A Literature Review on BERT, RoBERTa and T5 Models

Group Members: Isha Hemant Arora; Amulya Gupta Vangapalli

1. Abstract

Transfer learning's effectiveness in NLP comes from pre-training a model on abundantly available unlabelled text data with a self-supervised task, such as language modelling or filling in missing words. After pre-training, the model can be fine-tuned on smaller labelled datasets, often resulting in better performance than training on the labelled data alone.

GPT, BERT, XLNet, RoBERTa, ALBERT, and Reformer are examples of successful transfer learning models. Because of the rapid pace of advancement in the industry, determining which advancements are the most significant has proven difficult (e.g., what type of pre-training is the best).

BERT and RoBERTa are encoder models built on top of the Transformer and T5 is an encoder-decoder model with a baseline transformer model.

We discuss the models BERT, RoBERTa and T5 and discuss their implementations. Unlike other transformer models, these are bidirectional and provide better efficiency. The model results over GLUE, SQuAD v1.1 , SQuAD v2.0, SWAG and RACE. The models are discussed in detail and the improvements are reported. A comparison of the models is also discussed.

2. Introduction

Unlike the traditional models which are unidirectional, BERT aims to pretrain bidirectional representations i.e., it joins both left to right and right to left in all context layers. Consequently, only one additional output layer is required to finetune the model. It can be used for a wide range of tasks like question answering, language inference etc. This model has been applied on 11 state-of-the-art NLP tasks and increased the performance consistently.

RoBERTa is a model which enhances BERT because BERT was found to be significantly undertrained and had the potential to match or even exceed the performance of all models introduced later.

With T5, all NLP tasks are reframed into a unified text-to-text format where the input and output are always text strings.

3. Models

BERT

For applying pre-trained language representations to downstream activities, there are two options one of which is feature based and the other is fine-tuning. Since the traditional models are unidirectional, it restricts the models that can be used while pretraining. In this case since BERT uses a “masked language model”, the above-mentioned limitation is solved. The main idea behind the masked language model is to predict the vocabulary only based on the context. To make this more appropriate, a few random tokens from the input are masked. To solve the problem of unidirectionality, the MLM model combines the left and the right context giving way to pretrain a deep bidirectional transformer. BERT also includes another task of predicting the next sentences which jointly pretrains text-pair representations.

BERT's cohesive architecture across tasks is one of its most distinguishing features. The pre-trained design and the final downstream architecture are nearly identical.

BERT is basically built on top of a Transformer encoder model and is multi-layered and bi-directional. The model is tested in 2 different model sizes :

1. $BERT_{BASE}$ (L=12, H=768, A=12, Total Parameters=110M)
2. $BERT_{LARGE}$ (L=24, H=1024, A=16, Total Parameters=340M)

Where L = number of layers H=hidden size A=no of self-attention heads. BERTBASE has the same size as that of GPT just to compare results.

The input representation can represent both a single sentence or a pair of sentences in a single token sequence. Pairs of sentences are held together in a single sequence. The sentences are segregated in the following way. Firstly, a special token <SEP> is used to separate the sentences and then a word embedding is added to each token to identify which sentence it belongs to. For the embeddings, the WordPiece embeddings with 30000 vocabulary is used.

The input representation of a token is built by adding the corresponding token, segment, and position embeddings.

BERT includes 2 steps : pretraining and fine-tuning. In the pretraining task, the model is trained on unlabelled data from various pre training tasks. In case of fine-tuning, the pre-trained parameters are initialized first, and all their parameters are fine-tuned using labelled data.

Each downstream task has separate fine-tuned models, even though they are initialized with the same parameters.

TASK 1 : PRETRAINING BERT

For the pretraining , the BooksCorpus and English Wikipedia with 800M and 2500M words are used. From the English Wikipedia corpus, the data from tables, lists and headers is not considered.

The traditional left-to-right or right-to-left models aren't used for the pretraining. As mentioned above, a bidirectional model is used, and it is done using 2 unsupervised tasks.

1. MASKED LANGUAGE MODEL

   The MLM goal is to forecast the masked tokens with a cross-entropy loss.
   To train a deep bidirectional representation, some percentage of random input tokens are taken and masked, and it's then required to predict those masked tokens. This is called Masked Language Model or Cloze in some contexts. The final hidden vectors which correspond to the masked tokens are then sent into an output softmax. For the masking, an average of 15% was taken from the input vectors.

