Usefulness of sentence embeddings for information retrieval in long contexts

Mihailo Grbić
grbicm

Jakub Łucki
jlucki

William Wong
wiwong

Abstract

Our project investigates the usage of sentence embeddings to address the limitations of transformer-based language models (LMs) in processing longer texts due to their fixed input size. In particular, in the context of solving reading questions that involve longer articles, e.g., more than 5000 tokens, smaller language models face the challenge of truncating the text, which can lead to a loss of important information needed to accurately answer the question. To address this problem, we explore several retrieval methods that are capable of identifying relevant chunks within an article. By identifying these chunks, only a small subset of the text needs to be fed into the language model. This study looks at the use of sentence embeddings and compares their performance to simple retrieval methods such as TF-IDF, random selection, and selection with preference for the beginning and end of a long text. By investigating retrieval methods, the project aims to improve the effectiveness of transformer-based LMs in dealing with longer texts and to provide more accurate answers to reading questions.

1 Introduction

Transformer-based language models achieve impressive performance in a variety of language-related tasks such as translation, summarizing, and question-answering. However, NLP benchmarks have primarily focused on short texts, although a large portion of natural language is produced in the context of longer discourses such as books, movies, or articles. The main problem with using transformer-based LMs to process longer texts is the fixed input size of the transformer. While a lot of the recent improvements in NLP have been achieved by simply increasing its input size (Radford et al., 2019; Brown et al., 2020), this approach is not scalable since the complexity of the self-attention operation grows quadratically with sequence length. One popular approach in enabling transformer-based LMs to handle long text sequences has focused on altering the transformer architecture so that its complexity scales linearly with the sequence length, which enabled even bigger increases in the max input size of the model (Beltagy et al., 2020; Zaheer et al., 2020; Dai et al., 2019). However, even with such specific architectures, there will always be a limit to the maximum context size of the model. One adjacent approach that is much simpler and straightforward is to perform extractive summarization of the input text, keeping only parts of the text that are the most relevant to the task at hand. One of the main advantages of this approach is that it can be added to any existing NLP model, without any need for fine-tuning, enabling it to process larger input sizes.

2 Related Work

Most of the prior work for enabling LMs to work with longer texts could be broadly separated into two categories.

Retrieval-augmented language models Approaches from this category augment the input to the model using information extracted from an external datastore. The information in the datastore is usually stored as chunks and the relevant information is extracted by calculating a similarity measure between every chunk and the prompt. There are many methods for calculating this similarity measure such as using a simple ROUGE-1 score (Lin, 2004) or generating chunk embeddings using token-level models such as FastTest (Bojanowski et al., 2017), or specialized sentance-level models (Karpukhin et al., 2020). After each chunk is scored the top chunks are selected and used to augment the model input, either by using a specialized model architecture (Zhou et al., 2022) or by just simply appending the retrieved information to

model input (Shi et al., 2023; Ram et al., 2023).

Long-context transformers The methods under this category modify the standard transformer architecture to allow the model to work with longer inputs. Beltagy et al. (2020); Zaheer et al. (2020) increase input size by modifying the attention mechanism, so that its complexity is linear with sequence length, instead of quadratic. Bulatov et al. (2022) on the other hand uses standard pre-trained language models recurrently, by specifying part of the input and output tokens as memory.

3 Datasets

3.1 QuALITY

We utilize the QuALITY dataset (Pang et al., 2021), which includes a set of questions accompanied by articles, and four multiple-choice options. The articles consist of Project Gutenberg fiction stories, Slate magazine articles from Open American National Corpus, and other nonfiction articles from Long+Short, Freesouls, and the book Open Access. The questions were formulated by writers who have professional experience in literature and teaching, who were were specifically encouraged to write unambiguous and answerable questions (Pang et al., 2021). An example question can be seen in Figure 1.

Input

Story (6405 tokens, approx. 30 min to read)

Which is the best representation of Dr. Lessing's worries?

A. He is anxious about the amount of time it will take to revise
B. He is concerned that having to back up his claims could keep him from being objective
C. He is having second thoughts about his qualifications to publish a volume like this
D. He is not sure how he will be able to publish the facts without including the confusing information about the boy

Output

B. He is concerned that having to back up his claims could keep him from being objective

Figure 1: Example of a question from the QuALITY dataset (Pang et al., 2021).

