Attention over pre-trained Sentence Embeddings for Long Document Classification

Amine Abdaoui[1,2,*], Sourav Dutta[1]

[1]Huawei Ireland Research Center, Dublin, Ireland
[2]Oracle, Paris, France

Abstract

Despite being the current de-facto models in most NLP tasks, transformers are often limited to short sequences due to their quadratic attention complexity on the number of tokens. Several attempts to address this issue were studied, either by reducing the cost of the self-attention computation or by modeling smaller sequences and combining them through a recurrence mechanism or using a new transformer model. In this paper, we suggest to take advantage of pre-trained sentence transformers to start from semantically meaningful embeddings of the individual sentences, and then combine them through a small attention layer that scales linearly with the document length. We report the results obtained by this simple architecture on three standard document classification datasets. When compared with the current state-of-the-art models using standard fine-tuning, the studied method obtains competitive results (even if there is no clear best model in this configuration). We also showcase that the studied architecture obtains better results when freezing the underlying transformers. A configuration that is useful when we need to avoid complete fine-tuning (e.g. when the same frozen transformer is shared by different applications). Finally, two additional experiments are provided to further evaluate the relevancy of the studied architecture over simpler baselines.

Keywords

Transformers, Sentence Embedddings, Attention, Long Document Classification.

1. Introduction

The Transformer model [1] is now established as the standard architecture in Natural Language Processing (NLP). Several variants of the original model achieved state-of-the-art results in many tasks [2, 3, 4] including document classification. In addition to their accurate results, transformers are also efficient when compared to recurrent neural network encoders. However, this efficiency drops significantly on long sequences. Indeed, transformers compute $n * n$ self-attention matrices to get the contextualized representations. Therefore, the memory and computational requirements grow-up quadratically with the number of tokens n. For this reason, most transformer-based models are limited to a fixed number of tokens (usually 512 tokens).

To address this limitation, several attempts were made to improve the transformer efficiency on longer sequences. A first family of methods tries to simplify the self-attention complexity by

ReNeuIR'23: Workshop on Reaching Efficiency in Neural Information Retrieval
*This co-author is currently employed by Oracle but this work was conducted when he was at the Huawei Ireland Research Center.
amin.abdaoui@oracle.com (A. Abdaoui); surav.dutta2@huawei.com (S. Dutta)
0000-0002-6160-8461 (A. Abdaoui); 0000-0002-8934-9166 (S. Dutta)

reducing the number of computed weights. Concretely, instead of letting each token attend to every other token in the sequence, these methods restrict the computation of the attention weights to a small number of locations [5, 6]. Another popular approach consists in splitting the long input into smaller chunks that can be modeled more efficiently with a transformer. Then, the obtained representations can be combined using a recurrent neural network or another document-level transformer [7, 8]. Finally, instead of combining the outputs of different chunks using a new model, [9] and [4] implemented a caching mechanism that allows the first tokens of chunk i to have access to the hidden states of the last tokens of chunk $i-1$.

In this paper, we suggest to take advantage of *pre-trained sentence transformers* to get meaningful sentence representations without the need of any further pre-training [10]. The availability and variety of these models allow to easily adapt our framework to different domains and languages[1]. Based on these sentence representations, we evaluate the use of a small *attention layer* to form a document representation by giving higher weights to more important sentences. Note that we do not compute full self-attention matrices between all sentence pairs but only attention weights between the unique document representation and the different sentence embeddings. Indeed, we believe that sentence representations are less sensitive to external context than token embeddings. Similar architectures that also use linear weighted aggregations were evaluated on other tasks [11, 12].

To evaluate these assumptions in the case of long document classification, we compare our proposed architecture with the current state-of-the-art models on three standard datasets. To our knowledge, this is the first detailed evaluation of these models on the same datasets. In addition to complete fine-tuning, we include a setting where the underlying transformers are frozen. Such scenario might be useful when the same transformer is shared by different applications (each application trains only its own task-specific layers).

2. Related Work

Most of the work that tried to adapt transformers to long documents were evaluated on language generation [13, 9, 6]. In this section, we will mainly focus on methods than can be used in language understanding tasks such as classification.

