COMPSCI 199 - Supervised Independent Study

By: Hiva Mohammadzadeh*

1 Attention is all you need

1.1 Summmary

In this paper published in 2017, the authors introduce the Transformer model. The Transformer model replaces the traditional recurrent neural network (RNNs) and convolutional neural network (CNN) architectures used in Natural Language Processing (NLP) with an attention-based architecture that allows for parallel processing of inputs and outputs, eliminating the need for sequential processing. This results in faster training times and improved accuracy on NLP tasks.

In the past Recurrent Neural Network were used. The RNNs have an encoder (which given the input sequence and the previous hidden state, outputs the next hidden state) and a decoder (which given the hidden state and the previous word that was decoded, outputs the next word). Even though RNNs worked well in solving these NLP tasks, they had some problems including: their sequential nature made them very slow to train and difficult to parallelize, and they required the input sequences to be processed in a fixed order. Given these problems, it is very hard for the RNNs to learn the long range dependencies.

The Transformer architecture, addresses the problems of RNNs by using self-attention to allow the model to selectively attend to different parts of the input sequence, regardless of their position. This makes it easier to model long-term dependencies and parallelize training.

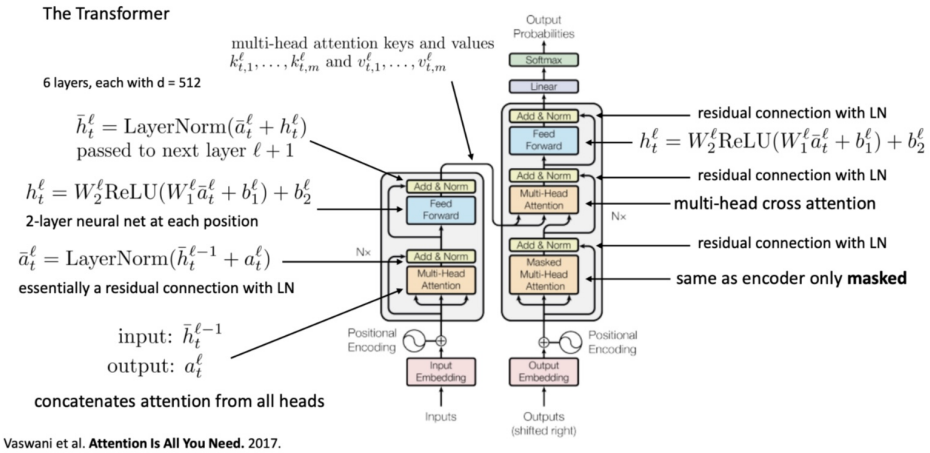


Figure 1: The Transformer architecture: consists of an encoder and a decoder, both of which are made up of a stack of identical layers. Each layer has a multi-head self-attention mechanism, and a position-wise fully connected feed-forward network. Figure from: Attention is all you need.

*4th year Electrical Engineering and Computer Sciences Undergraduate student.

Preprint. Under review.

The self-attention mechanism which is at the core of the Transformer, allows the model to selectively attend to different parts of the input sequence, without requiring positional information. This is done by computing an attention score between each element in the sequence and all other elements, and using these scores to weight the contributions of each element to the output of the layer. This done by using keys and values. The multi-head self-attention mechanism computes multiple attention scores in parallel, which allows the model to attend to different aspects of the input sequence at different levels of granularity. The encoder and decoder are connected by a third sub-layer, which uses masked multi-head self-attention to allow the decoder to attend only to the previously generated part of the output sequence during training. The previously generated part of the output is encoded into queries. The keys, values and queries are then passed through the third multi-head self-attention to produce the output.

1.2 Training Scheme

The Transformer model is trained using a supervised learning approach, which involves minimizing a loss function that measures the difference between the predicted outputs of the model and the true outputs of the training data. The loss function that is used during training of the transformer is the cross-entropy loss, which is used during pre-training to measure the difference between the predicted probability distribution over the vocabulary and the true distribution.

1.3 Contributions and Experiments

The authors conducted several experiments to show the performance of the Transformer Architecture and their significant contribution to the field.

They conducted experiments on two machine translation tasks (VMT 2014 English-to-German, and English-to-French) and showed that the model outperformed all the previous models. They achieved state-of-the-art performance and did it in only a fraction of training cost of the previous models with recurrent and convolutional layers. In VMT 2014 English-to-German task, they outperformed by more than 2.0 BLEU score and it only took 3.5 days. In English-to-French task, they decreased the training cost by 1/4 and increased the BLEU score as well.

The authors trained Transformer models on the One Billion Word Benchmark dataset and showed that the models achieved state-of-the-art performance in terms of perplexity. They also showed that the Transformer was able to capture longer-term dependencies in the data compared to previous models.

They also evaluated Transformer on English Constituency Parsing and even though they did run into a lot of challenges due to the lack of task specific tuning, the model still outperformed the previous models.

Results of these experiments show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. This is why a lot of the language modeling tasks and models today are Transformer based models.

The Github Repo: https://github.com/jadore801120/attention-is-all-you-need-pytorch

1.4 Related Work

This paper builds on a lot of **past research**:

The paper that sparked some of the first ideas was the papers: "Sequence to sequence learning with neural networks" published in 2014.