3.1.1 Length

The average length of an article is 5,159 tokens, taking approximately 30 minutes to read, and is significantly longer than typical current transformer models can process (Pang et al., 2021). The average question length is 12.5 tokens and the average option length is 11.2 tokens. The maximum length of an article, question, and option is 7,759 tokens, 103 tokens, and 75 tokens, respectively (Pang et al., 2021).

3.2 RACE

RACE is a reading comprehension dataset consisting of near 28,000 passages and nearly 100,000 questions created by English instructors. The passages and questions were collected from English exams targeted for middle and highschool students (Lai et al., 2017). Similar to QuALITY, it features multiple-choice questions with four answer options, wherein only one option is correct. However, the passages are much shorter, averaging around 351 words (Lai et al., 2017).

4 Methods

As our baseline, we selected three simple information retrieval methods that transform each article into a smaller form that fits the maximum input size of the multiple-choice question answering model:

1. Random selection: This method randomly selects sentences of the passage while preserving the original order of the sentences in the article.

2. Start+End selection: The chunks at the beginning and at the end are more likely to be selected according to the following probability mass function:

$$p(x) = \frac{0.1}{Z} \times \left(x - \frac{N}{2}\right)^2 + 1,$$

where N is the number of chunks in the article and Z is the normalization constant. This positive convex quadratic function places minimum on the middle chunk. Once the chunks are sampled, original order is restored. We decided to use this heuristic as a baseline, because Liu et al. (2023) found that language models perform best when relevant information is near the beginning or the end.

3. TF-IDF: Each sentence is treated as a document, and the TF-IDF representation of each sentence is calculated. Sentences with higher cosine similarty to the question are then selected.

Long context models: We experiment with a variant of the Longformer model (Beltagy et al., 2020) with a maximum input length of 4096 tokens. This model is a component of the

MQAG (Multiple-choice Question Answering and Generation framework) (Manakul et al., 2023) and was pre-trained on the RACE dataset (Lai et al., 2017).

Short context models: We also explore RoBERTa (Liu et al., 2019), a variant of the BERT model designed for short-context language understanding tasks with a maximum input size of 512 tokens. RoBERTa utilizes a transformer-based architecture and improves upon BERT's performance. We employ a RoBERTa model which has been pre-trained on the RACE dataset.

4.1 Baseline performance

Table 1 shows the performance of our baseline models on the validation set of QuALITY where the chunk size is one sentence.

Model	Random	Start+End	TF-IDF
RoBERTa	0.4573	0.4789	0.5211
Longformer	0.3960	0.3830	0.3854
DeBERTa	0.4770	0.4693	**0.5278**

Table 1: Accuracy on the QuALITY validation set with single sentence chunks. Best result indicated in bold.

4.2 Chunking

We observed that papers do not offer any heuristics nor guidance on how to split articles into chunks to perform information retrieval. As a consequence, we decided to investigate it ourselves. We used the following chunk types:

- *Length limited*. We splitted text into sentences. Then we added following sentences until we reached limit in terms of tokens. Individual sentences exceeding the limit form their own chunks. We used whole sentences instead of cutting them in the middle where necessary, because sentences in theory should present a single coherent idea. Hence, splitting individual sentences into parts could lose the information. We used chunks of sizes: 32, 64, 128, 200. We were interested in observing whether larger chunks consisting of several neighbouring sentences can provide more information as consequent sentences can be coreferent.

- *Sentence*. Each sentence is a separate chunk. Reasoning is similar to the one above. Each sentence should contain one idea.

- *Sentence clusters*. After running the main experiment discussed in Section 5, we noticed that chunk size significantly influences the performance of the models. Hence, we decided to introduce a dynamic chunk size based on sentence clusters.

  In this method, we split each article into sentences. Then we encode each sentence using Sentence-BERT discussed in Section 4.4. Next, we perform agglomerative clustering on the sentences. To do that, we compute the matrix of cosine similarities between all sentences in the article and inverted the values from the $[0, 1]$ to $[1, 0]$ scale. Therefore, we changed similarity measure to a distance, which is the base of agglomerative clustering.