The easiest way to deal with long sequences is to truncate them at the maximum sequence length supported by the model. Usually, the first 512 tokens are used and the following ones are just thrown away. Therefore, the first baseline in this paper will be simple truncation using the Roberta model [3], which is a widely used transformer for Natural Language Understanding. Furthermore, Roberta was used to initialize two other models that are also included in our experiments.

A more sophisticated approach uses sparse self-attention matrices to reduce the transformer complexity. Instead of computing all the $n * n$ weights in each matrix, the idea is to compute only the ones that convey important relationships. For example, Longformer [5] combines a windowed local attention for all tokens with a global attention for few important tokens. On the one hand, the authors proposed to compute attention weights between each token and all its neighbors that are included in a fixed window. On the other hand, they allow important

[1] https://github.com/UKPLab/sentence-transformers

tokens (e.g. [CLS]) to attend to the whole sequence of n tokens. Thanks to these optimisations, the authors were able to pre-train the Longformer model, which is able to handle 4096 tokens, starting from Roberta weights. Another work suggested to choose the computed attention weights dynamically based on the content [6]. However, the model has been designed for character-level language generation as most of the other sparse attention methods. Therefore, we will only include Longformer in our evaluations to represent this family of methods.

Another popular research direction is to use a hierarchical architecture in order to reduce the cost of the self-attention computation. Instead of applying one transformer to the whole sequence, the idea is to stack multiple models that handle a smaller number of inputs. Since transformer's complexity is $O(n^2)$, applying multiple transformers to smaller sequences is better than applying one transformer to the whole sequence. Several studies that used multiple levels of transformers or a recurrent neural network on top of transformers have been proposed [7, 14]. However, most of them have not been shared publicly with the community. SMITH, which is able to handle 2048 tokens, is the one of the rare pre-trained hierarchical models that is available online [8]. The proposed architecture is composed of two levels of abstraction: a sentence-level and a document-level. Each level uses a small transformer that has 4 and 3 layers respectively, 4 attention heads and 256 hidden dimensions. Therefore, the resulting model has less parameters than all the other models studied here which follow the common base architectures (12 layers, 12 attention-heads and 768 hidden dimensions). It was pre-trained using the usual masked word prediction and a novel masked sentence prediction task. Then, it was fine-tuned for document matching using a siamese architecture. In this paper, SMITH will be considered as a baseline in our experiments despite of its small size. To our knowledge, this is the first evaluation of SMITH on the document classification datasets considered.

Finally, [9] proposed TransformerXL which is able to model an unlimited number of tokens. The proposed auto-regressive model is also applied to smaller chunks extracted from the original long documents. However, the modeling is not conducted independently on each chunk. At each time step, the previous hidden states are reused to compute the current ones introducing a sort of memory that propagates across the different segments. Moreover, the usual absolute position embeddings were replaced by relative positional encoding in order to avoid confusion on token positions when handling different segments. The same authors also pre-trained XLNet [9] in order to improve auto-regressive models in NLU tasks. In addition to handling an unlimited sequence length (thanks to the caching mechanism and relative positional encoding proposed in TransformerXL), the authors used permutation language modeling to capture bidirectional context when pre-training the model. For all these reasons, we will include XLNet as a baseline in our experiments.

In this paper, we will compare the above mentioned baselines with an attention-based architecture that relies on pre-trained sentence transformers. These models are usually trained using a siamese architecture on sentence pair datasets to derive semantically meaningful sentence representations [10]. To our knowledge, this is the first attempt to use pre-trained sentence transformers for handling long documents.

3. Methods

In this section, we will detail the studied *Attention over Sentence Embeddings* (AoSE) architecture and compare its complexity and size with existing baselines.

