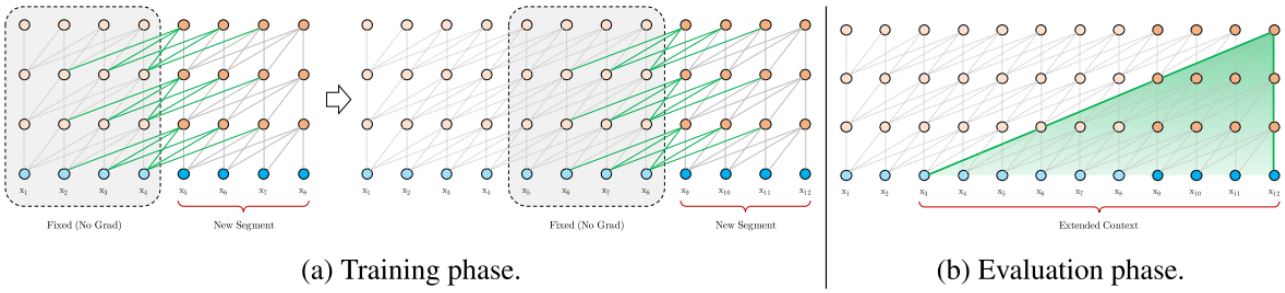
TransformerXL

Transformer is a pre-trained language model which follows an encoder-decoder arquitecture and uses an attention mechanism to handle dependencies instead of a RNN. Since RNN are not able to keep long-range dependencies between words while processing, this attention mechanism solves the problem. Transformers have a potential of learning longer-term dependency, but are limited by a fixed-length context in the setting of language modeling. Transformer-XL, a novel neural arquitecture enables learning dependency beyond a fixed length without disrupting temporal coherence. It consists of a segment-level recurrence mechanism and a novel positional encoding scheme. Our method not only enables capturing longer-term dependency, but also resolves the context fragmentation problem.

During training, vanilla language models don't make effective use of context information and segments are treated individually. In addition, semantic boundaries during segmentation are usually not respected since most methods employ standard chunked sequences of fixed lengths. During the evaluation, fixed-length contexts are used and segments are processed from scratch, which becomes expensive, even though context fragmentation is somewhat addressed. This paper aims to focus on the problem of efficiency by better modeling longer-term dependency.

In language modeling, Transformer networks are limited by a fixed-length context and thus can be improved through learning longer-term dependency. The paper proposes a novel method called Transformer-XL (meaning extra long) for language modeling, which enables a Transformer architecture to learn longer-term dependency — via a recurrence mechanism — beyond a fixed length without disrupting *temporal coherence*.

The method is different from other previous approaches that focus on other strategies to support long-term dependency such as *additional loss signals* and *augmented memory structure*.
A segment-level recurrent mechanism is introduced which enables the model to reuse previous hidden states at training time, addressing both the issues of fixed-length context and context fragmentation. In other words, the historical information can be reused and it can be extended up to as much as GPU memory allows.



TransformerXL [1]

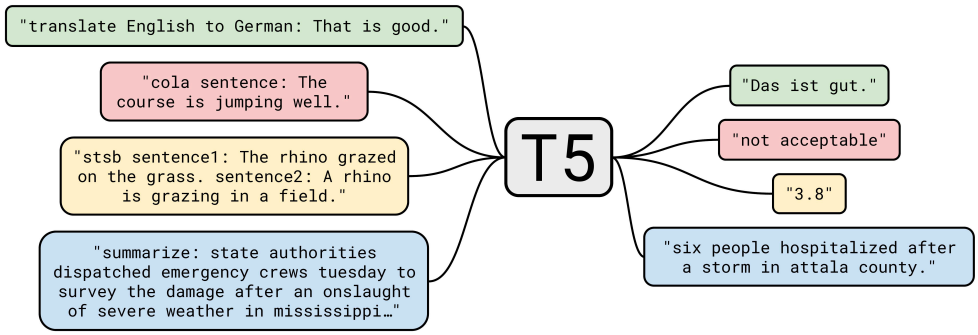
Google T5

T5, or **Text-to-Text Transfer Transformer**, is a Transformer based architecture that uses a text-to-text approach. Every task – including translation, question answering, and classification – is cast as feeding the model text as input and training it to generate some target text. This allows for the use of the same model, loss function, hyperparameters, etc. across our diverse set of tasks. The changes compared to BERT include:

- adding a *causal* decoder to the bidirectional architecture.
- replacing the fill-in-the-blank cloze task with a mix of alternative pre-training tasks.

With T5, we propose reframing all NLP tasks into a unified text-to-text-format where the input and output are always text strings, in contrast to BERT-style models that can only output either a class label or a span of the input. Our text-to-text framework allows us to use the same model, loss function, and hyperparameters on *any* NLP task, including machine translation, document summarization, question answering, and classification tasks (e.g., sentiment analysis). We can even apply T5 to regression tasks by training it to predict the string representation of a number instead of the number itself.

Consider the example of a BERT-style architecture that is pre-trained on a Masked LM and Next Sentence Prediction objective and then, fine-tuned on downstream tasks (for example predicting a class label in classification or the span of the input in QnA). Here, we separately fine-tune different instances of the pre-trained model on different downstream tasks.
The text-to-text framework on the contrary, suggests using the same model, same loss function, and the same hyperparameters on all the NLP tasks. In this approach, the inputs are modeled in such a way that the model shall recognize a task, and the output is simply the “text” version of the expected outcome. Refer to the above animation to get a clearer view of this.



[1] Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html