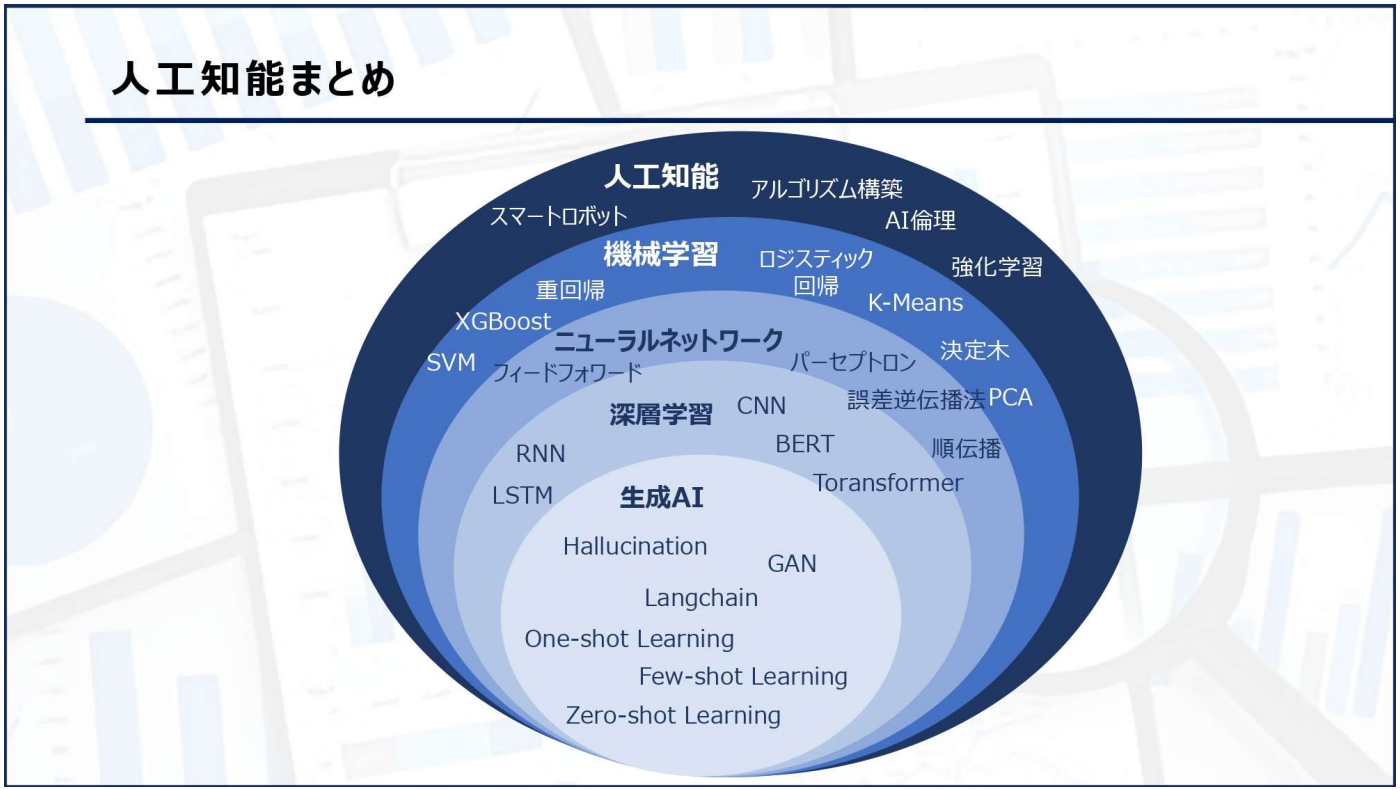
AI とはなにか？



Copilot

Copilot Microsoft Bing Image Creator

「夜の東京タワーとスカイツリーを作成してください。」

ChatGPT

ChatGPT

**「50代のイケてる、ヤングのハートをがっちりキャッチする
ジャージの着こなし方を教えてください。」**

こちらが50代男性のイケてるジャージの着こなしイメージです。

ジャージをスタイリッシュに着こなすポイントが詰まっています。

自信を持って、ファッションを楽しんでください！

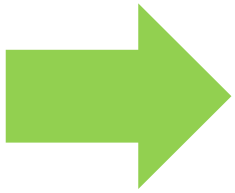
AIツールについて

技術革新のイメージ（産業革命と現代）

19世紀～

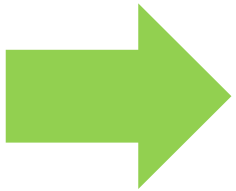
技術革新のイメージ（産業革命と現代）

19世紀～

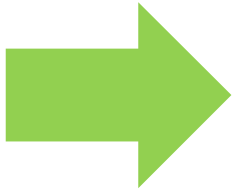


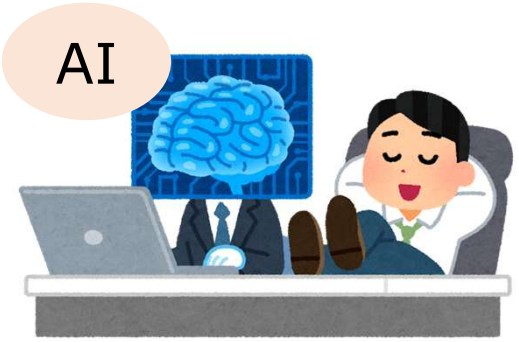
技術革新のイメージ（産業革命と現代）

19世紀～

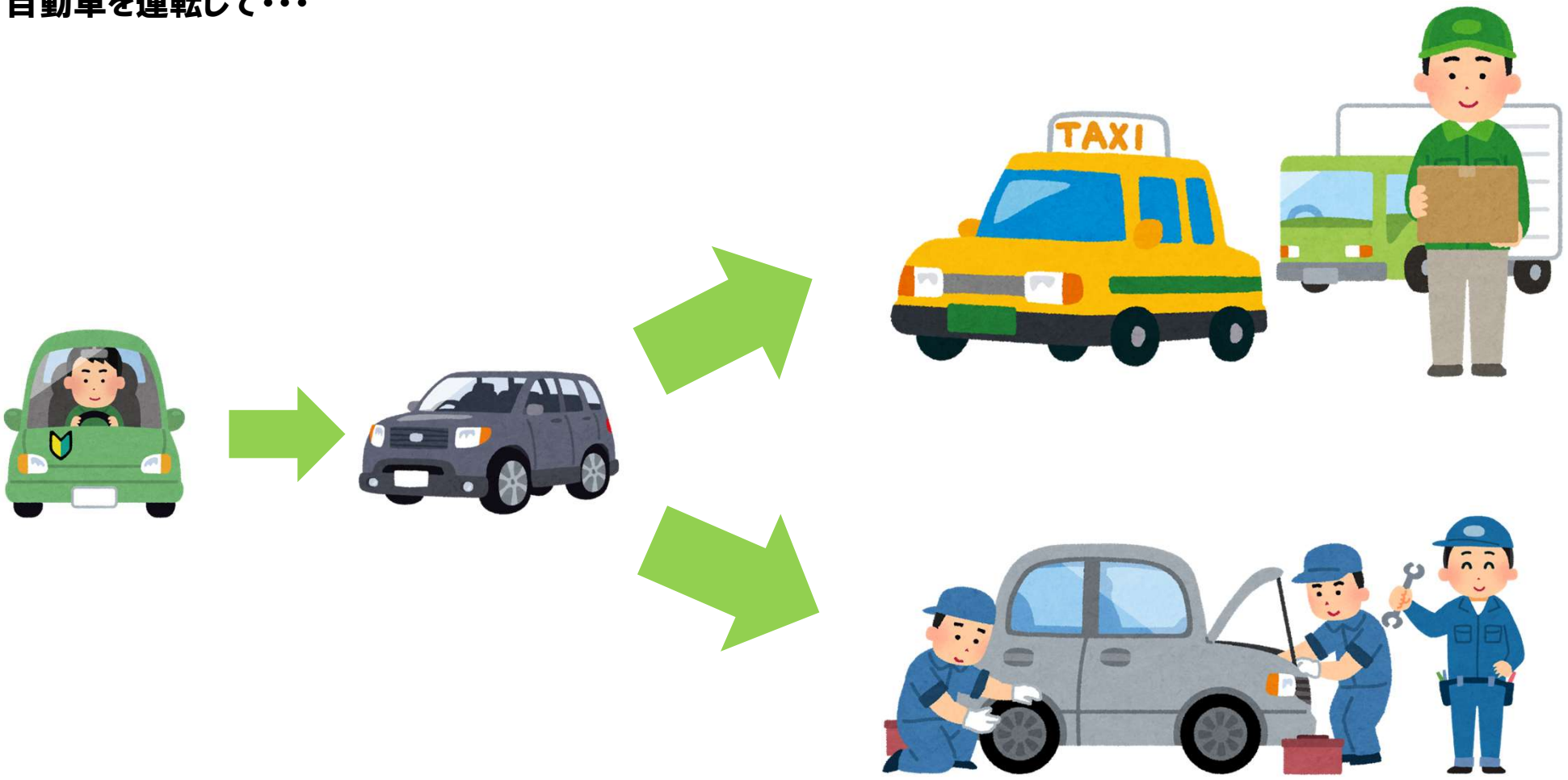


現代～

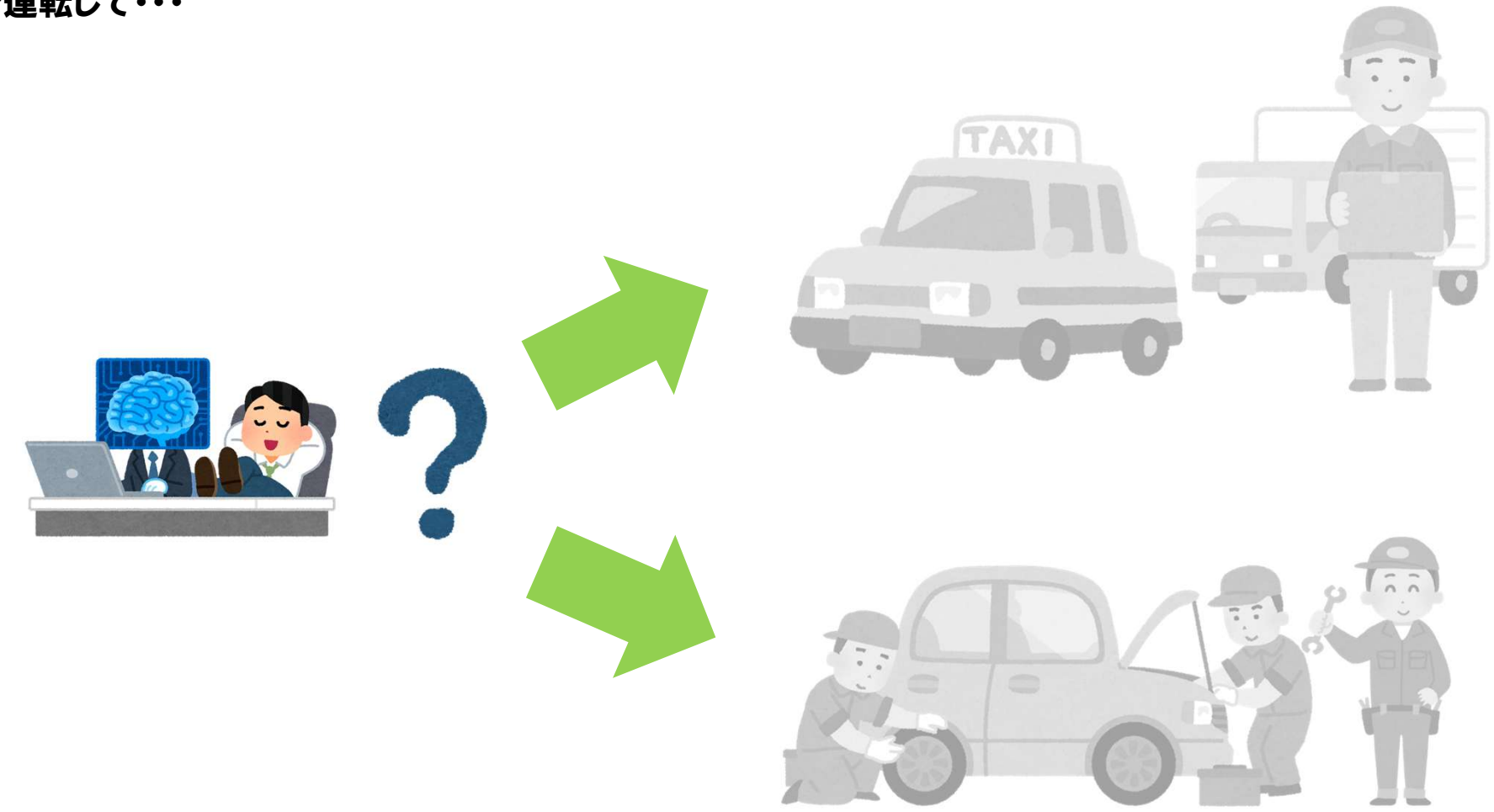




自動車を運転して・・・



自動車を運転して・・・



自動車を運転して・・・

ツールを中心に使いたい・・・

Mapify

Gamma

ChatGPT

perplexity

Claude

Copilot

Canva

仕組みを知っておくべきか・・・

