BERT Squared: Read + Verify System for SQuAD 2.0

Jiayu Lou
Stanford University
jiayul@stanford.edu

Abstract

Question answering (QA) is a well-researched problem in NLP. In spite of being one of the oldest research areas, QA has application in a wide variety of tasks, such as information retrieval and entity extraction. The publication of BERT last year has significantly improved the state-of-the-art performance of SQuAD v-1.1 and v-2.0. This paper proposes a “Read + Verify” model by stacking two BERT models on top of each other. After the first model predicts the preliminary results of SQuAD, these preliminary results are then fed into the second model in the form of trimmed context paragraph to be examined closely to reach a final decision. After fine-tuning, the “Read + Verify” system modestly improved the baseline BERT model for EM and F1.

1 Introduction

The task of Question Answering has gained prominence in the past few decades for testing the ability of machines to understand natural language, essentially a machine reading comprehension task focusing on an agent’s ability to read a piece of text and subsequently answer questions about it. As one of the foundations for human interactions and communications, this particular task sees increasing attention due to the advances in computational power, brilliant algorithms, and available datasets. While many QA datasets have been introduced over the past few years, SQuAD is one of the most popular among them, where data is represented as question-context pairs with a span in the context as the answer to the question. [3]

2 Related Work

In the past decade, there have been many successful attempts in attacking the problem, including both PCE(Pre-trained Contextual Embeddings) and non-PCE methods. One of the famous non-PCE models is the Transformer, published in the paper *Attention is All You Need* published in 2017 that models the long-range dependencies with attention without using RNN [5]. Built upon this paper, Google published another paper *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* in October 2018, which caused a stir in the Machine Learning community by presenting state-of-the-art results in a wide variety of NLP tasks, including Question Answering [1]. BERT pushed the performance on both SQuAD 1.1 and SQuAD 2.0 to a new level, and is considered a major breakthrough in QA recent years.

Besides these two famous models, this paper is also inspired by the idea from the paper *Read + Verify: Machine Reading Comprehension with Unanswerable Questions*, which leverages an answer verifier to decide whether the predicted answer is entailed by the input snippets [2].

3 Approach

Machine reading comprehension with unanswerable questions aims to abstain from answering when no answer can be inferred. In addition to extract answers, previous works usually predict an additional "no-answer" probability to detect unanswerable cases. However, they didn't validate the answerability of the question by verifying the legitimacy of the predicted answer. This model utilizes the baseline BERT models to preliminary prediction; after the baseline BERT predicts an answer, the model then snipped the original context to less than 192 words with the predicted answer centered in the middle, and then the snipped context is concatenated with the original query text to be fed into a second BERT which closely examines the trimmed context and predicts a binary result to determine if there is indeed an answer. Since the model uses the BERT model twice, it is necessary to understand the original BERT paper, and the Transformer mechanism thoroughly.

3.1 Transformers

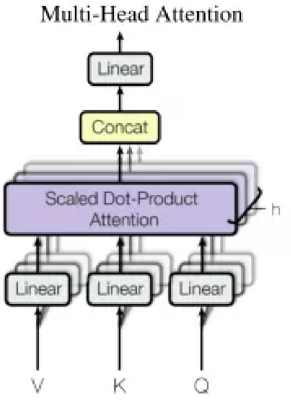

Figure 1: Multi-head attention

The attention mechanism in the Transformer is interpreted as a way of computing the relevance of a set of values (information) based on some keys and queries. Basically, the attention mechanism is used as a way for the model to focus on relevant information based on what it is currently processing. If we only computed a single attention weighted sum of the values, it would be difficult to capture various different aspects of the input. To solve this problem the Transformer uses the Multi-Head Attention block. This block computes multiple attention weighted sums instead of a single attention pass over the values – hence the name "Multi-Head" Attention.

As for the attention mechanism, the Transformer uses a particular form of attention called the "Scaled Dot-Product Attention" which is computed according to the following equation:

$$Attention\left(Q,K,V\right) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q is the matrix of queries packed together and K and V are the matrices of keys and values packed together. d_k represents the dimensionality of the queries and keys. The size of the dot product tends to grow with the dimensionality of the query and key vectors though, so the Transformer rescales the dot product to prevent it from exploding into huge values.