These paper introduced the encoder-decoder architecture for Neural Machine Translation (NMT). The authors proposed a model (seq2seq) consisting of two neural networks: an encoder that reads in the input sentence(source) and produces a fixed-length vector representation, and a decoder that uses this representation to generate the output sentence(target) one word at a time. During training, the decoder receives the previous word in the target sentence as input and uses its hidden state and the encoder representation to generate the next word. The authors also introduce several innovations to the seq2seq model to improve its performance, including the use of bidirectional RNNs in the encoder to capture both past and future context, and the incorporation of attention mechanisms to allow the decoder to selectively focus on different parts of the input sentence at each step. But a

major limitation of this model is that it relies on a fixed-length representation of the input sentence, which can be difficult to learn for longer sentences. Also, RNNs have several drawbacks, including difficulty in parallelization and the tendency to forget information from earlier in the sequence.

This lead to the paper "Neural machine translation by jointly learning to align and translate paper" published in 2015. This paper introduced the idea of attention mechanisms, which allow the decoder to selectively focus on different parts of the input sentence at each step of the decoding process. The attention mechanism allows the model to focus on different parts of the source sentence at each decoding step, which is particularly useful for translating long sentences or handling complex word order differences between languages. The attention mechanism is learned jointly with the translation model, which allows the model to learn to align the source and target sentences automatically during training. This paper presented experimental results on several machine translation datasets, demonstrating that the attention-based model outperformed previous state-of-the-art approaches, including the RNN-based seq2seq model. This idea led to the paper we discussed before and the development of Transformers.

The paper "Attention all you need" has also sparked a lot **further research** in the field of NLP:

One of the main papers is BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding published in 2018. BERT is a model designed to pre-train large neural networks on massive amounts of text data in an unsupervised fashion, and then fine-tune the resulting model on specific NLP tasks. The original BERT model was trained on a large corpus of text data from Wikipedia and the BookCorpus dataset, and achieved state-of-the-art performance on several NLP tasks. Since its introduction, BERT has become a widely used model in NLP and has inspired a lot of models, such as RoBERTa, ELECTRA, and ALBERT.

2 Decoder only vs. Encoder only vs. Encoder-Decoder Architectures

1. **Encoder only**: In this architecture, the input is fed into an encoder, which processes the input and generates a fixed-size representation (also known as a latent vector) that summarizes the input's salient features. The encoder does not generate an output; instead, its output is used as input for some downstream task. For example, in image classification, the encoder extracts relevant features from an image, and these features are then used by a classifier to predict the image's class. Encoders are bidirectional. BERT is an Encoder only architecture. It outputs all the information at once.

   Has a fixed-size output.

   Used for learning representations of input data that can be used for downstream tasks such as classification and clustering. They are particularly useful when labeled data is limited or unavailable, as they can be trained in an unsupervised manner.

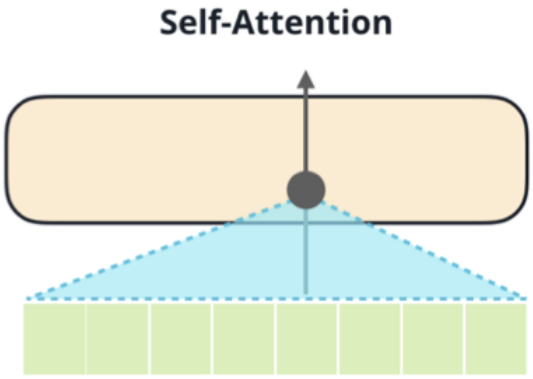


Figure 2: The Encoder's self attention visualization. Figure from: CS288 Lecture slides.

2. **Decoder only**: In this architecture, the decoder takes a fixed-size input vector (latent vector) and generates an output sequence. This architecture is commonly used in language generation tasks like machine translation, where the decoder generates a sequence of words given an encoded representation of a sentence. Decoders are unidirectional which means they either look at the input tokens or the tokens after it while predicting the current token. GPT is a Decoder only architecture and is an auto regressive inference model so it has to iteratively generate one token at a time.

   Has a variable length output sequences.

   Used when we want to generate new data that is similar to the input data, or when the input data distribution is not well-defined.

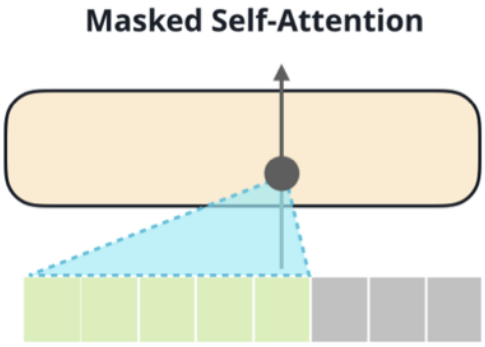


Figure 3: The Decoder's Masked self attention visualization. Figure from: CS288 Lecture slides.

3. **Encoder-Decoder**: An encoder-decoder architecture combines both the encoder and decoder to generate an output sequence from an input sequence. In this architecture, the encoder processes the input sequence and generates a fixed-size representation of the input sequence, which is then passed to the decoder. The decoder takes the encoder's representation and generates an output sequence. This architecture is commonly used in machine translation and image captioning. Sequence to sequence models.