   Though this solves the problem of bidirectionality it introduces a new issue of mismatched pretraining and fine-tuning since while fine-tuning, the mask isn't available. In order to solve this problem, the masked words are not always replaced with the mask token.

2. NEXT SENTENCE PREDICTION

   NSP is a binary classification loss that predicts if two segments in the original text will follow each other. Consecutive sentences from the text corpus are used to construct positive examples. Negative examples are made by combining sections from other documents. With the same probability, both positive and negative samples are sampled.

   We pre-train for a binarized next sentence prediction challenge that can be easily produced from any monolingual corpus to train a model that understands sentence relationships.

   For the pair of sentences, for 50% of the time, the second sentence is the sentence that follows the first one. And for the rest of the times, the second sentence is a random one.

<u>TASK 2 : FINE-TUNING</u>

Generally, when we come across text pairs, they are independently encoded before applying bidirectional cross attention. This is different in case of BERT; it uses self-attention mechanism to combine both the stages. For each of the tasks, the task-specific inputs and outputs are sent into BERT and all the parameters are fine tuned.

Pre-training sentences A and B are equivalent to (1) sentence pairs in paraphrase, (2) hypothesis-premise pairs in entailment, (3) question-passage pairs in question answering, and a degenerate text-pair in text categorization or sequence tagging at the input. The [CLS] representation is fed into an output layer for classification, such as entailment or sentiment analysis, and the token representations are fed into an output layer for token level tasks, such as sequence tagging or question answering.

RoBERTa

RoBERTa is an enhancement to the BERT model and the following modifications are applied

1.modelling the data with more data and with larger batches

2.excluding next sentence prediction

3. using longer sequences for training

4. dynamic changing of the masking pattern

TASK 1

BERT is optimized using the parameters : $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\varrho = 1e\text{-}6$ and L2 weight decay of 0.01. The learning rate is warmed up over the first 10,000 steps to a peak value of 1e-4, and then is linearly decayed. RoBERTa also follows the same approach but with little tweaks in the peak learning rate and number of warmup steps. While implementing the model, it was found that it is extremely sensitive to the Adam epsilon term and hence it was fine-tuned. Also, a $\beta 2$ value of 0.98 was applied to enhance the stability when training was performed with large batch sizes. The model only uses full-length sequences alone.

For the training data, 5 English language corpora with varying sizes and domains adding upto 160 GB of uncompressed data has been used.

1. BOOKCORPUS plus English WIKIPEDIA - The original dataset used to train BERT – 16GB
2. CC-NEWS collected from the English portion of the CommonCrawl News dataset. It contains news articles in English crawled between September 2016 and February 2019. It contains 63 million articles summing up to 76 GB after filtering.
3. OPENWEBTEXT – an open source recreation of the WebText corpus. The text is basically content extracted from Reddit with at least 3 upvotes – 38GB
4. STORIES – a subset of CommonCrawl filtered to match the story like style of Winograd schemas – 31GB

TASK 2

The original BERT model consists of this major aspect Next Sentence Prediction. After a lot of analysis, it was found that NSP is not a necessary task in modelling. Many training formats were compared before reaching this conclusion.

SEGMENT-PAIR+NSP

This is the formal of the original BERT model with NSP loss function. Each input has a pair of segments which may contain multiple natural sentences, but the combined length needs to be less than 512 tokens.

SENTENCE-PAIR+NSP

The input contains pair of natural sentences either extracted from a portion of a document or from separate documents. These are significantly shorter than 512 tokens, hence the batch size is increased to make the number of tokens similar to the original BERT model. The NSP loss function is retained.

FULL-SENTENCES

Each input is jam-packed with whole phrases from one or more documents, with a total length of no more than 512 tokens. It's possible that inputs will violate document boundaries. We start sampling phrases from the next document when we reach the end of the previous one, and we add an extra separator token between them. We get rid of the NSP loss.

DOC-SENTENCES

These are structured similar to the full sentences except the fact that these do not cross the boundaries. When inputs are sampled nearer to the end of the document, it may be smaller than 512 tokens and hence the batch sizes are dynamically increased to have similar tokens as that of full sentences.

