Project Proposal

Long Document Summarization: Augmenting Unlimiformer with Knowledge Graphs[1]

Patrick O'Callaghan (patocal)   Sheel Sansare (ssansa2)   Tristan Wang (aawang99)

Introduction

Augmenting large-language models to allow them to handle large documents using retrieval-based methods is currently a highly active area of research. Many real-world documents are long, with the typical government report being around 10,000 tokens and the average novel being well over 100,000 tokens. The key weakness of modern transformer-based LLMs is the token limit or context window of attention. For instance, the context window of ChatGPT is limited to 4096 tokens.

In this project, we wish to understand how knowledge graphs can improve a state-of-the-art approach to augmenting transformers. The task we perform is the summarization of long documents. Our baseline model is `unlimiformer` [Ber+23], a recent retrieval-based method for augmenting LLMs at the decoder level. The key innovations of `unlimiformer` are (i) to create a datastore of encodings: one for each token in the original document and (ii) using the k-nearest-neighbor algorithm for selecting the k most relevant tokens in the datastore during decoding. We will enrich their datastore with a knowledge graph and replace k-NN with graph-based notions of closeness and topology. We will use the GovReport and BookSum datasets, and evaluate performance using the usual ROUGE and BERTScore families of metrics.

Dataset

We choose to use the Hugging Face versions of the GovReport [Hua+21] and BookSum [Kry+21] (fullbook) datasets because they are well-established long-document summarization datasets that are both publicly available and ready-to-use. The `unlimiformer` paper also works with these datasets and our first task is to replicate their results on these two datasets. Our main focus is GovReport because it is the larger dataset and it has many real-world applications. We also choose BookSum not only because it contains longer documents, but also because it is easy to subjectively judge the quality of a summary for books that we have read before. The Hugging Face GovReport dataset has an approximate 90/5/5% split of approximately 19.5k document-summary pairs. The full-book BookSum dataset has an approximate 80/10/10% split of just over 400 document-summary pairs.

[1] October 26, 2023

Metrics

We will use ROUGE-1 (unigram), ROUGE-2 (bigram), ROUGE-L (sub-sequence), and BERTScore. The ROUGE metrics are a standard way of comparing summarization performance through lexical overlap between the model-generated and gold summaries. Similarly, the BERTScore is a standard way to compare the semantic similarity between the model-generated and gold summaries by comparing BERT embeddings of both summaries.

Baseline `unlimiformer` Model

Since Vaswani et al 2017, transformers have become the default approach to natural language processing. Transformers have succeeded due to their ability to capture long range dependencies between tokens. They do so by abandoning the sequential approach of recurrent neural networks and instead allowing the decoder to attend to a complete graph over the encoded hidden states. However, the complexity of complete graphs is quadratic in the number of tokens and this explains why LLMs have relatively small context windows.

To bypass this constraint, numerous creative approaches have been proposed and most involve breaking the document into chunks of size k, where k is equal to the context window length. Our baseline model `unlimiformer` (see figure 1) bypasses this constraint by changing the contents of the context window. That is, instead of passing the next chunk of text in the sequence, it feeds the decoder the k-nearest neighbors that it can find in a datastore that contains all the tokens in the entire document. Consider a simplified equation of attention

$$\text{Attn}(Q, K, V) = \text{softmax}(QK^T)V,$$

where, for each hidden-layer state h_d of the decoder and the final-hidden layer state h_e of the encoder, $Q = h_d W_Q$, $K = h_e W_K$ and $V = h_e W_v$.[2] The trick is to rewrite

$$(QK^T)_{i,j} = \langle p_{d,i}, h_{e,j} \rangle$$

where $p_{d,i} = h_{d,i} W_Q W_K^T$. This allows us to create a datastore $\{h_{e,j} \in \mathcal{H}_{\text{enc}} : j \in \text{LongDoc}\}$ and identify the k nearest neighbors in the datastore to the projection p_d. Only those k nearest neighbors are passed to the decoder of an otherwise standard LLM.

Augmenting `unlimiformer` with Knowledge Graphs

Our goal is to enrich `unlimiformer` approach with a knowledge graph (see figure 1). To do so, we will use standard entity-extraction techniques to identify key entities in each document (as in [Wu+20]). Since KGs store richer information than a plain datastores, we

[2] In the context of translation, think of h_d as the hidden state of tokens in "I am a student" and h_e as the hidden state of tokens in "Je suis étudiant".

aim to show that they enable us to generate more accurate and coherent summaries. We draw inspiration from [Wan+22] in the related task of multi-document summarization.

Our second step will be to identify an isomorphism between entities E in the KG and sets $H_{e,E}$ of top-level encodings of tokens (similar to [Gal+21]). For example, if the literary character $E =$ `Karamazov`, then our entity embedding is the set $H_{e,E} = \{h_{\texttt{Kar}}, h_{\texttt{amaz}}, h_{\texttt{ov}}\}$ in $\mathcal{H}_{\text{enc}}$. This set of tokens would form a clique in our encoding of the KG. Relations or edges may also be encoded as sequences of tokens that connect entities. We recognize that there are still some modelling choices to be explored in this respect. Yet this perspective of cliques as entities already brings to mind the `struct2vec` notion of structural similarity.