3.1. AoSE Architecture

First, long documents are segmented into sentences using common sentence separators (full stop, line break, etc.). We define a minimum and a maximum number of tokens to avoid generating very small and very long segments that do not correspond to real sentences. Then, a sentence transformer will be used to map each sentence to a fixed dense representation s_i. Relying on such pre-trained models that are already geared towards producing meaningful sentence embeddings is certainly an important advantage. After that, we use an attention layer to combine the normalized sentence embeddings s_i while giving higher weights to important sentences [15]. To calculate these weights α_i, we rely on a small neural network W_s and a trainable context vector u_s that is equivalent to the query in the Transformer's self-attention definition [1].

$$u_i = tanh(W_s \times s_i + b_s) \tag{1}$$

$$\alpha_i = \frac{\exp(u_i^T \times u_s)}{\sum_{j=1}^{t} \exp(u_j^T \times u_s)} \tag{2}$$

The document representation v is then computed using a weighted sum of the different sentence embeddings s_i.

$$v = \sum_{i=1}^{t} \alpha_i \times s_i \tag{3}$$

Finally, a dense layer can be added on top of the document embedding v in order to perform classification. All these steps are presented in Figure 1 below.

3.2. Model Complexity

Let's define the sequence length n as the product of the number of sentences t and the length of one sentence l. The complexity of a vanilla transformer (e.g. Roberta) is therefore $O(t^2 \times l^2)$.

The sentence transformer of SMITH computes full self-attention between all tokens in each sentence, while the document transformer applies a second full self-attention computation between all sentences. Thus, the complexity of the whole SMITH encoder is $O(t \times l^2 + t^2)$.

Longformer computes global attention for g important tokens and local attention for the remaining ones. Let w be the window size for local attention tokens. The Longformer's complexity is therefore $O(g \times t \times l + (t \times l - g) \times w)$.

XLNet computes full self-attention for each chunk. Let c be the maximum length of one XLNet segment (chunk). Even if all the previous hidden states are cached and reused, a given

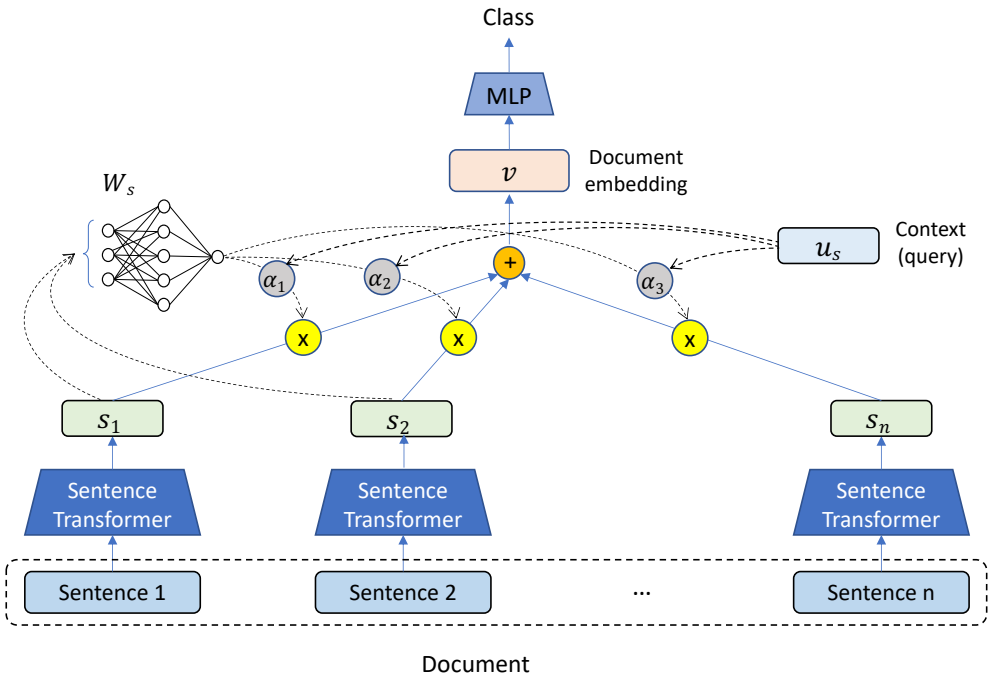

Figure 1: The proposed Attention over Sentence Embeddings (AoSE) architecture.

Table 1
Complexity of the different models (t is the number of sentences, l is the average number of tokens per sentence, g is the number of global attention tokens, w is the window size for local attention tokens, and c is the length of one XLNet segment).

