Large language model for information retrieval

Introduction

1. Need - Text retrieval is one of the main component of digital applications such as web seach, or product recommendation. It is also one of the main component of RAG pipeline which helps in mitigating hallucinations and enhancing accuracy and context of LLMs.
2. Function - Retrieves the top-k relevant texts from text corpus.
3. Another component is reranker. Which reranks the items retrieved by the retriever. Different in architecture - more computational expensive (only works with a subset of data)

In this we will be talking mainly about retriever.

The inspiration is mainly from the paper Fine Tuning Llama for multistage text retrieval.

Type of retrievers (based on transformers) -

LLMs are used to encode the query and the documents into vector representation, which can then be compared with each other through metrics like cosine similarity to calculate similarity.

1. Pompting fine tuned LLMs such as InstructGPT. This method sees retrieval or reranking as text generation task. (see Uncovering ChatGPT’s Capabilities in Recommender Systems)
2. Fine tuning pretrained Decoder only LLMs
3. Fine tuning pretrained Encoder only LLMs such as BERT. (in this project)

Diagram showing components of retriever -

Anchor > BERT > encoded anchor

> contrastive learning (using some distance metric)

Positives/Negatives > BERT > encoded text

In this project -

Objective - experimenting with the pipeline to see how retrieval is affected.

Approaches -

1. Forming a baseline model - Embeddings - CLS token of BERT, distancemetric used - cosine distance, loss function used - infoNCE loss.
2. Hyperbolic model - Embeddings - CLS token of BERT but projected to hyperbolic space, distance metric used - lorrentz distance, loss function used - a combination of entailment and infoNCE loss.
3. Average embedding model - Embeddings - Average of embeddings of all the tokens from last hidden layer of BERT, distance mertic - same as base model, loss function - same as base model
4. Effective model - Embeddings - same as basemodel, distance metric - effective distance, loss function - same as base model

Architecture -

Embedding model - bert-base-uncased (LoRA)

Dataset -MS-macro passage dataset

Number of negatives for each anchor - 5

Batch size - 32

Total epochs - 3

Learning rate -

InfoNCE loss -

Equation

Code implementation

Hyperbolic model

Paper -

The authors propose that lingustic concepts organize themselves in a hierarchy and giving this detail to the model can help it better understand and thus retieve the data. Example - A same sentiment can be described as a cat and dog playing in the street or an exhausted dog or “so cute”.

They proposed MERU model to come up with hyperbolic embeddings.

They used it for image-text retrieval. Applying it for text-text retrieval.

Figure showing the pipeline

Figure showing entailment loss

Equation - loss=contrastive loss+alpha\*entailment loss

Average embeddings model -

Paper -

Methodology - instead of using only the CLS token for representation of text, using average of embeddings of all the tokens in final hidden layer.

Reason - one token might not be sufficient enough to account for the whole text, hence using of more tokens should yield better representation, hence retrieval.

Effective model -