After comparing all these formats, it was found that getting sentences from a single document performs better than that of getting it from multiple documents. Also, since using DOC_SENTENCES gives varied batch sizes of results, FULLSENTENCES are used.

TASK 3

We train BERT with even larger batch sizes, up to 32K sequences.

The original BERT implementation used a 30K character-level BPE vocabulary that is learned after the input is pre-processed with heuristic tokenization techniques. RoBERTa uses a bigger BPE which contains 50k sub words without any further pre-processing or tokenization. This results in additional 15M and 20M parameters for $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$ respectively.

TASK 4

One of the major tasks in the original BERT model does is the masking of tokens. The model masks once during the data pre-processing which results in a single static mask. To overcome this, the training was duplicated so times so that each of the sequence gets a different mask over the 40 training epochs. This means that each different sequence gets 4 different masks during the training. Dynamic masking is also performed for comparison purposes. Dynamic masking means a mask is generated every time a sequence is fed into the model. This becomes crucial when pretraining for more steps or with larger datasets.

T5 (Text-to-Text Transfer Transformer)

MODEL

The architecture of the model follows the form of input sequence of tokens mapped to a sequence of embeddings (in turn passed into the encoder). These embeddings used are relative position embeddings rather than the fixed embeddings as used in the original Transformer model. The encoder consists of the two subcomponents, a self-attention layer followed by a small feed-forward network. To these subcomponents, a simplified version of layer normalization with rescaled activations and having no additive bias is applied. Following this, a residual skip connection is used to add each subcomponent's input to its output. Dropout is applied within the feed-forward network on the skip connection, attention weights and the input and output of the entire stack. The decoder segment of the model is similar in structure to the encoder. The only difference is that it includes a standard attention mechanism after each self-attention layer that attends to the output of the encoder. The self-attention mechanism in the decoder also uses a form of the autoregressive (causal) self-attention which makes sure that the model only attends to past outputs. The output of the final decoder block is fed into a dense layer with a softmax output whose weights are shared with the input embedding matrix. All the attention mechanisms are split up into independent heads whose outputs are concentrated before being further processed.

Thus, the major differences between the original Transformer and the one proposed is, removing the Layer Norm bias, placing the layer normalization outside the residual path, and using a different position embedding scheme.

THE DATASET

The paper takes reference of the fact that major previous work on transfer learning for NLP has been done on large unlabeled datasets for unsupervised learning. To generate a dataset as required, "Common Crawl" is used as a source of text scraped from the web. This is a publicly available web archive that provides "web-extracted text" by removing markup and other non-text content from scraped HTML files and has been used as a source of text data for NLP in historical model generation too. This data, although enormous and highly diverse, is a lot of gibberish and includes boiler-plate text (low-quality data) and thus goes under intensive cleaning making sure to focus on and include only English language pages. Since the idea is to fine-tune the model on machine translations for English to German, French and Romanian, the pages from Common-Crawl scraped also include those in these languages.

A dataset of 750 GB is created and dubbed the "Colossal Clean Crawled Corpus" (C4) and released as a part of the TensorFlow datasets. The idea of creating and using this immensely huge **unsupervised** dataset is to not only avoid overfitting, but also make sure that the model is able to be implemented for generic language understanding tasks.

