Advanced Machine Learning Final Project
Transformer for Text Summarization on Long Documents

Kent State Univeristy

MIS 64061- December 2022

Mukhtar Abubakar Yusuf

Lecturer: Prof. Murali Shanker

Abstract

This study aims to address the challenges of Transformers for Text Summarization on Long Documents. The release of the 'Attention Is All You Need paper allows us to attain new State-Of-Art performances in a lot of Deep Learning ecosystems. Natural Language Processing is one of the most impacted fields, given that Transformers are now able to process much longer text sequences than with a Recurrent Neural Network (RNN). However, some problems remain. Indeed, the text length remains an issue, especially when it is longer than thousands of tokens (pieces of words). Fortunately, some papers succeeded in modifying the Transformer architecture to be able to process very long sequences at low computational costs and high speed on very long texts. The comparative result of the two models on the text of different lengths suggests that the Longformer performance significantly outweighs the Bart when the sequence size increase. And that the application of simple tuning for training the models results in weak performance.

Contents

Introduction

The Transformer architecture came from the need to improve the computational complexity of models that process massive data such as sequences and images. The attention mechanism wasn't a new concept and thus wasn't invented by the "Attention Is All You Need" paper (Vaswani et al., 2017). This paper, however, brought a novel architecture that simplified a lot of the State-Of-Art models at that time (2017). Previously, models were based on profound models with Recurrent layers and/or Convolutional layers depending on the task. These models took a very long time to train and thus were difficult to fine-tune. The Transformer architecture proposed in this paper is about text translation and summarization, which uses an encoder/decoder architecture. Some words were less common in the Abstract in comparison to the core texture. Instead, a simple feed-forward network composed of dense layers was combined with the attention layers.

However, it is worth noting that even with this architecture, the performances (in terms of time, metrics, and computational costs) were still poor on very long sequences. That is profound, especially with those that contain thousands of words.

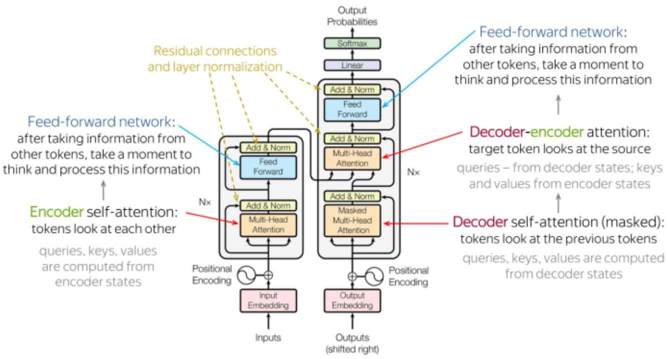
Fortunately, between 2019 and 2020, a few papers about the new type of Transformer changed everything. These new architectures were very similar to the original one but with a lighter and smarter attention layer. This allowed the Transformer to process very long sequences with a linear computational cost. Compared to the exponential computational costs of the regular architecture, it is a significant improvement.

BigBird (Zaheer et al., 2020), Reformer (Kitaev, Kaiser, & Levskaya, 2020), and Longformer (Beltagy, Peters, & Cohan, 2020), to name a few, are now considered as State-Of-Art models when it comes to long sequences. In this work, I will focus on the Longformer in one of the most challenging Natural Language Processing tasks: abstractive text summarization.

The Transformer Architecture

The detailed architecture is depicted in figure1

Figure1: Transformer Architecture


Output Probabilities
Softmax
Linear
Residual connections and layer normalization
Feed-forward network: after taking information from other tokens, take a moment to think and process this information
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Feed-forward network: after taking information from other tokens, take a moment to think and process this information
Decoder-encoder attention: target token looks at the source
queries – from decoder states; keys and values from encoder states
Nx
Nx
Add & Norm
Masked Multi-Head Attention
Encoder self-attention: tokens look at each other
queries, keys, values are computed from encoder states
Decoder self-attention (masked): tokens look at the previous tokens
queries, keys, values are computed from decoder states
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

Objective

The overall objective is to address the challenges of Transformer for Text Summarization on Long Documents

Text summarization

Text summarization is the task of summarizing a long text into a much shorter sequence with minimal information loss.