Model	Complexity
Roberta	$O(t^2 \times l^2)$
SMITH	$O(t \times l^2 + t^2)$
Longformer	$O(g \times t \times l + (t \times l - g) \times w)$
XLNet	$O(t \times l \times c)$
AoSE	$O(t \times l^2 + t)$

token can't attend to more than c locations. Therefore, the complexity of one chunk is $O(c^2)$ and the number of chunks is $t \times l/c$. Consequently, the complexity of XLNet is $O(t \times l \times c)$.

The sentence transformer of our proposed architecture is equivalent to the one used in SMITH, but since our document-level attention is linear with the number of sentences, the complexity of our architecture is $O(t \times l^2 + t)$.

Table 2
Different size comparisons of the evaluated models.

Model	Disk size	#Params (million)	Vocab. size	#Layers / #Heads	Hidden dim.	Max input #tokens
Roberta	478 MB	125 m	50.265	12 / 12	768	512
SMITH	47 MB	12 m	30.522	4+3 / 4	256	2.048
Longformer	570 MB	149 m	50.265	12 / 12	768	4.096
XLNet	445 MB	117 m	32.000	12 / 12	768	unlimited
AoSE	480 MB	126 m	50.265	12 / 12	768	unlimited

3.3. Model Size

Due to hardware limitations, we decided to work with the base versions of Roberta, Longformer and XLNet even if large versions were also available. Similarly, we have chosen a base sentence transformer[2] in our AoSE architecture. The chosen sentence transformer was initialized with Roberta base. Therefore, these two models share the exact same number of parameters. SMITH is the only exception as its only available version is much smaller in size. Table 2 presents several size-related measurements for all the models evaluated in this work.

We can also notice that our AoSE architecture has slightly more parameters than Roberta. As mentioned before, our architecture is composed of a sentence transformer (that has the same size as Roberta) and a small attention layer that has less than 1 million parameters. Most of them are located in the W_s matrix that has a shape of 768×768. Therefore, our proposed architecture does not add a lot of parameters when compared to a standard transformer.

4. Experiments

In this section, we will assess the performance of our architecture along with the selected baselines for long document classification.

4.1. Datasets

Three classification datasets (of several thousands of documents each) were chosen to conduct our experiments. The first one is the widely used IMDB dataset [16]. We used the binary version[3] that distinguishes positive and negative movie reviews. The second one is MIND [17], a large-scale dataset for news recommendation. We used the topic classification task from this dataset[4] and discarded a couple of topics that have less than 3 documents. The third one is the 20 News Groups dataset [18]. We used the cleaned version[5] that do not contain headers, signatures, and quotations. Table 3 below shows several statistics related to the number of documents, the length of these documents and the number of classes for each dataset. The only

[2]https://huggingface.co/sentence-transformers/nli-roberta-base-v2
[3]https://huggingface.co/datasets/imdb
[4]https://msnews.github.io
[5]https://huggingface.co/datasets/SetFit/20_newsgroups

Table 3
Statistics of the different datasets: (i) the total number of documents, (ii) the number of long documents (having more than 512 tokens), (iii) the average number of tokens per document, (iv) the maximum number of tokens per document, and (v) the number of labels. The number of tokens were computed using the roberta tokenizer.

Dataset	Split	#Docs all	#Docs >512	Avg #Tokens	Max #Tokens	#Labels
IMDB	train	25.000	7.729	323	3.240	2
	test	25.000	3.537	314	3.257	2
MIND	train	101.523	48.093	696	51.662	15
	test	28.275	13.160	651	28.287	15
20 News Groups	train	11.314	1.245	398	49.561	20
	test	7.532	784	372	132.115	20

additional preprocessing step that was applied to these datasets consisted in removing HTML tags.

4.2. Experimental Settings

Even if XLNet and AoSE can theoretically handle sequences of unlimited length[6], we had to set a maximum sequence length to each one due to practical hardware restrictions. Indeed, we were not able to fine-tune XLNet on very long sequences using our 2×16 GB GPUs (even with a batch size of 1). Therefore, we decided to set the maximum sequence length of XLNet to 4096 tokens in our experiments (which covers all IMDB and more than 99% of MIND and 20 News Groups). In the case our AoSE model, we were able to input up to 8192 tokens which covers almost all the documents included in the three datasets.