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* † illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

†Work performed while at Google Brain.

‡Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

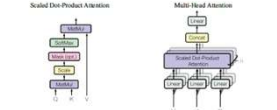


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

3.2.1 Scaled Dot-Product Attention

We call our particular attention "Scaled Dot-Product Attention" (Figure 2). The input consists of queries and keys of dimension d_k, and values of dimension d_v. We compute the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values.

In practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V. We compute the matrix of outputs as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

The two most commonly used attention functions are additive attention [2], and dot-product (multi-plicative) attention. Dot-product attention is identical to our algorithm, except for the scaling factor of $\frac{1}{\sqrt{d_k}}$. Additive attention computes the compatibility function using a feed-forward network with a single hidden layer. While the two are similar in theoretical complexity, dot-product attention is much faster and more space-efficient in practice, since it can be implemented using highly optimized matrix multiplication code.

While for small values of d_k the two mechanisms perform similarly, additive attention outperforms dot product attention without scaling for larger values of d_k [3]. We suspect that for large values of d_k, the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients [4]. To counteract this effect, we scale the dot products by $\frac{1}{\sqrt{d_k}}$.

3.2.2 Multi-Head Attention

Instead of performing a single attention function with d_{model}-dimensional keys, values and queries, we found it beneficial to linearly project the queries, keys and values h times with different, learned linear projections to d_k, d_k and d_v dimensions, respectively. On each of these projected versions of queries, keys and values we then perform the attention function in parallel, yielding d_v-dimensional

[4] To illustrate why the dot products get large, assume that the components of q and k are independent random variables with mean 0 and variance 1. Then their dot product, $q \cdot k = \sum_{i=1}^{d_k} q_i k_i$, has mean 0 and variance d_k.

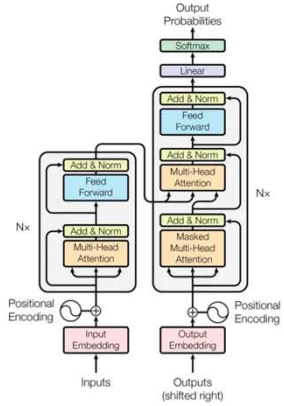


Figure 1: The Transformer - model architecture.