This task can be done in two different ways:

- Text Extraction, which extracts the most significant sentences and groups of words of the text. It is a classification task similar to Named Entity Recognition.
- Text Abstractive Summarization which resumes the long document into a simple and shorter text without losing its meaning of it.

On long sequences, the Text Abstraction task performances are fragile. Often, the best method was to combine the first step of text extraction followed by an abstractive text summarization of the result (which was a shorter text) at the cost of a loss of information.

Another method was to chunk the long sequence into pieces that could be processed separately by a regular Transformer. The results were then concatenated into a bigger text, on which a new step of text abstractive summarization was done.

The Debate Sum dataset

The Debate Sum Dataset is a dataset about competitive debates organized by the National Speech and Debate Association (US) since 2013. See typical debates in appendix (1).

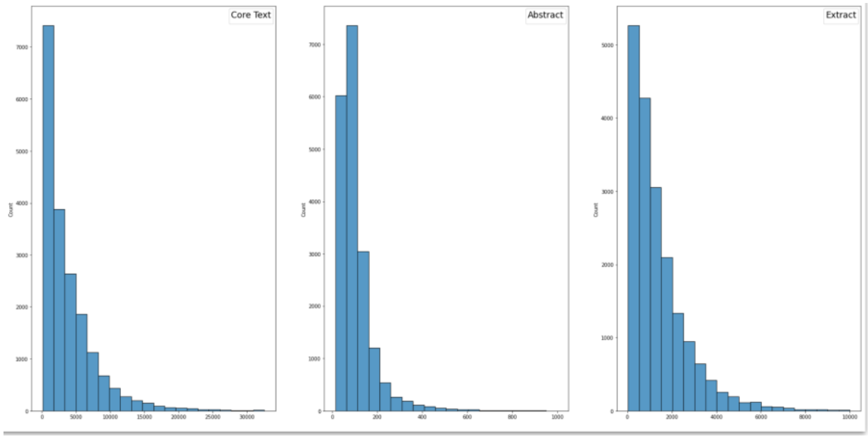
The dataset consists of 187328 debates with:

- One core text which can count more than 5000 words;
- One Abstract that generally counts less than 200 words;
- One Extractive summary that counts less than 2000 words.

The themes of the debates vary each year, but the general themes seem to be similar. During this experiment, I used the 2019 debates with the following theme:

"The United States federal government should substantially reduce Direct Commercial Sales and/or Foreign Military Sales of arms from the United States." The visualized Text Summaries are depicted in Figure 2.

Figure 2: visualized Text Summaries



The themes of the debates vary each year, but the general themes seem to be similar. During this experiment, I used the 2019 debates with the following theme:

"The United States federal government should substantially reduce Direct Commercial Sales and/or Foreign Military Sales of arms from the United States."

Thanks to a WordCloud analysis, I can see which word is the most represented in the dataset for this year. The same WordCloud analysis was done on the Abstracts to see if there were any differences, and it seems that the abstracts are indeed covering the same topics with minor differences.

In this dataset, I used the Core texts and the abstract. The extractive summaries were discarded.

The preprocessing was done using the correct model tokenizers:

- BART Tokenizer, which is similar to the Roberta tokenizer, uses a byte-level BPE.
- Longformer Tokenizer, which is identical to the ROBERTA tokenizer.