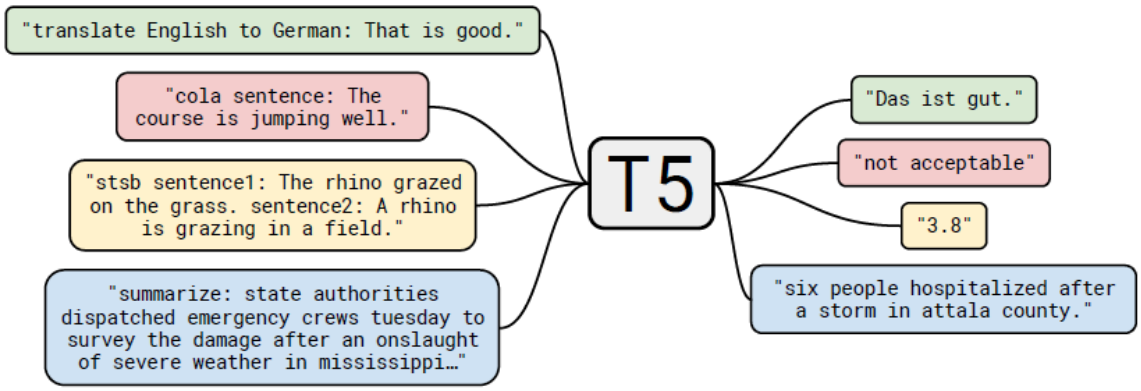

Fig: Text-to-text framework used

As a focus to measuring general language learning abilities, performance of multiple downstream tasks is studied (as on data from the TensorFlow Datasets). The datasets are distributed as by the GLUE (General Language Understanding Evaluation) and SuperGLUE benchmarks.

UNDERSTANDING T5

As understood from the name itself, the idea is to make sure to cast all tasks considered into "text-to-text" format (task where model is fed some text for content or conditioning and produces some output text). The framework of T5 provides a consistent training objective for both, pre-training and fine-tuning. The model is trained with a maximum likelihood objective regardless of the task. The task that should be performed is specified by adding as task-specific (text-based) prefix to the original input sequence before feeding it to the model.

The "text-to-text" framework used is an extension of previous work that casts multiple NLP tasks into a common format (referencing the Natural Language Decathlon). As a change from the Natural Language Decathlon, the model proposed stipulates allowing for separately fine-tuning the model on each individual task and use short task prefixes instead of an explicit question-answer format (the hyperparameter of the model).

The framework proposed also allows for generative tasks like machine translation and abstractive summarization where it is not possible to enumerate all possible output choices. The authors were, for the most part, able to cast all tasks to a text-to-text format except a regression task (STS-B). The workaround that the authors employed was to convert the regression problem as a 21-classification problem (a single word corresponding to the target label is predicted).

As the paper focuses on studying historical contributions in a systematic format, by taking a reasonable baseline, and altering one aspect of the setup at a time. While the authors were unable to perform a combinatorial exploration of all the factors considered in the study (as it would be computationally expensive), they hope to be able to do so as a part of their future work. The authors even tried to test model architectures similar in spirit to the BERT model for text classification tasks (as none of them are identical to the encoder model that BERT uses).

1. Standard Encoder-Decoder Transformer: The encoder is similar to the BERT model and the decoder to the GPT style causal manner. All the tasks are formulated as text-to-text tasks and all the tasks are trained using standard maximum likelihood (teacher forcing and cross-entropy loss). AdaFactor is used for optimization. Greedy decoding is used during the test time.
2. Language Model (LM): Used as the causal attention mechanism as an autoregressive modeling approach.
3. Prefix-LM approach: This is a combination of the BERT and Language Model approaches. The main difference between the Prefix LM and the BERT architecture is that the classifier is integrated into the output layer of the Transformer decoder in the former.

Denoising objectives have known to produce better performance and have, as a result quickly become the standard. It includes masking a sequence of words from the sentence and training the model to predict the masked words. It was noticed that most denoising objectives which train the model to reconstruct randomly missing or corrupted text, performed similarly in the "text-to-text" framework. As a result, objectives such as fill-in-the-blank-style denoising objectives are offered as a way to make unsupervised pre-training more computationally efficient.

Comparing these models, on the multiple architectures with the denoising objective, it was seen that the best results were seen with the Encoder-Decoder approach.

Architecture	**GLUE**	**CNNDM**	**SQuAD**
Encoder-Decoder	***83.28***	***19.24***	***80.88***
Language Model	74.70	17.93	61.14
Prefix LM	81.82	18.61	78.94

It can also been seen that including the BERT-style model to the Language Model also makes a good amount of difference. The unsupervised objective is another pre-training objective considered by the authors and have explored the practices of:

1. Prefix Language Modeling: Predicting the next word in the sentence considering all the words preceding that word.
2. Deshuffling: Words of a sentence are shuffled, and the model is trained to predict the original text.
3. Corrupting Spans: The denoising objective

Objective	**GLUE**	**CNNDM**	**SQuAD**
Prefix Language Modeling	80.69	18.94	77.99
Deshuffling	73.17	18.59	67.61
Corrupting Spans (Denoising): BERT-style	***82.96***	***19.17***	***80.65***

The denoising objective had the most promising results.