We also decided to perform our experiments in two different settings. In the first one, we train all the parameters of every architecture which correspond to standard fine-tuning. In the second setting, we decided to freeze the weights of the underlying transformers. In this case, the different transformers are applied once to produce frozen representations. Then, the training will only happen on the top level parameters of the different architectures. This means that we will only train classifiers for all the baselines, and the classifier along with the attention layer of our AoSE architecture. Training our linear attention layer do not take more time than training the classifier itself.

Regarding the other experimental settings, we set the minimum number of tokens per sentence for our AoSE system to 5 and the maximum value to 250. The document representation in the frozen setting is average pooling as it allowed us to obtain better results for all models. When fine-tuning, we use the default pooling strategy implemented by each model. For all models and all datasets, we set the learning rate to $2e^{-}5$ and the batch size to 16. When the memory of our GPUs is exceeded, we reduce the batch size but use gradient accumulation to simulate the same parameters update as with a batch size of 16. When freezing the transformers, we train all models for 50 epochs on each dataset. When fine-tuning, we train all models for 20 epochs

[6]There is no structural limitation caused by the model definition (for example, the size of the position embeddings matrix).

Table 4
Results on IMDB.

Model	Max seq. length	Fine-tuning Time (hh:mm)	Acc. all	Acc. <=512	Acc. >512	Freezing Time (hh:mm)	Acc. all	Acc. <=512	Acc. >512
Roberta	512	05:40	95.2	95.6	93.0	00:12	91.8	92.2	88.4
SMITH	2048	04:28	91.0	91.0	90.8	00:29	74.0	74.2	73.0
Longformer	4096	22:50	95.6	95.6	95.4	00:30	92.0	92.0	91.2
XLNet	4096	22:20	95.6	95.5	95.7	00:29	92.2	92.2	91.8
AoSE	8192	15:50	**95.7**	**95.7**	**95.8**	00:35	**93.2**	**93.2**	**93.1**

Table 5
Results on MIND.

Model	Max seq. length	Fine-tuning Time (hh:mm)	Acc. all	Acc. <=512	Acc. >512	Freezing Time (hh:mm)	Acc. all	Acc. <=512	Acc. >512
Roberta	512	09:20	83.1	80.7	85.8	00:45	77.4	73.8	81.6
SMITH	2048	13:30	80.8	77.4	84.8	00:52	76.0	71.6	81.1
Longformer	4096	66:45	**84.1**	**80.7**	**88.0**	02:05	77.7	73.7	82.3
XLNet	4096	81:58	83.4	79.8	87.3	04:12	78.3	74.2	82.9
AoSE	8192	83:10	83.5	79.8	87.5	04:58	**79.1**	**75.0**	**83.9**

on IMDB and News Groups, and for 10 epochs on MIND.

4.3. Evaluations

Tables 4, 5 and 6 present the results obtained by the different models on each dataset[7].Overall, we can observe that Longformer, XLNet and AoSE obtain better accuracies on long documents when compared with the remaining baselines. When fine-tuning the transformers, there is no clear best model between them across the three datasets, which joins the conclusions drawn in [19]. However, when freezing the transformers, our AoSE model obtains systematically the best results. We believe that this setting is useful for applications that use the same underlying transformer as encoder for multiple tasks or s imply for applications that cannot afford expensive training. Finally, being much smaller than the other models, SMITH obtains the worst accuracies in both settings across all the datasets.

Regarding the training speed, we can observe that the frozen setting reduces drastically the training time. Since the *Max seq. length* differs from one model to the other, comparing their training times is not straightforward. However, we can use the IMDB dataset for a fair comparison between Longformer, XLNet and AoSE as all IMDB documents can be modeled entirely without any truncation by these three models[8]. In this case, AoSE is faster than the two other models in the fine-tuning setting, but slightly slower in the frozen setting.