  Furthermore, the chunks have to consist of consecutive sentences, thus, we only kept the similarities (now distances) of the directly neighbouring sentences and set all the other distances between different sentences to maximum value of 1. Using single linkage ensures that only consecutive sentences can form a cluster.

  Unfortunately, agglomerative cannot find the number of clusters on its own. We tried Bayesian mixture models as an alternative clustering method without this limitation but we found it to be too computationally expensive. Therefore, we decided to set the number of clusters to the number of tokens in the article divided by 64 and rounded. We decided to use 64, because chunks of size up to 64 performed best on the downstream task.

4.3 DeBERTa

As an additional model we decided to use DeBERTa (He et al., 2021), which improves upon RoBERTa-Large on various downstream tasks and provides the strongest baseline introduced by the authors of QuALITY (Pang et al., 2021). It is a short-context method with context window of 512 tokens.

We could not find a large version of DeBERTa (1.5B parameters) pretrained on RACE dataset. However, we found a model which was pretrained on 520 different NLP downstream tasks (Sileo, 2023). Authors trained the model by using a common language model on top of which there was an adapter trained for each task. We hypothesized

that such a model should have strong generalization capabilities and performance. However, after further several trial runs, we noticed that the model is under-performing on multiple-choice question tasks.

Therefore, we decided to further finetune the model on RACE dataset. However, due to the lack of computational power it was not possible to finetune it on the whole dataset. Hence, we randomly chose 10k samples from the "high" part of the dataset. We finetuned it for 3 epochs using stepsize of 10^{-5}, weight decay of 0.001 and batch size of 2. The training loss, validation loss and acccuracy (computed using regular validation split) during training can be seen in the Figure 2. We have decided to use the model with the highest validation accuracy, which in our case was the one at the second epoch.

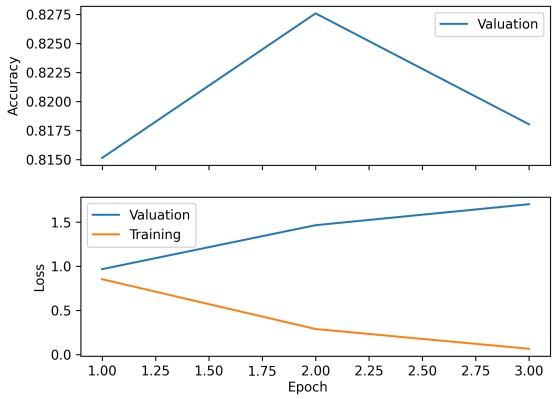


Figure 2: Finetuning of Deberta on $10k$ samples from RACE dataset.

4.4 Sentence-BERT

While BERT (Devlin et al., 2019) enables semantic search by computing representations for two sentences, there is computation overhead. For instance, finding the most similar pair in a collection of 10,000 sentences with BERT requires about 50 million inference computations (∼65 hours) (Reimers and Gurevych, 2019). To alleviate this issue, (Reimers and Gurevych, 2019) have developed SBERT which is a modification of the BERT network to derive fixed-length sentence embeddings. Cosine similarity can then be used to find the most similar pair using the embeddings. The complexity for finding the most similar sentence pair is reduced from 65 hours with BERT to computing 10,000 sentence embeddings (∼5 seconds) and computing cosine similarity (∼0.1 seconds) (Reimers and Gurevych, 2019).

As illustrated in Figure 3, we first split the article in chunks. Then, we derive the embeddings of all chunks and of the question using a pre-trained SBERT model, called "multi-qa-mpnet-base-dot-v1", which was fine-tuned on the semantic search task. Since the model is limited by 512 word pieces, meaning text longer than that will be truncated, we are only experimenting with different chunk sizes up to 200 words. Then we use cosine similarity to select the top-k chunks with the highest simlarity score where k is the maximum possible number of chunks the multiple-choice question answering model can process as an input.