In order to compare the results, two transformers were used:

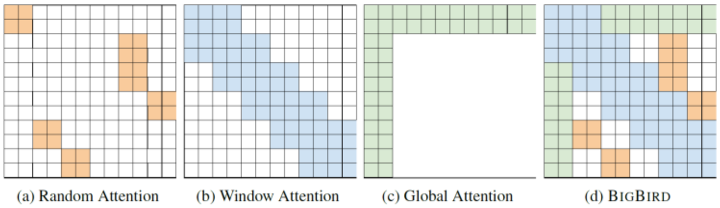
- The longformer is a new attention mechanism;
- The BART (Lewis et al., 2019) transformer.

Longformer

The Longformer is a new type of Transformer that use a **Global Sliding Attention** mechanism. This mechanism is different from the regular Transformer because it doesn't compute the attention on every word. It is a 'sparse' attention that allows the model to be faster and to reduce memory costs (Beltagy et al., 2020) drastically.

To the left is the original self-attention that computes the scaled dot product to every word in the sequence. On the right, the Global Sliding attention mechanism doesn't compute the Scaled Dot Product on every word (see the white squares). This figure comes from the Longformer paper. A token with global attention attends to all tokens across the sequence, and all tokens in the sequence attend to it. The Attention Structures are illustrated in figure 3.

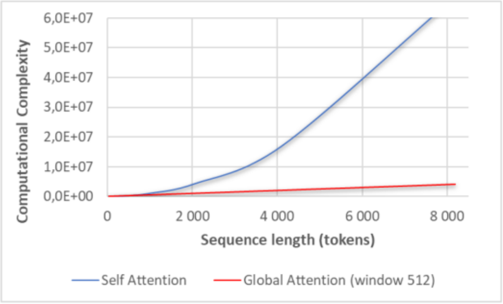
Figure 3: Various Attention Structures



(a) Random Attention (b) Window Attention (c) Global Attention (d) BIGBIRD

The computational complexity of the global attention layer is O(nw) with n the sequence length and w the attention window size. It is far less expensive compared to the O(n²d) with d the embedding size of the regular transformers (regular self-attention). The computational complexities are depicted in Figure 4.

Figure 4: Computational Complexity



The Longformer is mostly used for NLP tasks such as:

- Question Answering
- Language Modelling
- Abstractive summarization

Its usage is still very low compared to regular transformer models such as BERT.

Longformer models can typically handle more than 1 thousand tokens (a piece of words) and can process sequences up to 16k tokens.

BART

BART uses a standard Tranformer-based neural machine translation architecture similar to the one of BERT and GPT. Thus, it uses the regular self-attention layer, and its computational complexity is $O(n^2d)$.

It is one of the best models when it comes to text summarization.

BART has a sequence length limit of 512 tokens. Above this limit, the performance starts to drop significantly.

Methods

Performance comparison

This idea was to compare the validation loss on both models on different text lengths.

During this experiment, I used the debates from the year 2019, and I kept the ones with a sequence length of over 2000. I then conduct four different experiments with the same parameters on both models with different text lengths (512, 1024, 2048, and 4096). Smith proposes several efficient ways to set the hyper-parameters that significantly reduce training time and improves performance (Smith, 2018).

These are the experiments:

1. BART with a Complex_hp_tune_BART_BASE.ipynb (model_1)
2. Longformer with a Complex_hp_tune_Longformer_BASE.ipynb (model_1)
3. BART with a Simple_hp_tune_BART_BASE.ipynb (Model_2)
4. Longformer with a Simple_hp_tune_Longformer_BASE.ipynb (model_2)

All of the texts come from the same text dataset and are truncated (and padded if needed) to the correct length. Both models were built with the help of the HuggingFace library architectures. The architectures were then wrapped into Tensorflow custom training loops.