Overall, T5 was first pre-trained on the C4 dataset (unsupervised dataset) with focus on the denoising (corrupting spans) objective with an Encoder-Decoder architecture (as chosen from the above results). This model is then fine-tuned on the downstream tasks with a supervised objective with the appropriate input modeling for the "text-to-text" framework.

RESULTS

T5 has achieved the state of the art in many GLUE, SuperGLUE tasks along with translation and summarization benchmarks.

The authors also experimented with various approaches for training the model on multiple tasks at once (mixing examples from different datasets when constructing the batches) and setting the proportion of each task toi train on. While the authors were unable to find a concrete strategy for setting mixing proportions such that it matched the performance of the basic approach (unsupervised pre-training followed by supervised fine-tuning), but it was seen that if fine-tuning was performed after pre-training on a mixture of tasks, performance comparable to that of unsupervised pre-training was achieved.

While it was a little obvious, a major conclusion that was achieved was that large models (that require extremely huge hardware) tended to perform better. It is noted that as the hardware used for running these models is continually getting cheaper and more powerful suggests that scaling up might be a promising way to achieve better performance. A benefit that transfer learning has is that it is possible to attain a good performance on low-resource tasks too.

T5 can be extended to two novel tasks: closed-book question answering and fill-in-the-blank text generation with variable-sized blanks.

Closed-book question answering is about reading comprehension problems where the model is fed some context along with the question is trained in a way that it can find the answer from the context.

To generate realistic text, T5 relies on a fill-in-the-blanks type task with which it is familiar during the pre-training phase. Thus, a new downstream task to accommodate for this was created, called "sized fill-in-the-blank" where the size for the number of words to be filled were provided and the words of the sentence, then predicted

4. Evaluation Results

The model BERT has been evaluated on GLUE, SQuAD v1.1, SQuAD v2.0 and SWAG.

MODEL	GLUE		SQuAD 1.1		SQuAD 2.0		SWAG
	MNLI	QNLI	EM	F1	EM	F1	Accuracy
$\mathbf{BERT_{BASE}}$	84.6	90.5	80.8	88.5	-	-	81.6
$\mathbf{BERT_{LARGE}}$	86.7	92.7	84.1	90.9	88.7	81.9	86.6

The model is examined by checking the performance of the model with and without the NSP task. After close inspection, it is found that removing NSP takes a toll on the performance. Also, a number of BERT models were examined by changing the number of layers, hidden units, and attention heads. It was observed that larger models have better accuracy in all the

four evaluation methods. A feature-based approach was also investigated along with the fine tuning approach and it was found that the model does well in both the techniques.

Basically, RoBERTa is trained with large mini batches, large BPE, FULL-SENTENCES with no NLP loss and dynamic masking. Though these modifications are simple they show a significant difference in the performance of RoBERTa which includes better score on GLUE, SQuAD, SQuAD V1.1 and RACE.

MODEL	GLUE		SQuAD 1.1		SQuAD 2.0		RACE
	MNLI	QNLI	EM	F1	EM	F1	Accuracy
BERT$_{LARGE}$	86.6	92.3	84.1	90.9	79.0	81.8	72.0
RoBERTa	90.2	94.7	88.9	94.6	86.5	89.4	83.2

5. Extensions

BERT and RoBERTa are both based on Transformers. Generally, they can be formulated as the following equations

$$\hat{h}^l = \mathrm{LN}(h^{l-1} + \mathrm{MHAtt}(h^{l-1}))$$
$$h^l = \mathrm{LN}(\hat{h}^l + \mathrm{FFN}(\hat{h}^l)),$$

h0 is the BERT/RoBERTa input representation, formed by the sum of token embeddings, position embeddings, and segment embeddings. MHAtt stands for multi-head self-attention; FFN has three levels, the first of which is a linear projection layer, followed by an activation layer, and finally another linear projection layer; l stands for the depth of Transformer layers. BERT and RoBERTa feature 12 and 24 Transformer layers, respectively, in their base and large versions.[1]

To assess the quality of the natural sentence generation, BERTScore is used.

$$R_{BERT} = \frac{1}{|x|} \sum_{x_j \in x} \max_{\hat{x}_j \in \hat{x}} x_i^T \hat{x}_j$$ [2]

The models BERT and RoBERTa can be used for a variety of tasks. We have examined papers on Measurement of Semantic Textual Similarity in Clinical Texts[3], Text-based emotion recognition[4], Covid analysis on twitter data[5] and have observed that in all of these, RoBERTa performs better than BERT and all other extensions of BERT with a 2-20% better performance.