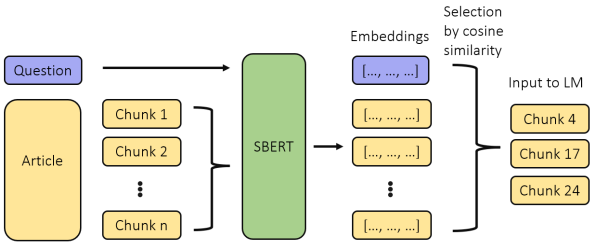


Figure 3: Pipeline of information retrieval

5 Experiments

Our experiments involved testing various chunking sizes and retrieval methods, including both simple retrieval methods and the use of sentence embeddings. We conducted the experiments for the three defined models to determine the optimal one. To assess performance, we have used the validation set of QuALITY. The results of the retrieval accuracy for RoBERTa, Longformer, and DeBERTa can be seen in Table 2, Table 3 and Table 4, respectively. We underline the highest accuracy for each retrieval method and highlight the highest accuracy overall in bold.

RoBERTa					
Retrieval method	**Chunk size**				
	Sent.	**32**	**64**	**128**	**200**
Random	0.457	0.458	0.449	0.464	0.466
Start+End	0.478	0.471	0.468	0.463	0.457
TF-IDF	0.521	0.527	0.519	0.508	0.507
Sent. Emb.	0.539	0.531	**0.542**	0.538	0.496

Table 2: Retrieval accuracy for RoBERTa

The best performing model with fixed chunk size is DeBERTa with sentence embedding retrieval and a chunk size of 64, achieving an accuracy of 0.5503. Conversely, the Longformer model underperforms in every category. This is interesting given the fact

Longformer					
Retrieval method	**Chunk size**				
	Sent.	**32**	**64**	**128**	**200**
Random	0.396	0.404	0.388	0.388	0.402
Start+End	0.383	0.385	0.403	0.400	0.387
TF-IDF	0.385	0.402	0.387	**0.410**	0.394
Sent. Emb.	0.396	0.386	0.408	0.402	0.391

Table 3: Retrieval accuracy for Longformer

DeBERTa					
Retrieval method	**Chunk size**				
	Sent.	**32**	**64**	**128**	**200**
Random	0.477	0.468	0.467	0.459	0.454
Start+End	0.469	0.471	0.456	0.454	0.450
TF-IDF	0.527	0.524	0.524	0.515	0.504
Sent. Emb.	0.537	0.544	**0.550**	0.538	0.510

Table 4: Retrieval accuracy for DeBERTa

that some articles have less than 4096 tokens which is the maximum input size of the Longformer.

Since the chunk size significantly influences the performance, we conduced additional experiments using dynamic chunk sizes with the two better models RoBERTa and DeBERTa, for each retrieval method. The accuracies for dynamic chunk size can be seen in Table 5.

Retrieval method	**DeBERTa**	**RoBERTa**
Random	0.466	0.453
Start+End	0.453	0.481
TF-IDF	0.515	0.518
Sent. Emb.	**0.542**	0.535

Table 5: Retrieval Accuracy with dynamic chunk size for DeBERTa and RoBERTa

The code to reproduce the experiments is available on GitHub. [1]

6 Analysis

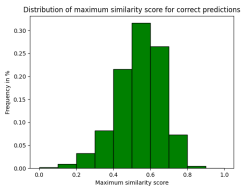
The experiments have revealed that for RoBERTa, the optimal chunk size for sentence embedding is 64, resulting in an accuracy of 0.542. This represents a 2% improvement over TF-IDF in accuracy, 7% improvement over Start+End selection and a 9% improvement over random selection with the baseline chunking.

For Longformer, the optimal chunk size for sentence embeddings is also 64, yielding an accuracy of 0.408. This shows a 2% improvement over TF-IDF, 2% improvement over Start+End selection and a 1% improvement over random selection with baseline chunking.

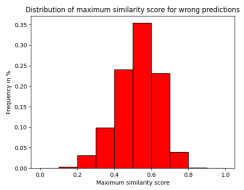
For DeBERTa, the best chunk size for sentence embeddings is also 64, resulting in an accuracy of 0.55. This demonstrates a 2% improvement over TF-IDF, 8% improvement over Start+End selection, and a 7% over random selection with the baseline chunking.

Furthermore, the experiments have shown that fixed chunk sizes outperform dynamic chunk sizes with sentence clustering in this task. The best retrieval method, sentence embeddings, achieves an accuracy of 0.542 for DeBERTa and 0.535 for RoBERTa. These values are 1% worse than fixed size of 64 for DeBERTa and RoBERTa respectively.

In addition, we conducted an analysis on the predictions made by the best model, which was DeBERTa. The results of this analysis are presentented in the following sections.