   Used when mapping the input data to output data that is of a different length or different meanings. They can learn to align the input and output sequences in a meaningful way.

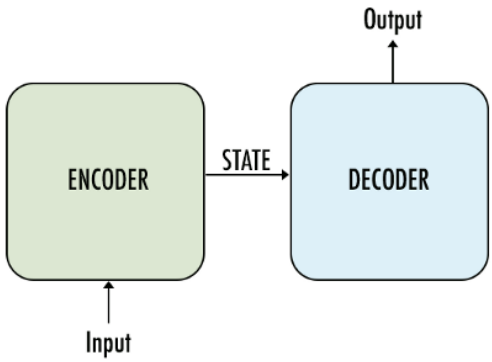
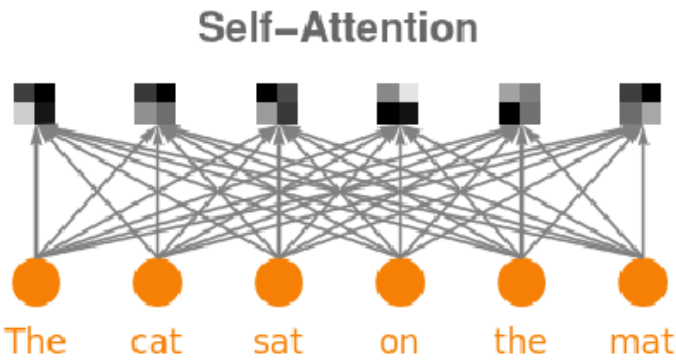


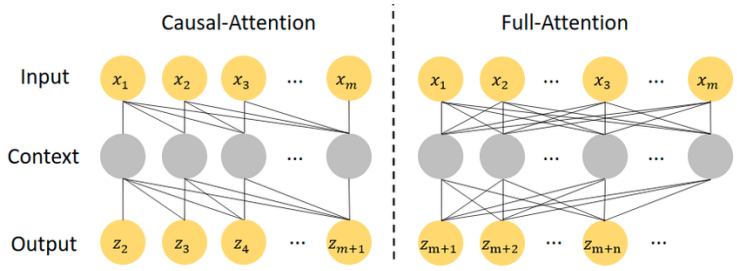
Figure 4: The Encoder Decoder visualization.

3 Forward attention vs Causal attention vs Triangle attention

1. **Forward Attention(Self-attention:** In forward attention, each query attends to all the keys and values in the input sequence up to its current position. This means that each query can access information from the past and present, but not the future. Forward attention is used in language modeling, where the goal is to predict the next token in a sequence given the previous tokens.



2. **Causal Attention:** Causal attention, also known as autoregressive attention, is similar to forward attention but also includes a mask that prevents each query from attending to any keys and values that come after its current position. This means that each query can only access information from the past and present, but not the future. Causal attention is used in tasks such as text generation, where the goal is to generate a sequence one token at a time.



3. **Triangle Attention:** Triangle attention is a variant of self-attention in which each query only attends to a subset of the keys and values in the input sequence, forming a triangular pattern. The triangular pattern is formed by setting a maximum distance between the query and key positions, beyond which the attention is not allowed to flow. Triangle attention is used to reduce the computational cost of attention and improve the ability of the model to capture long-term dependencies in the input sequence as it encourages the model to attend to positions that are further away.

4 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

4.1 Summmary

Bidirectional Encoder Representations from Transformers (BERT) is a pre-trained language model that uses a transformer-based architecture and is trained using unsupervised learning on a large corpus of text data. It is designed to pre-train deep Bidirectional Representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained model can be fine-tuned with just one additional output layer to achieve state-of-the-art performance on multiple natural language processing tasks. It uses a deep bidirectional Transformer architecture that

allows the model to attend to both the left and right context of each word, and capture long-range dependencies in the input text.

BERT pre-trains a deep bidirectional Transformer encoder on large amounts of unlabeled text using Masked Laguage Modeling(MLM), where a certain percentage of tokens are randomly masked in the input text, and the model is trained to predict the masked tokens based on the surrounding context. It also trains on a next sentence prediction task, where the model is trained to predict whether two input sentences are consecutive in the original text or not. This helps the model to better capture sentence-level relationships and coherence. Masked Language Model (MLM) and Next Sentence Prediction (NSP), which helped the model learn contextualized word embeddings and sentence-level understanding.

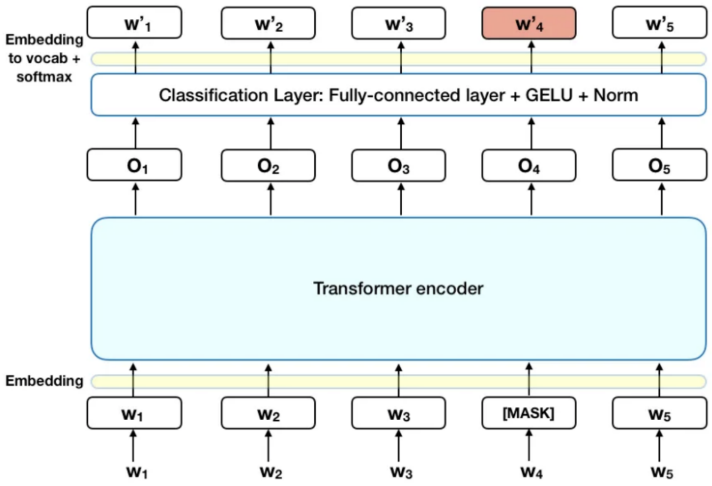


Figure 5: Structure of the BERT model.

