EEE 486/586

Statistical Foundations of Natural Language Processing

Assignment 3

Implementing Transformer Architecture for Dialogue Summarization

(Due 27/04/2025, 23:59 PM)

General Instructions

Groups: You are expected to work alone.
TA: Enes Koşar (please mail to enes.kosar@bilkent.edu.tr if you have any questions).

Over the past few years, natural language processing (NLP) has changed a lot thanks to big models like BERT, GPT, and DeepSeek. These changes started with a new design called the transformer architecture, which was introduced in the paper "Attention is All You Need" [3]. This new design helps understand the meaning of words in sentences better by looking at the words around them. It also makes it easier to see how words are connected over long sentences and lets computers process data faster at the same time.

In this assignment, you will create your own transformer architecture and attention mechanism from scratch. You'll use this to summarize dialogues, turning long conversations into short summaries. You are allowed to use existing tools to help prepare your data, but you need to build the transformer part yourself. You are provided a notebook as a guideline [1]. You will follow the guidelines and implement the missing parts in this notebook.

This project will help you learn more about how modern NLP works and give you a chance to build something on your own.

Submission Guidelines:

(i) Write a brief report (max 15 pages) that explains the details of your procedure part-by-part and answer any further questions in the assignment.

(ii) Name the report "report_SurnameNameID.pdf".

(iii) You should also submit your .ipynb notebook and your results should be visible there.

(iv) **Failing to meet these requirements may result in loss of grades. Wrong naming of files and/or submitting to the wrong places in Moodle will also be penalized by grade deductions.**

PART A: Theory and Implementation

For this part, two suggested references are "Attention is All You Need" paper [3] and related chapters of your textbook: "https://web.stanford.edu/ jurafsky/slp3/" [2]. However, you can refer to any other reference as well.

i) Import Dataset and Preprocess Data
Go to: https://github.com/cylnlp/dialogsum and download the dataset. Upload it on your environment in google drive which you have your google colab notebook. Check the dataset and make necessary modifications in code to read and preprocess the data properly. Explain tokenization, positional encoding, and masking (padding mask and look ahead mask) and their functionalities with a few sentences.

ii) Explain, Implement, and Test Scaled Dot Product Attention
You will implement scaled dot product attention which takes in a query, key, value, and a mask as inputs to return rich, attention-based vector representations of the words in your sequence. First, explain the scaled dot product attention mechanism with the help of diagrams or figure plots that illustrate the flow of data and components involved in the process. Then, complete the missing parts in scaled_dot_product_attention(q, k, v, mask) function. Finally, test your code with given values and share your output and code implementation.

iii) Explain, Implement, and Test Encoder Layer and Full Encoder
Explain the encoder architecture with the help of diagrams or figure plots that illustrate the flow of data and components involved in the process. Explain the multi-head attention block separately. Then, fill the missing parts in EncoderLayer and Encoder classes. Test your code with given values and share your output and code implementation.

iv) Explain, Implement, and Test Decoder Layer and Full Decoder
Explain the decoder architecture with the help of diagrams or figure plots that illustrate the flow of data and components involved in the process. Then, fill the missing parts in DecoderLayer and Decoder classes. Test your decoder layer and full decoder code with given values and share your output for the test case. Also, share the part of the code that you implemented.

v) Explain, Implement, and Test Transformer Architecture
Explain the full transformer architecture with the help of diagrams or figure plots that illustrate the flow of data and components involved in the process. Then, fill the missing parts in Transformer class. Test your code with given values and share your output. Also, share the part of the code that you implemented.

PART B: Model Training, Results, and Discussion

i) Model Training
Your model is ready. Now, you need to train it to learn contextual embeddings and text summarization. Fill the missing part in next_word() function, test it with the given values,

and share your results and part of the code you implemented. Then, train your model using the provided code and share the plot of the loss function over the iterations.

ii) Sample Summarization Results and Discussion
Share one sample from training data and one sample from test data. For each sample share your summary and the ground truth summary. Comment on the performance of your model on the training set and test set. Are your results reasonable? Could you get better results? What might be the factors limiting your model's performance?

iii) Calculate your BERTScore on Test and Training Set
There are several methods to calculate similarity between two texts. You will use BERTScore to measure the similarity between your summaries and the ground truth summaries given in the dataset. You can adjust the given code if necessary. Share your average results for training and test sets.

Report Preparation

1. **The deadline for the Final Report is 27/04/2025, 23:59PM.** No late submissions will be allowed.
2. I recommend using LaTeX, though it is not mandatory. If you did not use it before, take this as a chance to get used to that good practice.

Important remarks:

- Collaboration and code sharing among students are prohibited.
- You are allowed to use any libraries and frameworks you want for preprocessing. However, transformer architecture should be your own implementation.
- Properly label all your figures and tables throughout your report.
- Your reports will be evaluated based on the proper completion of tasks, clarity of presentation of results, the sufficiency of discussions regarding the results, quality of writing, plots, and organization of the report, and your possible insights and comments.
- There might be slight deviations in your results depending on your hardware facilities etc.
- Please see the link for information about academic honesty and plagiarism:
- You are expected to submit both your codes and report in a zip folder to Moodle. Please do not include model weights.

References

[1] DeepLearning.AI. Natural language processing specialization, 2020. Accessed: 2025-03-17.

[2] Daniel Jurafsky and James H. Martin. Speech and language processing (3rd ed. draft), 2023. Accessed: 2025-03-17.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.