[7]For more information about the *Max seq. length* mentioned in these tables, see subsection 4.2.
[8]the longest IMDB document has less than 4096 tokens.

Table 6
Results on 20 News Groups.

Model	Max seq. length	Fine-tuning				Freezing			
		Time (hh:mm)	Acc. all	Acc. <=512	Acc. >512	Time (hh:mm)	Acc. all	Acc. <=512	Acc. >512
Roberta	512	02:15	72.5	71.5	83.2	00:05	63.7	62.6	75.0
SMITH	2048	02:30	60.0	58.6	74.5	00:07	56.4	55.0	71.8
Longformer	4096	10:20	72.7	71.4	**84.3**	00:10	64.1	62.8	77.7
XLNet	4096	13:36	**72.8**	**71.7**	84.2	00:17	65.3	63.8	79.5
AoSE	8192	14:30	72.7	71.5	83.9	00:25	**66.0**	**64.4**	**79.9**

Table 7
Ablation study conducted on IMDB: (i) S-Roberta refers to the chosen sentence transformer used alone and applied to the whole sequence (truncated after 512 tokens); (ii) AoSE-xxx refers to the application of the proposed architecture that also uses S-Roberta (input sequences are truncated after xxx tokens).

Model	Max sequence length	Fine-tuning			Freezing		
		Acc. all	Acc. <=512	Acc. >512	Acc. all	Acc. <=512	Acc. >512
S-Roberta	512	95.3	95.6	92.8	92.2	92.7	89.1
AoSE-512	512	95.4	95.7	93.0	92.6	93.0	89.9
AoSE-1024	1024	95.7	95.7	95.7	93.0	93.1	92.5
AoSE-2048	2048	95.7	95.7	95.8	93.1	93.1	92.9
AoSE-4096	4096	95.7	95.7	95.8	93.2	93.2	93.1

4.4. Ablation Study

We conduct an ablation study to further investigate the relevancy of the studied architecture over simpler baselines. We compare the results obtained by the selected sentence transformer alone (S-Roberta) with different versions of our AoSE architecture that use the same sentence transformer as their first component. The new AoSE versions are fed with sequences that are truncated after 512, 1024, 2048 and 4096 tokens respectively.Table 7 shows the obtained results on the IMDB dataset. It appears that the AoSE architecture is relevant and benefits from increasing the sequence length especially on long documents (that have more than 512 tokens). We also observe a slight improvement on short documents (having less than 512 tokens) in the frozen setting, which may be explained by a better attention layer after training on the additional sentences that appear at the end of long documents.

4.5. Impact of the chosen sentence transformers

Finally, we evaluate the impact of the chosen sentence transformer on the final results. Table 8 below shows the results obtained with three different sentence transformers in the same two settings presented earlier (fine-tuning and frozen). In addition to the already evaluated S-Roberta, two other sentence transformers have been included here: S-BERT[9] and S-MPNet[10].

[9]https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens
[10]https://huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2

Table 8
Results of different sentence transformers used either alone and inside our architecture on the IMDB dataset. Ao(S-Roberta) refers to attention over S-Roberta, the same model used in our previous experiments. Ao(S-BERT) refers to attention over S-BERT. Ao(S-MPNet) refers to attention over S-MPNet.

Model	Max sequence length	Fine-tuning			Freezing		
		Acc. all	Acc. <=512	Acc. >512	Acc. all	Acc. <=512	Acc. >512
S-Roberta	512	95.3	95.6	92.8	92.2	92.7	89.1
Ao(S-Roberta)	8192	**95.7**	**95.7**	**95.8**	**93.2**	**93.2**	**93.1**
S-BERT	512	94.0	94.5	86.2	87.0	87.9	82.1
Ao(S-BERT)	8192	**94.8**	**94.8**	**94.6**	**90.7**	**90.7**	**90.4**
S-MPNet	512	95.2	95.5	93.3	91.7	92.2	88.3
Ao(S-MPNet)	8192	**95.6**	**95.5**	**96.0**	**93.3**	**93.3**	**92.8**

Again, the IMDB dataset is used to conduct these experiments.