4.2 Training Scheme

BERT is pre-trained on large amounts of unlabeled text, and then fine-tuned on downstream natural language processing tasks such as question answering, text classification, and named entity recognition. Fine-tuning BERT on these tasks leads to significant performance improvements over previous state-of-the-art models.

BERT is pre-trained in two stages. In the first stage, called pre-training on the Masked Language Model (MLM) objective where a certain percentage of the tokens in each input sequence are randomly masked. The model is then trained to predict the masked tokens based on the surrounding context. The MLM objective allows the model to learn contextualized word embeddings, as it requires the model to understand the context in which a word appears in a sentence to accurately predict the masked token. This pre-training objective has been shown to be effective in capturing deeper semantic understanding of language, improving the performance of the BERT model on various downstream NLP tasks.

In the second stage, called pre-training on the Next Sentence Prediction (NSP) objective, the model is trained to predict whether two input sentences are consecutive or not. It is designed to help the model understand the relationship between two sentences in a document. The NSP objective helps the model capture sentence-level semantics and improve its ability to handle tasks that involve multiple sentences, such as question answering or document classification. it has been shown to contribute to the model's ability to handle complex NLP tasks that involve multiple sentences. Overall, the NSP objective is a key component of the BERT model's pre-training process and helps to improve its state-of-the-art performance on various natural language processing tasks.

After pre-training, the BERT model can be fine-tuned on specific downstream tasks, such as text classification or question answering, by adding task-specific layers to the pre-trained model and fine-

tuning the entire model on the task-specific data. For fine-tuning, we follow three steps: adapting the input representation, adding a task-specific output layer, and training the model on the task-specific dataset. The pre-trained BERT model is fine-tuned on a task-specific dataset, with the weights of the pre-trained layers frozen, and only the task-specific output layer getting updated. The authors found that fine-tuning BERT resulted in significant improvements in performance on these downstream NLP tasks, achieving state-of-the-art performance on several benchmark datasets.

4.3 Contributions and Experiments

The main contribution of the paper is the introduction of a new pre-training objective called masked language modeling (MLM), which allows the model to learn bidirectional representations of text. during pre-training, 15% of the tokens in a sentence are randomly selected and masked, and the model is trained to predict the masked tokens based on the context of the surrounding tokens.

BERT also uses the next sentence prediction (NSP) loss during pre-training. The NSP objective is to predict whether two sentences follow each other in the original text or not. This task helps the model learn to understand the relationship between sentences and to capture the global context of a document.

The authors reported several experiments to evaluate the performance of the BERT model on a wide range of NLP tasks.

The paper evaluated the effectiveness of the MLM pre-training objective by comparing the performance of BERT models trained with and without MLM. The results showed that MLM significantly improved the quality of the learned representations.

The paper also evaluated the effectiveness of the NSP pre-training objective by comparing the performance of BERT models trained with and without NSP. The results showed that NSP also contributed to the model's ability to understand sentence-level semantics.

The paper also compared the performance of BERT with several existing models on several benchmark NLP tasks, including question answering, text classification, and named entity recognition. The results showed that BERT outperformed existing models on most of these tasks.

The paper evaluated the effectiveness of fine-tuning BERT on various downstream NLP tasks, including sentiment analysis, question answering, and named entity recognition. The results showed that fine-tuning on these tasks further improved the performance of the model.

5 RoBERTa: A Robustly Optimized BERT Pretraining Approach

5.1 Summmary

This paper proposes an optimized version of BERT, called RoBERTa. The authors introduce several key enhancements to BERT, including a dynamic masking strategy for pre-training, larger scale pretraining, longer training time, and optimization strategies such as larger batch size and dynamic learning rate. The paper also evaluates RoBERTa on several benchmark natural language processing tasks, showing that it outperforms BERT on most of them. Overall, the paper presents a robust and optimized approach to pre-training language models that leads to better performance on various NLP tasks.

Overall, RoBERTa represents a significant improvement over BERT in terms of its pretraining process and performance on downstream NLP tasks. Its dynamic masking strategy, larger scale pretraining, longer training time, and additional training data all contribute to its improved performance.

5.2 Contributions and Experiments

The Roberta paper has some contributions that have improved the performance of the BERT model.

Dynamic Masking: The authors propose a new masking strategy for BERT pre-training, called dynamic masking. Unlike the static masking approach used in BERT, dynamic masking randomly masks different tokens at different times during pre-training, rather than always masking the same tokens. The goal of it is to improve the model's ability to handle masked language modeling tasks like predicting masked words in a sentence.

Dynamic masking adds an additional step to this process. Instead of always masking the same percentage of tokens, the percentage of tokens that are masked is varied dynamically during training. Specifically, at each training step, a random percentage of the input tokens are selected for masking. The percentage of tokens selected for masking varies between a minimum and a maximum value, which are specified in advance.