(a) Distribution of the maximum similarity score for correct predictions



(b) Distribution of the maximum similarity score for wrong predictions

Figure 4: Distribution of maximum similarity scores for each input

6.1 Similarity score as a metric

We investigated whether an input which contains a chunk with high similarity to the question leads to a more accurate answer. To verify this, we analyzed the frequency distribution of the maximum similarity score for correctly predicted questions. Similarly, we performed the same analysis for falsely predicted questions and compare the results. The corresponding plots are presented in Figure 4a and Figure 4b.

The findings indicate that for correctly predicted questions, it is more common to have chunks with a maximum similarity score over 0.6 (around 34%) compared to false predictions (around 27%). This notion is further supported by Table 2, which demonstrates the accuracy based on the maximum similarity score. The table reveals that as the maximum similarity score increases, the likelihood of the model correctly predicting the answer also in-

[1] https://github.com/J4Q8/CS4NLP

creases.

Range	#Correct	#Total	Accuracy
$[0.1, 0.2)$	40	62	0.645
$[0.2, 0.3)$	109	209	0.522
$[0.3, 0.4)$	328	620	0.529
$[0.4, 0.5)$	631	1198	0.527
$[0.5, 0.6)$	690	1243	0.555
$[0.6, 0.7)$	391	645	0.606
$[0.7, 0.8)$	89	127	0.701
$[0.8, 0.9)$	5	6	0.833

Table 6: Accuracy by maximum similarity score

Interestingly, as seen as in Table 6, questions and input chunks with low similarity scores such as in range $[0.1, 0.2)$ are also answered accurately by the model. Further investigation revealed that low similarity scores were associated with broad and general questions that do not require specific information from the article. Examples of such questions include "What is the author's thesis?", "Do you think this story has a happy ending?" and "What is the overall tone of this article?". In these cases, the model does not necessarily require a specific paragraph of text as input and can generate accurate answers based on chunks with low similarity scores.

6.2 Question types

Another hypothesis is that the LM performance is influenced by the question type, meaning that certain question types are more difficult to answer than others.

Type	Mean question length	Accuracy
According	16.27	0.864
Why	10.10	0.578
How	9.95	0.557
What	10.49	0.533
Which	11.56	0.527
Where	7.93	0.515
Who	8.12	0.431
When	9.23	0.353

Table 7: Accuracy and mean question length by question type

As shown as in Table 7, accuracy is highest for questions that begin with the word "according". This result can be attributed to the inherent nature of these questions, which consistently provide more contextual information. It is noteworthy that the average question length for questions with the word "according to" is significantly longer which improves the ability to identify relevant chunks.

6.3 Number of entities

We also investigated whether the number of entities influences the predictive performance of the model. To determine this, we conducted named entity recognition (NER) using the 'en_core_web_sm' model from spaCy (Explosion, 2023) on the input for the language model and compare it between correctly predicted questions and falsely predicted questions. Upon analysis, we observed that the mean, variance and median of the number of entities were similar between correct and false predictions, as shown in Table 8.

Prediction category	Mean	Var	Median
Correct	17.61	70.25	16.0
False	16.80	55.23	16.0

Table 8: Mean, variance and median of number of entities by correct and false predictions

These findings suggest that the number of entities cannot be considered a reliable indicator of predictive performance.

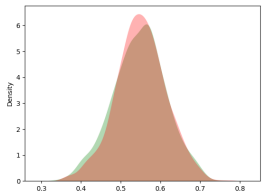
6.4 Importance of diversity of selected chunks

Furthermore, we decided to check how the diversity of the information in the selected chunks influences the performance of the model. To do that we came up with two measures of accuracy, which are described in the following subsections.

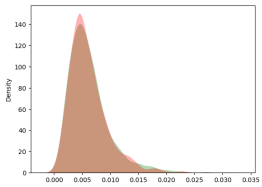
6.4.1 Cosine similarity between selected chunks

We have computed cosine similarity among all unique pairs of different chunks among the selected ones. Then we have computed the mean similarity and variance for each set of selected chunks. Then we splitted them into two sets, one of correct predictions and one of incorrect predictions. Next we plotted the density graphs shown in Figure 5.