3.2 BERT

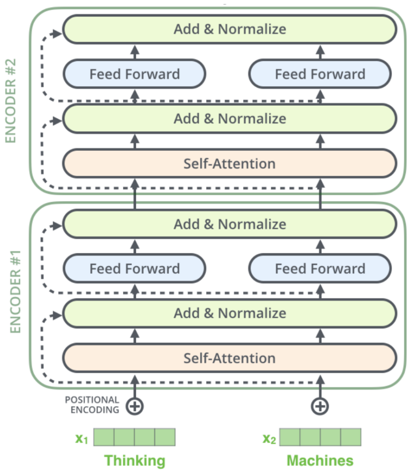

Figure 2: BERT encoders

Built upon Transformers, BERT's model architecture is a multi-layer bidirectional Transformer encoder with the hidden layer size as 768, number of layers as 12, and number of self-attention heads as 12. The chart on the right is a high-level description of a Transformer with 2 stacked encoders. The input is a sequence of tokens, which are first embedded into vectors and then processed in the neural network. The output is a sequence of vectors of size H, in which each vector corresponds to an input token with the same index.

BERT applies the bidirectional training of Transformer to language modelling. The paper's results show that a language model which is bidirectionally trained can have a deeper sense of language context and flow than single-direction language models. BERT achieves this goal by proposing two new pre-training objectives: the "masked language model" (MLM) and the "next sentence prediction" task.

Before feeding word sequences into BERT, 15% of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence.

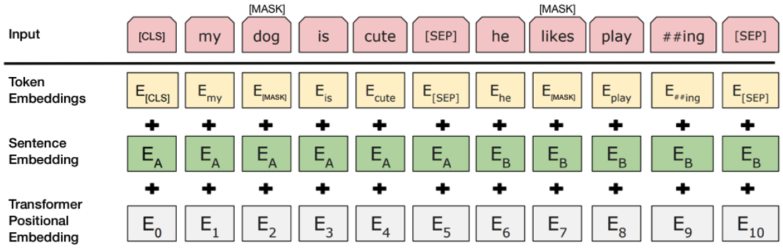

Figure 3: BERT MLM and next sentence prediction

The second task is next sentence prediction.In order to train a model that understands sentence relationships, the paper pre-trains a binarized next sentence prediction task that can be trivially generated from any monolingual corpus. Specifically, when choosing the sentences A and B for each pretraining example, 50% of the time B is the actual next sentence that follows A, and 50% of the time it is a random sentence from the corpus.

3.3 Read + Verify Model

This paper proposes a model that utilizes the BERT baseline model with weights pretrained for these two tasks, and stacks a linear layer upon that baseline for answer index prediction (Model 1), or a one-layer CNN model upon that for binary answer classification (Model 2). The baseline BERT model has a hidden size of 768, a hidden dropout rate of 0.1, attention dropout of 0.1, 12 attention heads, 12 hidden layers, and intermediate size of 3072 (the size of feed-forward layer in the Transformer encoder).

After model 1 predicts a result, the original context is trimmed to be a fixed length that is much shorter than the original length, with the predicted answer centered in the middle. If Model 1 predicts that the question is unanswerable, the best non-null answer is used instead. The trimmed context is then concatenated with the original query text, separated by label [SEP] and then fed into model 2 to be more closely examined and verified. The purpose of model 2 is to scrutinize the one or a few sentences that contains the predicted answer, and eventually make a binary prediction (0 for no answer vs. 1 for there is an answer) to determine the legitimacy of the predicted answer. Note that, if the trimmed context doesn't contain the actual correct answer provided in the training data, then Model 2 should try to predict a label of 0. When Model 2 predicts a question has an answer, then the best non null answer predicted from Model 1 will be adopted.