It also forces the model to adapt to different degrees of masking, making it more robust to varying levels of input noise. By exposing the model to a wider range of masked inputs during pre-training, it is better able to generalize to unseen masked inputs during fine-tuning.

Large Scale Pre-training: The authors trained RoBERTa on a large corpus of web pages, books, and other text sources, totaling 160GB of text. This is significantly larger than the corpus used to train BERT, which was 16GB.

Longer Training: The authors trained RoBERTa for longer than BERT, using a maximum sequence length of 512 tokens and training for 100 epochs.

Optimization Strategies: The authors used several optimization strategies to improve the training of RoBERTa, including a larger batch size, a dynamic learning rate, and gradient accumulation.

Evaluation Metrics: The authors evaluated RoBERTa on several benchmark natural language processing tasks, including question answering, text classification, and named entity recognition, and found that it outperformed BERT on most of them.

6 Language Models are Unsupervised Multitask Learners

6.1 Summmary

This paper introduces a new pretraining approach called GPT(Generative Pre-trained Transformer) for building language models that can perform multiple tasks without the need for task-specific architecture or fine-tuning. GPT is a decoder only and a unidirectional model which means that it generates output in an autoregressive fashion which means that it learns to predict the next word in a sequence of text, given the previous words in the sequence.

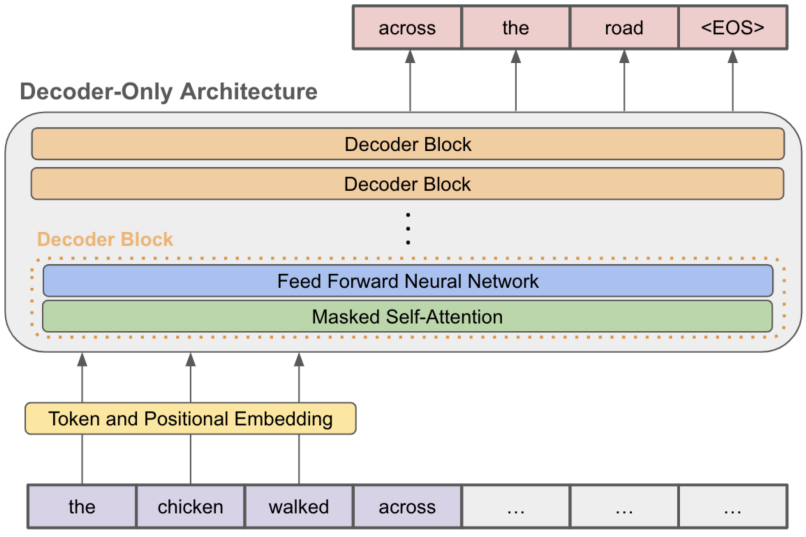


Figure 6: This figure shows structure of the GPT2 model.

This large transformer model is trained on a massive dataset of text in an unsupervised manner and is then fine-tuned on specific downstream tasks, such as language translation or text classification, with only a small amount of labeled data. It is trained on a massive dataset of web pages called WebText and is able to generate coherent and realistic text by predicting the next word in a sequence of words.

The authors also propose a novel pre-training objective called "masked language modeling," which involves randomly masking some tokens in the input sequence and training the model to predict the masked tokens based on the surrounding context.

6.2 Contributions and Experiments

The paper presents several contributions to the field and experiments to support those contributions:

An evaluation of the performance of the GPT language model on a range of NLP tasks, including the Stanford Question Answering Dataset (SQuAD), the General Language Understanding Evaluation (GLUE) benchmark, and several others.

An investigation of the impact of model size, training data size, and pre-training duration on the performance of the GPT language model.

An ablation study to assess the impact of different components of the GPT architecture on performance.

The paper also highlighted the importance of language understanding in natural language processing tasks. The authors demonstrated that the pre-trained GPT model can generate coherent and fluent text and can perform well on tasks that require understanding of the context and meaning of language.

7 Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

7.1 Summmary

This paper introduces a new architecture called text-to-text transformer(T5) which is trained on a divers range of NLP tasks. It can be fine tuned to perform a wide range of natural language processing (NLP) tasks, including text classification, question answering, summarization, and translation.

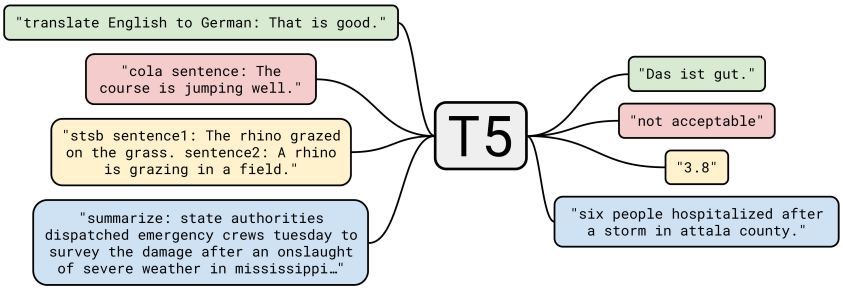


Figure 7: This figure shows the T5 architecture and all the tasks that is able to do at the same time. This architecture allows us to use the same model, loss function, hyperparameters, etc. across the different tasks. Figure from: T5 paper.