Training transformer models on very long texts can be complex because:

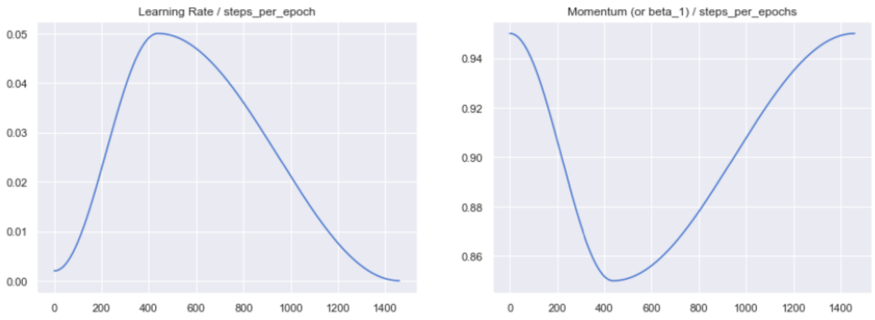
- First, the training time is very long, and the hyperparameter tuning is complex;
- Second, the complexity of the model may lead to overfitting and request for a lot of training data.

To answer both of these problems, I used pre-trained layers. The pre-trained used was the 'allenai/led-base-16384' for the longformer and the 'Facebook/bart-base' for the BART.

To keep the pre-trained weights intact, I used a 1Cycle scheduler with a meager learning rate. The 1Cycle scheduler starts with a low learning rate and then quickly increases to the maximum learning rate (1e-6).

It does the opposite for the momentum, allowing the model to stabilize around a good minimum, as shown in figure 5.

Figure 5: 1 Cycle scheduler Vs. Momentum Stabilizer



The number of trainable parameters for each model was similar.

The BART model is composed of 6 encoders and six decoders with a feed-forward network composed of two dense layers for a total number of trainable parameters of 139,470,681.

The Longformer model is also composed of 6 encoders and six decoders with two dense layers feeding the forward network but for a total number of trainable parameters of 161,894,745.

Training and fine-tuning:

Both models were then fine-tuned completely on 3140 texts with a length of over 1000 words. The configs were identical to the previous experiment.

For the **BART model**, I used a sequence length of 512 tokens by truncating every text from the corpus to the correct length.

For the **Longformer model**, the sequences were truncated at 4096 tokens and padded otherwise.

Training parameters:

• Batch size: 8 (Bart), 1 (longformer) , • Learning rate: 1e-5 (Bart), 1e-6 (long former), • Max_length: 512 (Bart), 4096 (Longformer), • Epochs: 5 . and • Callbacks: Early Stopping + Scheduler 1Cycle.

I used one Epoch for model_2.

For the training, a GPU P100 on google colab was used.

Note: to not harm the weights, I tried to freeze some parts of the Transformer (encoder, decoder, embeddings) and slowly unfreeze them during the training. However, the results weren't promising; thus, I skipped that part.

For the ***evaluation*** part, I used two different metrics:

• BLEU which is how much the words (and/or n-grams) in the machine-generated summaries appeared in the human reference summaries. An n-gram is a sequence of words (2-gram is a sequence of two words).

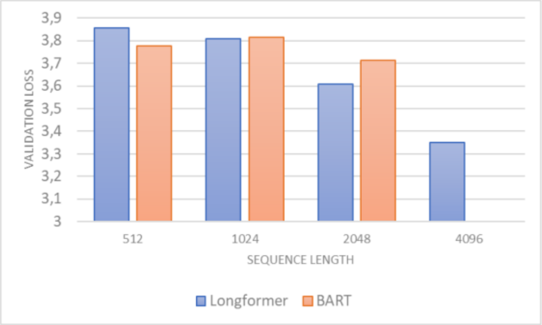
• Rouge-L which is similar to BLEU but uses the sentence structure by keeping the most extended standard sequence to compute the scores. The order of the words in the sentence is the most important here.

However, in model_2, we only used BLEU.