Each model is first applied directly to the whole sequence (truncated after 512 tokens)[11]. Then, it is used inside the studied architecture that is able to handle up to 8192 tokens. Overall, it appears that S-BERT obtains lower results than S-Roberta and S-MPNet either when used alone or inside our architecture. Therefore, we can say that the choice of the sentence transformer has an important impact on the final results. But more importantly, we observe that using the same models inside our AoSE architecture allow to improve the results of all models. This improvement is observable in both settings (fine-tuning and freezing), regardless of the underlying sentence transformer.

5. Conclusion

In this paper, we investigated the relevancy of pre-trained sentence transformers for long document classification. To our knowledge this is the first time these pre-trained models are used to handle long documents. To do so, we combine the sentence representations using a trainable attention layer to give high weights to important sentences. We have shown that this simple method is competitive when compared with current state-of-the-art models in the standard fine-tuning mode. We also considered another mode where the underlying transformers are frozen, which allows to speed-up the training and to share the same underlying transformer between different applications. In this case, the AoSE architecture obtains better results. Additional experiments have shown an improvement over the direct application of the same sentence transformers. Finally, relying on pre-trained sentence transformers allows to easily extend our architecture to different domains and languages. For example, we can simply replace the English sentence transformer used in this paper with a multilingual one[12] to handle multilingual texts, whereas XLNet and Longformer need to be pre-trained again on multilingual datasets.

[11]Initial evaluations have shown that applying sentence transformers directly to the whole document gives better results than applying them to each sentence and then averaging the different embeddings.
[12]For example: https://huggingface.co/sentence-transformers/stsb-xlm-r-multilingual

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017.

[2] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[3] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[4] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, Advances in neural information processing systems 32 (2019).

[5] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The long-document transformer, arXiv preprint arXiv:2004.05150 (2020).

[6] A. Roy, M. Saffar, A. Vaswani, D. Grangier, Efficient content-based sparse attention with routing transformers, Transactions of the Association for Computational Linguistics 9 (2021) 53–68.

[7] R. Pappagari, P. Zelasko, J. Villalba, Y. Carmiel, N. Dehak, Hierarchical transformers for long document classification, in: IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), volume https://ieeexplore.ieee.org/abstract/document/9003958, 2019, pp. 838–844. doi:10.1109/ASRU46091.2019.9003958.

[8] L. Yang, M. Zhang, C. Li, M. Bendersky, M. Najork, Beyond 512 tokens: Siamese multi-depth transformer-based hierarchical encoder for long-form document matching, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 1725–1734.

[9] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, R. Salakhutdinov, Transformer-XL: Attentive language models beyond a fixed-length context, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 2978–2988. doi:10.18653/v1/P19-1285.

[10] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019.

[11] C. Li, A. Yates, S. MacAvaney, B. He, Y. Sun, Parade: Passage representation aggregation for document reranking, arXiv preprint arXiv:2008.09093 (2020).

[12] S. Althammer, S. Hofstätter, M. Sertkan, S. Verberne, A. Hanbury, Parm: A paragraph aggregation retrieval model for dense document-to-document retrieval, in: Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part I, Springer, 2022, pp. 19–34.

[13] N. Kitaev, L. Kaiser, A. Levskaya, Reformer: The efficient transformer, in: International

Conference on Learning Representations, 2019.
[14] C. Wu, F. Wu, T. Qi, Y. Huang, Hi-transformer: Hierarchical interactive transformer for efficient and effective long document modeling, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), 2021, pp. 848–853.
[15] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, 2016, pp. 1480–1489.
[16] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, 2011, pp. 142–150.
[17] F. Wu, Y. Qiao, J.-H. Chen, C. Wu, T. Qi, J. Lian, D. Liu, X. Xie, J. Gao, W. Wu, et al., Mind: A large-scale dataset for news recommendation, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3597–3606.
[18] K. Lang, Newsweeder: Learning to filter netnews, in: Machine Learning Proceedings 1995, Elsevier, 1995, pp. 331–339.
[19] H. Park, Y. Vyas, K. Shah, Efficient classification of long documents using transformers, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 702–709. doi:10.18653/v1/2022.acl-short.79.