Although, in case of both means and variances densities for correct and incorrect predictions. However, there are small differences which suggest that greater diversity in selected chunks contribute to correct predictions to some extent. For example, in terms of means, the green curve is higher than the red one for lower values. This suggests that for small mean similarity of selected chunks there are more correct predictions. In case of variance



(a) Density plot of means of cosine similarity between selected chunks. Green contains means from correct predictions, red contains means from incorrect predictions.



(b) Density plot of variance of cosine similarity between selected chunks. Green contains means from correct predictions, red contains means from incorrect predictions.

Figure 5: Density plots of means and variances of cosine similarities between selected chunks

we can see that for larger variances green dominates the red color, which also suggests that greater variance in similarity contributes to more correct predictions.

6.4.2 Semantic clusters

As a second way of quantifying diversity, we decided to cluster the sentence embeddings of all chunks in the article. We did it using traditional agglomerative clustering (however, we again changed similarity measure to distance measure). As a desired number of clusters we chose the number of tokens in the article divided by 256 and rounded. This was chosen arbitrarily to keep balance between number of clusters and the coherence of each cluster.

Then, we computed the proportion of clusters in which there is at least one representative of selected chunks. We call this the total diversity ratio. Then we plot the densities of total diversity ratios for correct (green) and incorrect predictions (red). The result is Figure 6.

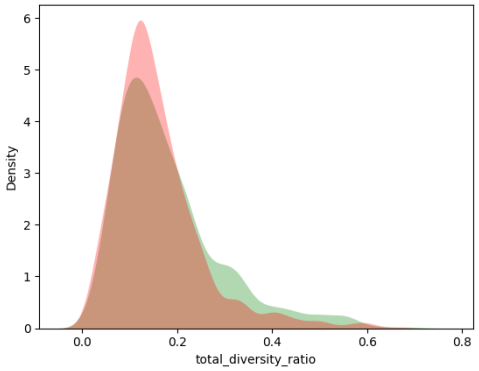


Figure 6: Density plots of total diversity ratio of correct (green) and incorrect (red) predictions.

6.5 Lexical overlap

As an additional experiment, we decided to check how lexical overlap influences the cosine similarity between sentence embeddings of selected chunks and questions to the article. We defined lexical overlap as the proportion of the words in the question that are present the selected chunks. Both selected chunks and question were lemmatized beforehands using default NLTK lemmatizer.

We run logistic regression, using lexical overlap as a predictor and correct prediction as an outcome.

6.6 Human evaluation

We conducted a human evaluation to assess the potential of the performance of the short context models. To achieve this, each member independently selected ten samples from the validation set and attempted to answer the questions using only the chunks selected using cosine similarity. On average, the members achieved an accuracy of 0.65 which is comparable to the performance of the language model on these samples (0.6).

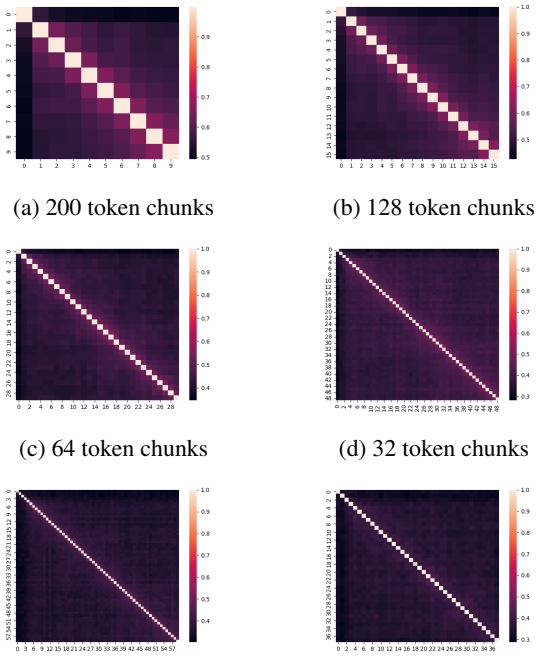
This suggests that extractive summarisation using solely semantic similarity does not provide all the necessary information to successfully solve the task. This indicates that the current moderate performance on the downstream task cannot be attributed to the short context model but primarily to the information retrieval method. Furthermore, we observed that the questions which were impossible to answer based on selected chunks were ones where the relevant information spanned whole article. An example would be ”What is the attitude of one character towards the other?”.