Results

1. For moel_1, The comparison of the two models on the text of different lengths showed that the Longformer effectively beats the Bart when the sequence size increase. It also showed that the BART memory consumption increased more than the Longformer one because the 4096 didn't fit into the GPU even if the BART had less trainable parameters.
2. For model_1, The Longformer validation loss steadily decreases with the increasing length, whereas the BART validation loss didn't show any improvement and wasn't measurable with the 4096 tokens sequence length as depicted in figure 7

Figure 7: Performance based on Validation Loss

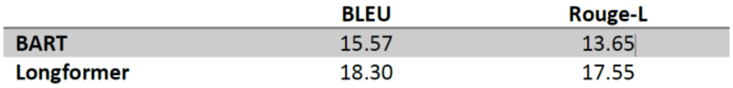


The experiment also showed that the BART outperforms the Longformer on short sequences. This can be due to the sparsity of the attention layer of the Transformer.

The training was, however, always in favor of the BART model. This can be explained by the higher number of trainable parameters of the Longformer model used in this experiment.

For model_1, the full training of the BART and the Longformer took 10 minutes and 1h10 per Epoch, respectively. This is mainly due to the sequence length used for training (8 times smaller for the BART model), as shown in Table 1.

Table 1: Performance on Evaluation Metrics



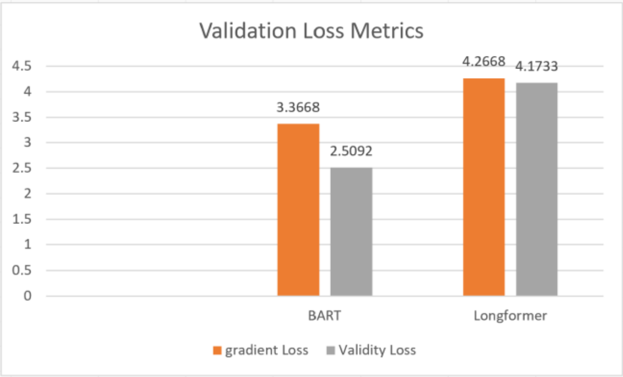
	BLEU	Rouge-L
BART	15.57	13.65
Longformer	18.30	17.55

The results are a bit better for the Longformer. However, it is not enough to affirm that the Longformer is better than a regular transformer such as BART on this particular dataset.

This surprising result may be due to the fact that the abstracts of this dataset can be constructed from the beginning of the text. Hence, it seems that the end of the text doesn't bring anything new that the model doesn't already know from the beginning of the text.

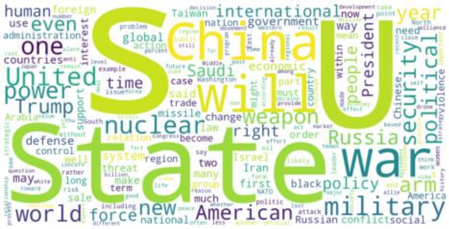
3. Surprisingly, in model_2, the Longformer validation loss steadily increases with the increasing length, as shown in figure 8.

   Figure 8: Validation Loss in Model_2

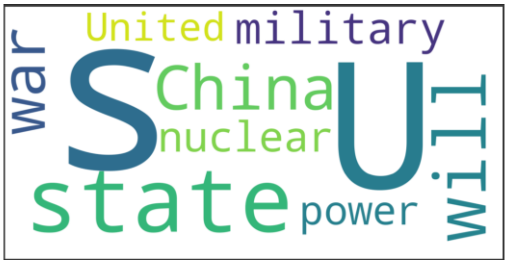


4. Upon the WordCloud analysis, it is evident that the abstracts are indeed covering the same topics with minor differences. Some words were less common in the Abstract in comparison to the Core texts

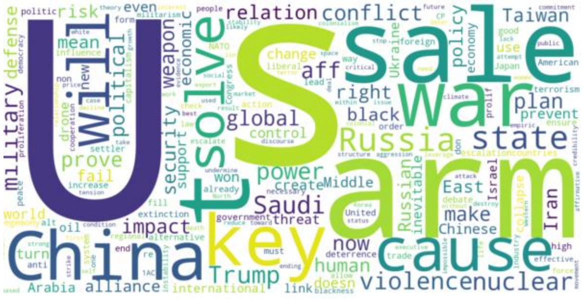
Core texts:



Ten Maximum words



Abstracts:



Some words were less common in the Abstract in comparison to the Core texts. For example, **Sale**, **Key,** and **Arm** are more common in the abstracts than in the core texts.