Introducing KGs will thus allow us to employ more refined notions of k-"nearest". In terms of our earlier mathematical discourse, `unlimiformer` defines "nearest" according to the topology generated by the linear functional $\langle p_{d,j}, \cdot \rangle : \mathcal{H}_{\text{enc}} \to \mathbb{R}$ for each token j in the target sequence. This is the *weak topology* over the hidden state space. We will explore more graph-based topologies such as structural similarity (in the spirit of `struct2vec` [RSF17]) which may be more appropriate for long document summarization.

A Diagram of the combined model

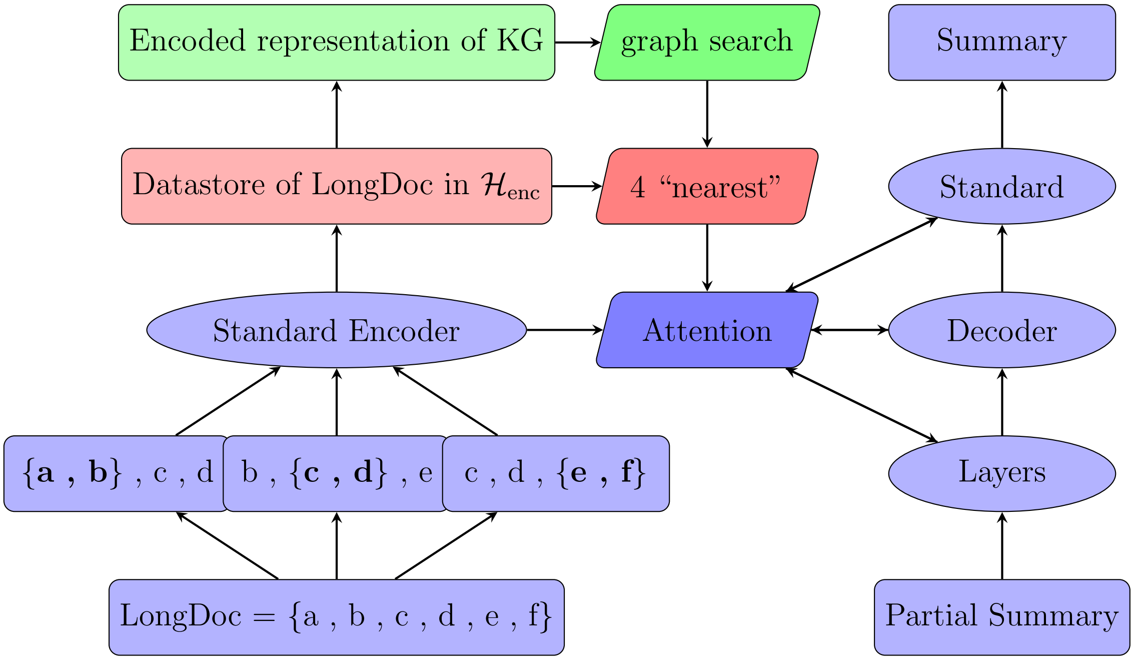

Figure 1: A stylised transformer with length-4 context window (in blue). The standard approach to handling long documents is to process consecutive overlapping chunks of tokens sequentially. In red, `unlimiformer` augments this process by creating a complete datastore and selecting which tokens are fed into the attention mechanism. We augment this process further using a Knowledge Graph to aid selection.

References

[RSF17] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. “struc2vec: Learning node representations from structural identity”. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017, pp. 385–394.

[Wu+20] Zeqiu Wu et al. “Extracting summary knowledge graphs from long documents”. In: *https://arxiv.org/pdf/2009.09162.pdf* (2020).

[Gal+21] Mikhail Galkin et al. “NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs”. In: *CoRR* abs/2106.12144 (2021). arXiv: 2106.12144. URL: https://arxiv.org/abs/2106.12144.

[Hua+21] Luyang Huang et al. “Efficient Attentions for Long Document Summarization”. In: *https://arxiv.org/pdf/2104.02112.pdf* (2021).

[Kry+21] Wojciech Kryściński et al. “BookSum: A Collection of Datasets for Long-form Narrative Summarization”. In: *https://arxiv.org/abs/2105.08209* (2021).

[Wan+22] Pancheng Wang et al. “Multi-Document Scientific Summarization from a Knowledge Graph-Centric View”. In: *Proceedings of the 29th International Conference on Computational Linguistics*. 2022, pp. 6222–6233.

[Ber+23] Amanda Bertsch et al. “Unlimiformer: Long-Range Transformers with Unlimited Length Input”. In: *https://arxiv.org/pdf/2305.01625v1.pdf* (2023).