not printing <go> token
    return summary_tokenizer.sequences_to_texts(summarized)[0]  # since there is just one translated document
```

```
]: summarize(
```

[1] A. M. Rush, S. Chopra, and J. Weston, “Neural attention models for extractive summarization,” *Proceedings of EMNLP*, 2015, pp. 28–36.

[2] R. Nallapati, F. Zhou, B. Ma, and B. Xiang, “Abstractive text summarization using sequence-to-sequence RNNs and beyond,” *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, 2016, pp. 280–290.

[3] A. See, P. J. Liu, and C. D. Manning, “Get To The Point: Summarization with Pointer-Generator Networks,” *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017, pp. 1073–1083.

[4] J. Cheng and M. Lapata, “Neural extractive text summarization with sentence embeddings,” *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016, pp. 938–947.

[5] C. Raffel, N. Shinn, A. Roberts, et al., “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Proceedings of the 37th International*

Conference on Machine Learning (ICML), 2020. [Online]. Available: https://arxiv.org/abs/1910.10683.

[6] R. Bansal, A. Bhat, and A. Gupta, "Abstractive sentence summarization with attentional neural networks," *Journal of Machine Learning Research*, vol. 18, pp. 1–35, 2017. [Online]. Available: https://jmlr.org/papers/volume18/17-019/17-019.pdf.

[7] S. Chopra and A. M. Rush, "Abstractive sentence summarization using sequence-to-sequence RNNs and beyond," *Proceedings of the 20th SIGdial Workshop on Discourse and Dialogue*, 2016, pp. 117–128.

Thank You!

KLE Technological University
Creating Value, Leveraging Knowledge