Also, I used a batch size of 1 for the Longformer because of memory consumption. A meager batch size may reduce the impact of the first samples seen during each Epoch (the model will gradually forget about the first samples).

Conclusions

The Longformer is an efficient tool for processing long sequences for Natural Language tasks such as text summarization. It reduces the computational costs compared to the regular Transformer and improves performances thanks to its new attention mechanism.

Although in Deep Learning, there is no free meal but given the results of the two model, we confidently assume that the application of a complex hyper-parameter in assessing the performance of a Transformer yield a better result (model_1). This is much better than when we used a simple model with simple fine-tuning.

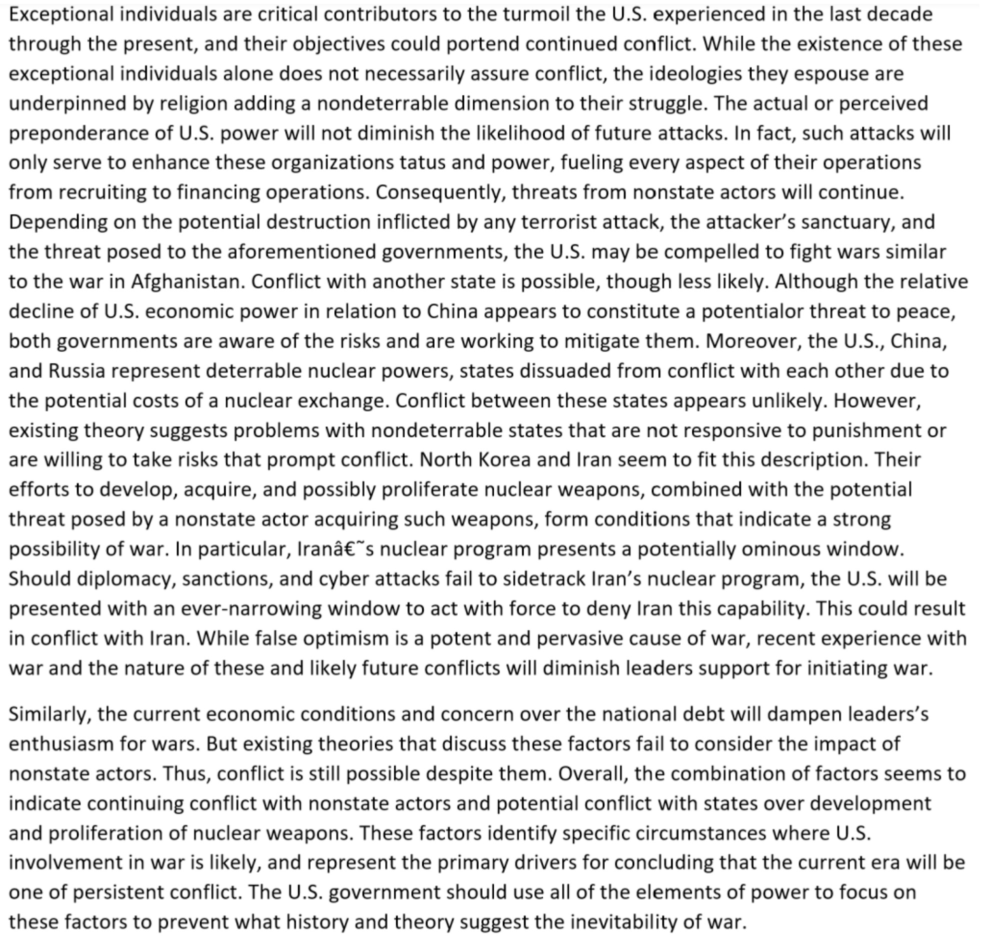
The dataset used there is very complex. It appears that the use of a regular transformer in the beginning of the long texts is sufficient to keep up with the performances of the Longformer on long sequences (with six times less training time) as indicated in model_1.