6.6.1 Can we predict if a question is answerable using seleced chunks?

To answer this question, we trained logistic regressor on all features mentioned in above analyses on 80% of the validation dataset. Then we measured accuracy on the remaining 20%. We obtained accuracy of $\sim 54\%$. This indicates that problem of deciding whether a language model can answer correctly given a set of selected chunks is as difficult as answering itself.

6.7 Relative position of chunks

We were also interested in how different chunks sizes relate to their neighbours in terms of cosine similarity between their respective embeddings. Therefore, for each chunk size we have computed



(a) 200 token chunks (b) 128 token chunks

(c) 64 token chunks (d) 32 token chunks

(e) Sentence chunks (f) Sentence cluster chunks

Figure 7: The cosine similarity between chunks at different positions. 0 corresponds to the first chunk in the article.

cosine similarity between first x chunks in all articles and averaged them. Here, x refers to the number of chunks in the shortest article in the dataset. This ensures that when taking the average all positions are represented evenly. The results are presented in the Figure 7.

We can make 2 main observations. First, across all six chunk sizes the first or few first positions are very dissimilar to all other positions in the article. This is indicated by very dark left and top borders across all heatmaps. Potential explanation of this is that the first paragraph of the text can oftentimes be a note from the author or editor, or a summary of the whole passage, or description of the setting. All of those are quite different from the rest of the text, hence the dissimilarity.

Secondly, in the length defined chunks we can notice that the chunks are more similar to their neighbours, whereas sentence and sentence cluster chunks are more distinct from their neighbours (denoted by darker colours in the matrix). The reason for that is that sentence clusters were designed explicitly so that the similar neighbouring chunks belong to one cluster. The same holds for sentences, which theoretically should hold one separate thought each.

7 Conclusion

We have confirmed that sentence embeddings are an effective method for retrieving relevant information, achieving the highest accuracy in our experiments and showing improvements of up to 8 % compared to the other retrieval methods. However, it is important to note that the highest achieved accuracy remains at 0.55, without finetuning on QuALITY dataset itself. This indicates that approximately half of the questions in the QuALITY dataset are still not answered correctly. Therefore, sentence embeddings and extractive summarization provide an intermediate solution to the problem of fixed input size of transformers.

However, our additional experiments have shown that sentence embeddings have significant limitations and similarity between them cannot unilaterally distinguish between relevant and irrelevant chunks as shown by multiple density plots. Furthermore, we discovered that similarity is biased toward lexically similar chunks, which can be misleading at times.

Additionally, our experiments have revealed that a chunk size of 64 delivers the best performance across all three models on this task, highlighting the impact of chunk size on overall performance. Upon manual inspection of the selected chunks, we have observed a notable presence of irrelevant information which makes it difficult for a reader to answer the question. This issue arises when a chunk contains a single relevant sentence, while the remaining content within the chunk is irrelevant. Particularly, in cases where multiple relevant pieces of information are distributed across longer spans, fixed chunk sizes often result in the majority of the retrieved text being irrelevant. This limitation could be addressed through generative summarization techniques (which is the current state of the art on the leaderboard[2]). By compressing the most relevant information, this technique could further enhance the retrieval process.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with

[2] https://nyu-mll.github.io/quality/

subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. 2022. Recurrent memory transformer. *Advances in Neural Information Processing Systems*, 35:11079–11091.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Explosion. 2023. spacy models. `https://github.com/explosion/spacy-models/releases/tag/en_core_web_sm-3.6.0`. Version 3.6.0.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Mqag: Multiple-choice question answering and generation for assessing information consistency in summarization.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel R. Bowman. 2021. Quality: Question answering with long input texts, yes! *CoRR*, abs/2112.08608.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2023. Replug: Retrieval-augmented black-box language models.

Damien Sileo. 2023. tasksource: Structured dataset preprocessing annotations for frictionless extreme multi-task learning and evaluation. *arXiv preprint arXiv:2301.05948*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Yucheng Zhou, Tao Shen, Xiubo Geng, Chongyang Tao, Guodong Long, Can Xu, and Daxin Jiang. 2022. Fine-grained distillation for long document retrieval. *arXiv preprint arXiv:2212.10423*.