7.2 Contributions and Experiments

The paper presents several contributions to the field and experiments to support those contributions:

It introduces a new training objective called "pre-training tasks," where the T5 model is trained to predict a wide range of tasks from a single unified model. This training objective helps the model learn a more general representation of language that can be applied to many different tasks It is trained on a massive dataset called the Common Crawl, which consists of billions of web pages. The authors argue that this large-scale training leads to better performance on downstream tasks.

The authors show that the T5 model can achieve state-of-the-art performance on a wide range of NLP benchmarks, surpassing previous models that were specifically designed for each task. The T5 model achieves this by using a "text-to-text" framework, which involves framing all NLP tasks as text-to-text problems, where the model is trained to transform the input text into the output text.

The authors also conducted several experiments to analyze the behavior of the T5 model, such as studying the effect of model size on performance and the ability of the model to generalize to new tasks.

The structure of the T5 model is the same as of the original encoder-decoder transformer.

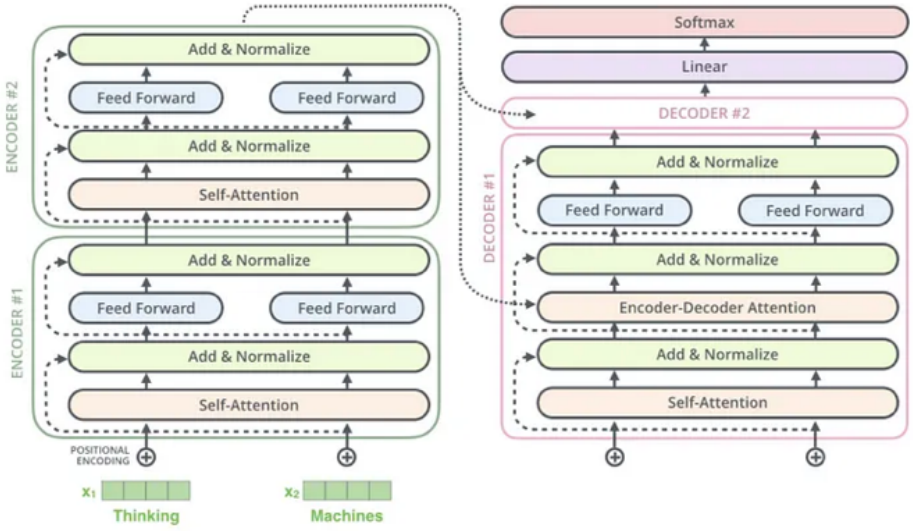


Figure 8: The encoder-decoder architecture used by the T5 model. Figure from: The Illustrated Transformer blog.

7.3 Training Scheme:

The model is pre trained on the "colossal clean crawled corpus." They leverage Common Crawl as a source of text scraped from the web. They change the data by removing any lines that didn't end in a terminal punctuation mark, removing line with the word javascript and any pages that had a curly bracket, and also deduplicate the dataset by taking a sliding window of 3 sentence chunks and deduplicate it so that only one of them appeared the dataset.

In the original text, some words are masked out with a unique sentinel token. Words are masked out independently uniformly at random. The model is trained to predict basically sentinel tokens to delineate the masked out text. This alows the model to be able to predict multiple words missing in the model and therefore filling in the blanks missing from the input.

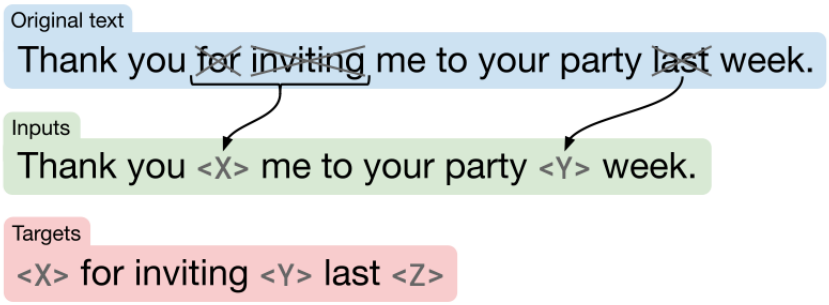


Figure 9: This figure shows the pre training procedure of the T5. Figure from: T5 Paper: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer .

In order to be able to use a single model on different tasks, the authors fine tune the model on multiple tasks. The do this by converting all the tasks into a text-to-text format. Then in order to incorporate different tasks, they use "Prefix Conditioning" where each task is specified using a text prefix that is prepended to the input text before feeding the text into the model. During training, the T5 model learns to generate output text that corresponds to the specified task, conditioned on the input text and the task prefix.

As an example given int the paper, to ask the model to translate the sentence “That is good.” from English to German, the model would be fed the sequence “translate English to German: That is good.” and would be trained to output “Das ist gut.”

8 Language Models are Few-Shot Learners

8.1 Summmary

The paper presents a series of experiments that demonstrate the remarkable few-shot learning capabilities of GPT-3. The model is able to generate high-quality text in a wide range of tasks, such as language translation, question answering, and even programming tasks, with little or no task-specific training data.

The GPT-3 model has 175 billion parameters which is 10x more than any previous non-sparse language models.

8.2 Contributions and Experiments

The paper shows the remarkable few-shot learning capabilities of GPT-3, which is able to generate high-quality text in a wide range of tasks with little or no task-specific training data. This helps the models learn and adapt to new tasks with minimal supervision.