As an extension to the T5 paper, the T5 model was trained to answer trivia questions in a more difficult “closed-book” setting, without any access to external knowledge. The only way the model can answer a question is to use knowledge stored in its parameters that it picked up during the unsupervised pre-training. It is considered as a constrained form of “open-domain question answering”.[6]

An application of T5 was seen in a Chatbot setting where the T5 model was to paraphrase the training set in order to augment it with further data. Seven transformer-based text classification algorithms (BERT, DistilBERT, RoBERTa, DistilRoBERTa, XLM, XLM-RoBERTa, and XLNet) are used after fine-tuning in the training data for two epochs. It was found that all

models were improved when using the training data augmented by the T5 model. The average increase in the classification accuracy was seen to be approximately 4.01%. The best result in these models were seen on that of the RoBERTa model with a 98.96% classification accuracy (which is huge!). The best performing Transformer model was then found using Logistic Regression of output label predictions. [7]

The Transformer Language Model T5 was also seen to be able to gain knowledge of the visual world without being able to see or feel (as from the statistical properties of the language). It was seen that T5 possesses an implicit understanding of attributes of physical beings (some more than the others, probably determined by the data used and any underlying implicit patterns in the data) and that on increasing the number of parameters, the fit to human judgements improved dramatically (thus suggesting that the difference between humans and these models might ultimately disappear).[8]

6. Comparisons

The original transformer had both an encoder and a decoder, but BERT only has an encoder. BERT is a transformer-based model because it employs an encoder that is extremely close to the original encoder of the transformer. As a result, BERT employs solely attention and feed-forward layers rather than recurrent connections. BERT also uses segment embeddings, while the original transformer only uses word embeddings and positional encodings. The T5, uses the baseline of the original transformer: the encoder-decoder structure and includes a BERT-style structure for the encoder and language modeling style decoder. This is the major difference that sets a BERT model and T5 truly apart. The output for a T5 model is not a label or a span of the input to the input sentence, but is, in fact a text-string.

While Prefix LM (as one of the models considered for the T5 architecture) can effectively be applied to both NLG and NLU tasks, the architecture is inherently more restricted than Seq2Seq architecture (it is required to have a text-to-text architecture) whereas the original BERT does not follow such kind of a Seq2Seq architecture (modifications can be made to the original BERT to include this constraint if necessary).

7. Citations

[1] Yang X, He X, Zhang H, Ma Y, Bian J, Wu Y : Measurement of Semantic Textual Similarity in Clinical Texts: Comparison of Transformer-Based Models

[2] Müller, M., Salathé, M., & Kummervold, P. E. (2020). Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. *arXiv preprint arXiv:2005.07503*.

[3] Koroteev, M. V. (2021). BERT: a review of applications in natural language processing and understanding. *arXiv preprint arXiv:2103.11943*. -bertscore

[4] A. F. Adoma, N. -M. Henry and W. Chen, "Comparative Analyses of Bert, Roberta, Distilbert, and Xlnet for Text-Based Emotion Recognition," 2020 17th International Computer

Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2020, pp. 117-121, doi: 10.1109/ICCWAMTIP51612.2020.9317379.

[5] Dai, J., Yan, H., Sun, T., Liu, P., & Qiu, X. (2021). Does syntax matter? a strong baseline for aspect-based sentiment analysis with roberta. *arXiv preprint arXiv:2104.04986*.

[6] Roberts, A., Raffel, C., & Shazeer, N.M. (2020). How Much Knowledge Can You Pack into the Parameters of a Language Model? *ArXiv, abs/2002.08910*.

[7] Bird, J. J., Ekárt, A., & Faria, D. R. (2021). Chatbot Interaction with Artificial Intelligence: human data augmentation with T5 and language transformer ensemble for text classification. *Journal of Ambient Intelligence and Humanized Computing*, 1-16.

[8] Shi, H., & Wolff, P. (2021). What Transformers Might Know About the Physical World: T5 and the Origins of Knowledge. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 43, No. 43).