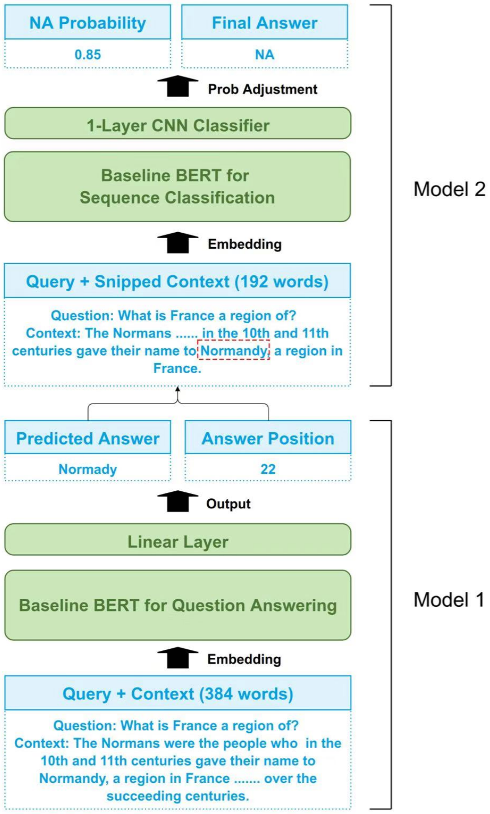
NA Probability
0.85
Final Answer
NA
Prob Adjustment
1-Layer CNN Classifier
Baseline BERT for Sequence Classification
Embedding
Query + Snipped Context (192 words)
Question: What is France a region of?
Context: The Normans ...... in the 10th and 11th centuries gave their name to Normandy, a region in France.
Model 2
Predicted Answer
Normady
Answer Position
22
Output
Linear Layer
Baseline BERT for Question Answering
Embedding
Query + Context (384 words)
Question: What is France a region of?
Context: The Normans were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France .... over the succeeding centuries.
Model 1

Figure 4: Read + Verify model layers

Due to the limited computational resources, I didn't train two BERT models end-to-end, instead the task was decomposed into two steps. Model 1 was trained first to get a reasonable performance; then the results of Model 1 is freezed to be used as the training data for model 2. Both BERT models will be fine tuned from the pre-trained weights, which are generated from HuggingFace implementation (https://github.com/huggingface/pytorch-pretrained-BERT.git).

After Model 2 predicts the answer, we will need to adjust the NA probability because the classifier has been trained on the training set that has a different class distribution compared to dev/test set: in training only ⅓ of the questions are answerable, but in the dev/test set 50% of the questions are unanswerable. Due to that difference in data distribution, the probability predicted from Model 2 will be adjusted to account for that difference. Specifically,

$$\hat{p}\left(\omega_i|\mathbf{x}\right) = \frac{\frac{\hat{p}(\omega_i)}{\hat{p}_t(\omega_i)}\hat{p}_t\left(\omega_i|\mathbf{x}\right)}{\sum_{j=1}^{n}\frac{\hat{p}(\omega_j)}{\hat{p}_t(\omega_j)}\hat{p}_t\left(\omega_j|\mathbf{x}\right)}$$

Where $\hat{p}_t\left(\omega_i\right) = N_t^i/N_t$, representing the estimated a priori probability of belonging to certain class in the training set, $\hat{p}\left(\omega_i\right)$ denotes the a priori probability in dev/test set, $\hat{p}_t\left(\omega_i|\mathbf{x}\right)$ denotes the estimated a posteriori probability of belonging to that class provided by the classifier, and $\hat{p}\left(\omega_i|\mathbf{x}\right)$ denotes the corrected a posteriori probability.

The baseline model is based on BiDAF in the paper *Bidirectional Attention Flow for Machine Comprehension*. Unlike the original BiDAF model, this implementation does not include a character-level embedding layer. The result achieved on the baseline model is EM: 55.991 and F1: 59.291 [4].

4 Experiments

4.1 Data

We used Stanford Question Answering Dataset (SQuAD), containing around 150k questions in total, and roughly half of the questions cannot be answered using the provided paragraph.

4.2 Evaluation Methods

Performance is measured via two metrics: Exact Match (EM) score and F1 score, as described in the default project handout.