The paper introduces a new benchmark dataset called SuperGLUE, which evaluates the performance of language models on a diverse set of natural language understanding tasks. The GPT-3 model achieves state-of-the-art performance on the SuperGLUE benchmark, surpassing the performance of previous state-of-the-art models by a significant margin.

The paper demonstrates that GPT-3 can perform zero-shot learning, which means that it can generate high-quality text in tasks it has not been explicitly trained on. This shows that the models can learn and generalize t new tasks.

Open-ended generation: The paper demonstrates that GPT-3 is capable of generating open-ended text, which means that it can generate text that is not constrained by a specific prompt or task. This represents a significant advance in the development of AI systems that can generate natural and coherent language. This helps the model be able to generate natural and coherent language.

The authors also show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-the-art fine- tuning approaches.

8.3 Training Scheme

The model that was pre-trained on a massive corpus of text data using unsupervised learning techniques. The pre-training process involved training the model on a diverse range of text sources, including books, articles, and web pages. It was trained to predict the next word in a given text sequence based on the preceding words.

First, the model was trained on a large corpus of text data using a transformer-based neural network architecture. The transformer architecture allows the model to learn long-term dependencies in the text data and capture contextual relationships between words. They then fine-tuned the model on several downstream tasks, such as language translation, question-answering, and sentiment analysis. This process helped the model to develop a more nuanced understanding of language and improve its ability to generate high-quality text.

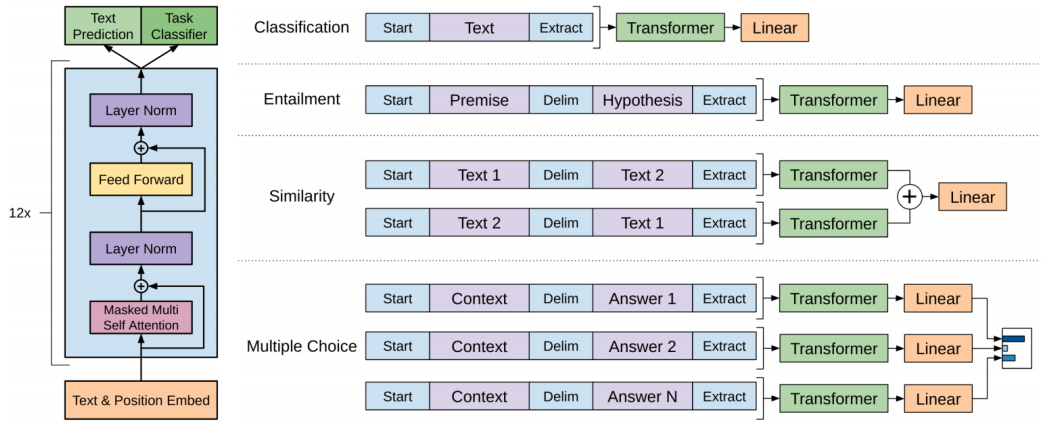
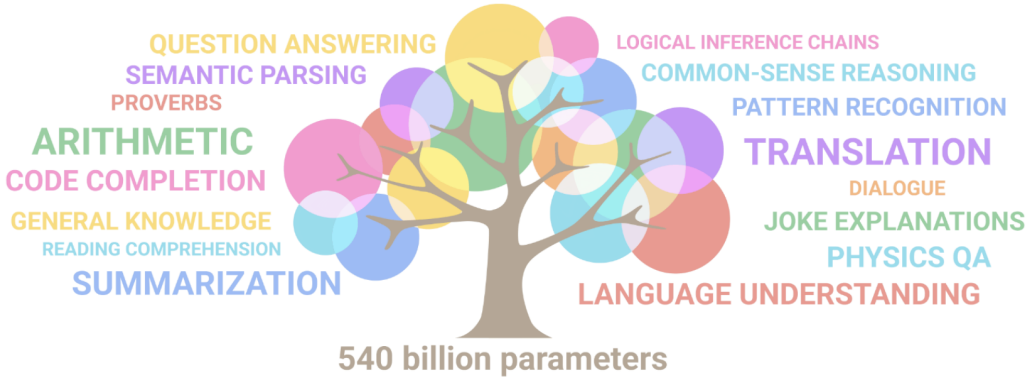


Figure 10: This figure shows structure of the GPT2 model. The main difference between the GPT2 model and GPT3 model is the number of layers. Figure from: GPT2 Paper: Language Models are Unsupervised Multitask Learners.

9 PaLM: Scaling Language Modeling with Pathways

9.1 Summmary

This paper proposes a novel language model architecture called Pathway-LM (PaLM) which is a 540-billion parameter, dense decoder-only Transformer model trained with the Pathways system, which enabled us to efficiently train a single model across multiple TPU v4 Pods. It improves the efficiency and effectiveness of language modeling tasks. PaLM uses a hierarchical architecture with multiple pathways that allow the model to capture different levels of linguistic information. The pathways can be trained independently, allowing for faster training and better scalability.



PaLM uses parallel layers to speed up training and multi-query attention to speed up inference.

The paper also introduces a new pre-training objective called MLM-Mix that improves the model's ability to handle out-of-domain data. The experimental results show that PaLM achieves state-of-the-art performance on various language modeling benchmarks while being more efficient than other large-scale language models.

