Transformers are limited by their input size due to O(n^2) computations. Furthermore, they perceive the whole input at the same time, which forces them to have a fixed maximum size.

In literature we have identified three main approaches to this problem. First is to make the inputs smaller by selecting the most relevant parts (text summarization + retrieval). Second is to modify the architecture of the transformer so that it can accommodate larger contexts (e.g. longformer, kernelization of the attention computations). Third approach is to modify existing architectures so that they support recurrent inputs i.e. resemble RNNs (<https://arxiv.org/abs/2305.13048>).

The first approach is clearly less scalable as the input size increases and provides a temporary remedy rather than ultimate solution. However, it is less expensive as it enables smaller models to process long inputs and at the moment performs better than transformers with long contexts (looking at our baselines and the leaderboard).

The second approach is more expensive (in terms of model size and computing power), e.g. gpt 4 (32,768 tokens), or claude (<https://www.anthropic.com/index/introducing-claude>, 100k tokens). Although the contexts for these models are significant they still have fixed input size. Thus, at some point certain tokens will be forgotten. Thus, at that point these models will have to either implement some kind of permanent memory or retrieve the most important parts of the “forgotten” tokens.

Third approach is promising in terms of scalability, however current models perform poorer compared to pure transformer-based models. One prominent work with competitive results is this one <https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00547/115346>, authors use sliding window to pass information from one chunk to the next.

We have decided to focus on the first approach for two reasons. To test to what extent can pure retrieval be competitive to large context models. Secondly, because as long as transformers will have fixed input size at some point extraction of the most relevant parts of the old text will be needed – something that information retrieval and text summarization are perfect for.

**Basic retrieval:**

* Random sentences,
* Start-end prior – first few sentences are often a good summary of a text (find citation)
* TF-IDF

**Models:**

* Roberta:
  + Pretrained on Race
  + (Option) Pretrained on Quality
* Longformer:
  + Pretrained on Race
  + (Option) Pretrained on Quality
* (Option) Deberta

**Methods:**

* Sentence similarity
* Ranker
* Drop assumption that chunks are independent:
  + Bayesian optimization for text summarization of long texts (<https://www.sciencedirect.com/science/article/abs/pii/S0950705123000862>)
  + Conditional text summarization (e.g. try Bart) <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7757121/>, <https://arxiv.org/pdf/2212.10423.pdf>
  + Beam search instead of greedy

**Question 1:** how to score relevance of the span of text to the question, which is needed for the above methods

**Question 2:** how to divide text into chunks? Sentences, fixed lenght

**Preprocessing:**

* Remove stop words,
* Named entity mapping