My Contributions to the Class.

Research in Deep Learning that has added value to the body of knowledge in terms of the evolution of Natural Language Processing (NLP) that seems to have outperformed RNN using the "Attention Concept," which process much longer Text Sequences on Transformer Architecture

Appendix 1

An example of debate (from the paper of the dataset (Lewis et al., 2019))



Exceptional individuals are critical contributors to the turmoil the U.S. experienced in the last decade through the present, and their objectives could portend continued conflict. While the existence of these exceptional individuals alone does not necessarily assure conflict, the ideologies they espouse are underpinned by religion adding a nondeterrable dimension to their struggle. The actual or perceived preponderance of U.S. power will not diminish the likelihood of future attacks. In fact, such attacks will only serve to enhance these organizations tatus and power, fueling every aspect of their operations from recruiting to financing operations. Consequently, threats from nonstate actors will continue. Depending on the potential destruction inflicted by any terrorist attack, the attacker's sanctuary, and the threat posed to the aforementioned governments, the U.S. may be compelled to fight wars similar to the war in Afghanistan. Conflict with another state is possible, though less likely. Although the relative decline of U.S. economic power in relation to China appears to constitute a potentialor threat to peace, both governments are aware of the risks and are working to mitigate them. Moreover, the U.S., China, and Russia represent deterrable nuclear powers, states dissuaded from conflict with each other due to the potential costs of a nuclear exchange. Conflict between these states appears unlikely. However, existing theory suggests problems with nondeterrable states that are not responsive to punishment or are willing to take risks that prompt conflict. North Korea and Iran seem to fit this description. Their efforts to develop, acquire, and possibly proliferate nuclear weapons, combined with the potential threat posed by a nonstate actor acquiring such weapons, form conditions that indicate a strong possibility of war. In particular, Iranâ€˜s nuclear program presents a potentially ominous window. Should diplomacy, sanctions, and cyber attacks fail to sidetrack Iran's nuclear program, the U.S. will be presented with an ever-narrowing window to act with force to deny Iran this capability. This could result in conflict with Iran. While false optimism is a potent and pervasive cause of war, recent experience with war and the nature of these and likely future conflicts will diminish leaders support for initiating war.

Similarly, the current economic conditions and concern over the national debt will dampen leaders's enthusiasm for wars. But existing theories that discuss these factors fail to consider the impact of nonstate actors. Thus, conflict is still possible despite them. Overall, the combination of factors seems to indicate continuing conflict with nonstate actors and potential conflict with states over development and proliferation of nuclear weapons. These factors identify specific circumstances where U.S. involvement in war is likely, and represent the primary drivers for concluding that the current era will be one of persistent conflict. The U.S. government should use all of the elements of power to focus on these factors to prevent what history and theory suggest the inevitability of war.

References:

Beltagy, I., Peters, M. E., & Cohan, A. 2020. Longformer: The Long-Document Transformer.

Kitaev, N., Kaiser, Ł., & Levskaya, A. 2020. Reformer: The Efficient Transformer. ***Published as a conference paper at ICLR 2020***.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. ***arxiv***.

Smith, L. N. 2018. A disciplined approach to neural network hyper-parameters. ***US Naval Research Laboratory Technical Report***.

Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A. N., Gouws, S., Jones, L., Kaiser, Ł., Kalchbrenner, N., Parmar, N., Sepassi, R., Shazeer, N., & Uszkoreit, J. 2017. Attention Is All You Need. ***31st Conference on Neural Information Processing Systems***.

Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., & Ahmed, A. 2020. Big Bird: Transformers for Longer Sequences. ***deepsea.princeton.edu***.