We demonstrate continued benefits of scaling by achieving state-of-the-art few-shot learning results on hundreds of language understanding and generation benchmarks. On a number of these tasks,

PaLM 540B achieves breakthrough performance, outperforming the fine-tuned state-of-the-art on a suite of multi-step reasoning tasks, and outperforming average human performance on the recently released BIG-bench benchmark.

They also discuss the ethical considerations related to large language models and discuss potential mitigation strategies.

9.2 Contributions and Experiments

The PaLM architecture uses multiple pathways to capture different levels of linguistic information, allowing for faster training and better scalability.

It enabled Google to efficiently train a single model across multiple TPU v4 Pods1. PaLM demonstrates the first large-scale use of the Pathways system to scale training to 6144 chips, the largest TPU-based system configuration used for training to date1. PaLM also uses data parallelism at the Pod level across two Cloud TPU v4 Pods while using standard data and model parallelism within each Pod1. PaLM has been used to achieve state-of-the-art few-shot learning results on hundreds of language understanding and generation benchmarks2.

9.3 Training Scheme

PaLM is pre-trained using a novel scheme that jointly pre-trains an autoencoding and autoregressive language model on a large unlabeled corpus, specifically designed for generating new text conditioned on context. To force the encoder to comprehend the meaning of the given context, Masked Language Modeling (MLM) is added to pre-train the encoder. It is then fine-tuned on downstream comprehension-based generation tasks.

PaLM demonstrates the first large-scale use of the Pathways system to scale training to 6144 chips, the largest TPU-based system configuration used for training to date. The training is scaled using data parallelism at the Pod level across two Cloud TPU v4 Pods , while using standard data and model parallelism within each Pod4.

10 On the Opportunities and Risks of Foundation Models

10.1 Summmary

This explores the opportunities and risks associated with foundation models. The authors argue that foundation models have the potential to transform a wide range of industries and fields, from healthcare to education to finance.

They also note that foundation models raise several ethical concerns, such as the potential for bias and the concentration of power in the hands of a few large tech companies. The paper calls for increased transparency, accountability, and collaboration among researchers, policymakers, and industry stakeholders to ensure that foundation models are developed and deployed in a responsible and beneficial way.

10.2 Contributions and Experiments

It provides a comprehensive overview of foundation models, which are large-scale language models like GPT-3 that can perform a wide range of natural language processing tasks. It explores the opportunities and potential benefits of foundation models in various industries and fields, such as healthcare, education, and finance. It highlights the ethical concerns and risks associated with foundation models, such as the potential for bias, the concentration of power in the hands of a few large tech companies, and the potential impact on employment. It provides recommendations for researchers, policymakers, and industry stakeholders to ensure that foundation models are developed and deployed in a responsible and beneficial way, including increased transparency, accountability, and collaboration.

11 GPT-4 Technical Report

11.1 Summmary

This paper introduces GPT-4, a large-scale, multimodal model which can accept image and text inputs and produce text outputs. According to the report, GPT-4 exhibits human-level performance on various professional and academic benchmarks. It is a Transformer model pre-trained to predict the next token in a document, using both publicly available data and data licensed from third-party providers4. While less capable than humans in many real-world scenarios, GPT-4 exhibits human-level performance on various professional and academic benchmarks.

11.2 Contributions and Experiments

The authors introduce GPT-4 which is a language model that can handle more complex tasks than previous GPT models. It was developed to improve alignment and scalability for large models of its kind.

One of the key contributions of GPT-4 is that it is a large-scale, multimodal model that can accept image and text inputs and produce text outputs.

GPT-4 visual input example, Chicken Nugget Map:	
User	Can you explain this meme?
	
GPT-4	This meme is a joke that combines two unrelated things: pictures of the earth from space and chicken nuggets. The text of the meme suggests that the image below is a beautiful picture of the earth from space. However, the image is actually of chicken nuggets arranged to vaguely resemble a map of the world. The humor in this meme comes from the unexpected juxtaposition of the text and the image. The text sets up an expectation of a majestic image of the earth, but the image is actually something mundane and silly.

Table 18: Example prompt demonstrating GPT-4's visual input capability.

Figure 11: This an example of turning images into text by the GPT-4 from the paper.

The post-training alignment process results in improved performance on measures of factuality and adherence to desired behavior. While less capable than humans in many real-world scenarios, GPT-4 exhibits human-level performance on various professional and academic benchmarks, including passing a simulated bar exam with a score around the median of human test-takers.

It was trained on both publicly available data and data licensed from third parties.

GPT-4 is a significant improvement on GPT-3. It outperforms other models in English, and far outperforms it in other languages. It can handle longer prompts than GPT-3. Specifically it can analyze, read and generate up to 25,000 words. It is much better at processing programming instructions.

It is also highly steerable. Where GPT-3 would respond in a uniform tone and style, users can tell GPT-4 how they would like it to respond with explicit instructions. This can help with framing prompt and improve prompt engineering.

GPT-4 is trained to limit the possibility of harmful responses and refuse to respond to requests for disallowed content. For example, it was trained to refuse queries about synthesizing dangerous chemicals and answered questions about buying cigarettes without encouraging smoking.