4.3 Experimental Details

As noted in the above section, the two models are trained separately, and the model 1 is freezed when training model 2. Here are parameters that I have used that are different from the default value implemented in Huggingface version of baseline BERT.

Table 1: Model parameters

Parameters	Model 1	Model 2
Train Batch Size	12	12
Learning Rate	5e-5	3e-5
Num Training Epoch	3	3
Max Sequence Length	384	192
Max Query Length	64	64
Null Score Diff Threshold	-3	N/A
Kernel Size	N/A	5
CNN Dropout	N/A	0.4

4.4 Results

The results from two sources are presented below in Table 2: only using Model 1, and using the full model (Model 1 + Model 2). Overall, the model has considerable bias as demonstrated by the large discrepancy between the training performance and dev/test performance. Compared to simply using Model 1, Model 2 slightly improved the dev performance by 3 in EM and 3.55 in F1, and improved the test performance by 2.21 in EM and 2.13 in F1. Model 2's major contribution is in boosting accuracy in answerable questions, while the performance for unanswerable questions dropped.

Table 2: Experimental results

	Model 1 Only			Model 1 + Model 2		
Metrics	**Training**	**Dev**	**Test**	**Training**	**Dev**	**Test**
EM	80.65	72.83	71.50	91.03	75.83	73.71
F1	86.35	75.14	74.28	95.16	78.69	76.41
HasAns EM	72.85	57.93	N/A	87.07	66.52	N/A
HasAns F1	81.40	62.76	N/A	93.26	72.51	N/A
NoAns EM	96.24	86.52	N/A	98.93	84.37	N/A
NoAns F1	96.24	86.52	N/A	98.93	84.37	N/A

5 Analysis

The key assumption that Model 2 outperforms Model 1 is because Model 2 has a much shorter context: since the irrelevant parts of context have already been removed thanks to the prediction of Model 1, the trimmed context reduces the distraction and allows Model 2 to fully focus on the most relevant sentences to fine-tune the answer. To better understand why Model 2 performs better in classification, I selected a few examples where Model and visualized the attention distribution.

5.1 Example 1: Model 2 correctly predicts an answerable while Model 1 fails

Question: From which countries did the Norse originate?

Model 1 Answer: NA

Model 2 Answer: Denmark, Iceland and Norway

Correct Answer: Denmark, Iceland and Norway

Context: They were descended from Norse raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia.

Table 3: Top 5 words attended by "originate" in Model 1

Word	descended	originate	raiders	[SEP]	[SEP]
In scope	yes	yes	yes	no	yes
Attention prob	0.6280	0.1463	0.0414	0.0273	0.0263

Table 4: Top 5 words attended by "originate" in Model 2

Word	descended	they	were	from	[SEP]
Attention prob	0.8089	0.0328	0.0272	0.0251	0.0220

Table 5: Top 5 words that attend "Denmark" the most in Model 1

Word	which	frankish	roman	iceland	##ish
In scope	yes	yes	yes	yes	yes
Attention prob	0.2578	0.1097	0.1086	0.1010	0.0951

Table 6: Top 5 words that attend "Denmark" the most in Model 2

Word	which	countries	##ish	from	roman
Attention prob	0.7680	0.5168	0.0862	0.0813	0.0778

Analysis on the example: As expected, Model 2 has a better ability to focus on the key words compared to Model 1. for example, the word couple couple “originate” and “descended” is assigned a

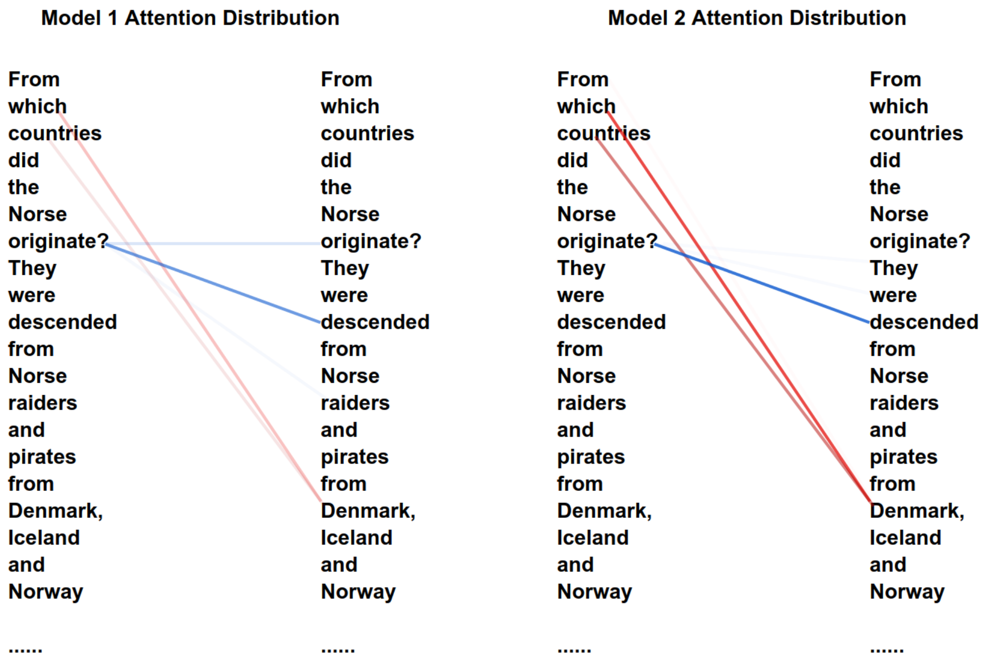

Figure 5: Attention distribution in Example 1

higher attention score in Model 2 than in Model 1, which indicates that Model 2 has more confidence in believing that “originate” and “descended” are synonyms. However, in 9 out of 10 cases, the top 5 words in Model 1 are already within in the scope of the snipped context (which is 128 words long). This partially explains why Model 2 didn’t manage to significantly outperform Model 1: since the self attention layers Model 1 are less “spread out” or “distracted” than we believe, it already had great performance in focusing on the right scope. Therefore, the value added by Model 2 is much less significant than previously believed.

5.2 Example 2: Model 2 mistakenly predicts an unanswerable question

Question: When did the Frankish identity emerge?

Model 1 Answer: NA

Model 2 Answer: first half of the 10th century

Correct Answer: NA

Context: The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

Table 7: The top 5 words attended by “Frankish” in Model 1

Word	frankish	norman	##s	[SEP]	[SEP]
In scope	yes	yes	yes	no	yes
Attention prob	0.3202	0.1747	0.0932	0.0449	0.0445

Table 8: The top 5 words attended by “Frankish” in Model 2

Word	norman	##s	century	10th	the
Attention prob	0.4343	0.1501	0.0849	0.0725	0.0377

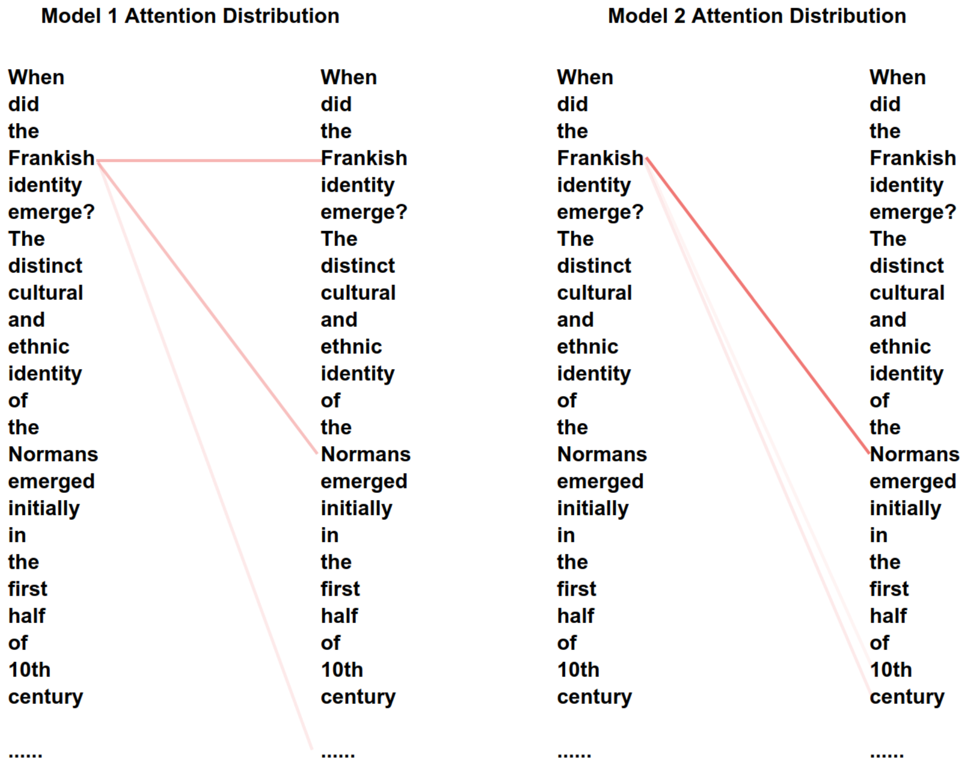
Model 1 Attention Distribution
Model 2 Attention Distribution
When did the Frankish identity emerge? The distinct cultural and ethnic identity of the Normans emerged initially in the first half of 10th century
......

Figure 6: Attention distribution in Example 2

Analysis on the example: In this example where the question is unanswerable, the ability to focus backfires. In Model 1, the attention scores very often demonstrate a long tail shape, spreading out across many far away tokens. Since Model 2 has a much shorter context, the attention weights are more "squeezed" toward a few potentially relevant tokens, therefore increasing their positive relationships with the word "Frankish". In this example, Model 2 gives a much higher attention score due to this "squeezing" effect, and therefore mistakenly inflates the perceived positive relationship between "Normans" and "Frankish".

5.3 Summary

By closely examining the attention distribution of these two examples, one may realize that the shortened context is both a blessing and a curse. On one hand, it helps the attention layers to concentrate on the most relevant keywords and increase the confidence level when the predicted answer is correct. On the other hand, when the question is actually unanswerable but Model 1 is predicting a null likelihood on the edge of yes or no, the "squeezing effect" might mistakenly boost the confidence level for that answer. The final performance of the model is a result of these two opposing forces, depending on the actual distribution of the two classes.

The primary purpose of Model 2 is to enforce a stronger attention mechanism by artificially trimming off a large part of the context. However, Model 2 only manages to improve Model 1 by a limited margin because it seems that Model 1 already has a good ability to "concentrate". As demonstrated in example 1, in some effective self-attention layers the top 5 words being attended are already in the scope of snipped context and accounts for more than 80% of the attention distribution, therefore leaving limited space for improvement to the enforced shortening.

Another possible drawback of the trimmed context is the loss of general context. In some edge cases, the part being trimmed off might contain very relevant information despite not containing the actual answer itself. For example, in some contexts, the actual subjects are only referred in the first sentence, and then continuously referred as a pronoun in the following sentences. In the unfortunate case that the first sentence gets trimmed off, the model might face great challenges in decoding the actual subjects to map to the query.

6 Conclusion

The proposed Read + Verify system achieves moderate improvement on BERT baseline model. The snipped context helps the self-attention layers to be more concentrated on the relevant part of the context, but sometimes also artificially boosts the non null probability and leads to mistakes in unanswerable questions. Given the strength of the model in centralizing the attention probability, one interesting exploration is to apply this model on datasets with longer contexts, since these datasets very likely suffer from the disperse attention distribution.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, page arXiv:1810.04805, Oct 2018.

[2] Minghao Hu, Furu Wei, Yuxing Peng, Zhen Huang, Nan Yang, and Dongsheng Li. Read + Verify: Machine Reading Comprehension with Unanswerable Questions. *arXiv e-prints*, page arXiv:1808.05759, Aug 2018.

[3] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250, Jun 2016.

[4] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional Attention Flow for Machine Comprehension. *arXiv e-prints*, page arXiv:1611.01603, Nov 2016.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv e-prints*, page arXiv:1706.03762, Jun 2017.