生成AIで『便利になった』ということ

最近、知人の経営者から聞いた。

「生成AIで『便利になった』というのは、電子レンジができて『便利になった』と同じこと」

という言葉は、とても的を得ていると思った。（竹内謙礼）

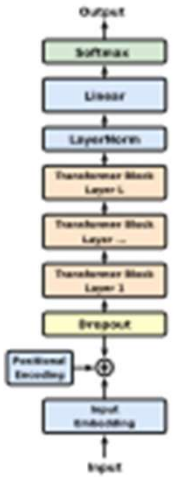
ChatGPT

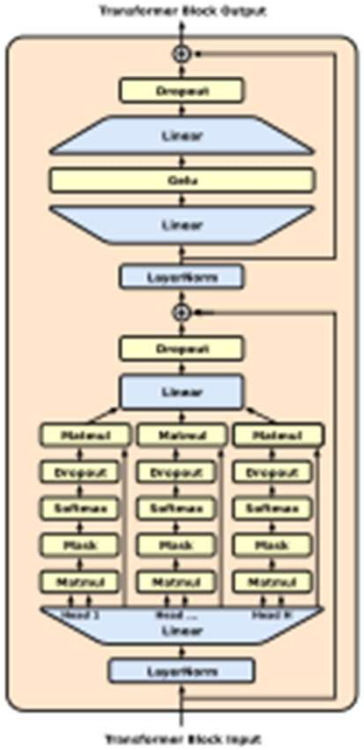
GPT

GPT

ChatGPT

Generative Pre-trained Transformer





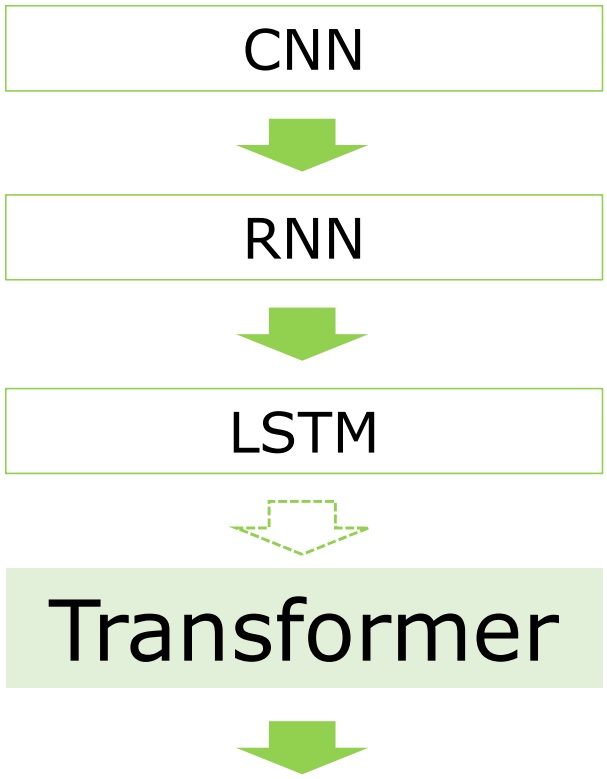
GPT

Mapify × Gamma

Transformerの概要

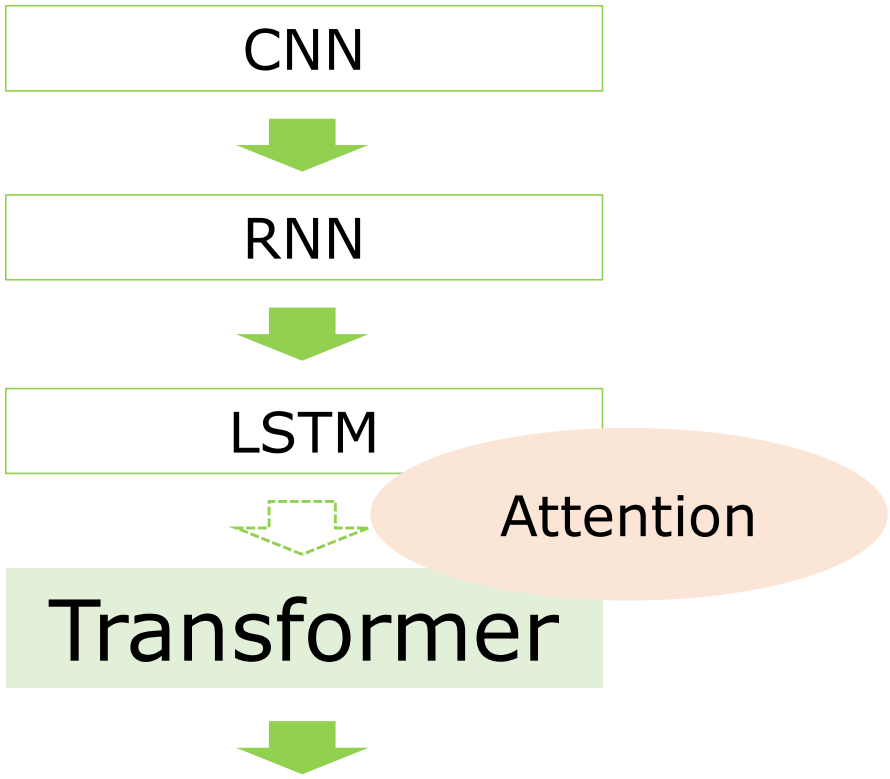
Transformerは、自然言語処理や機械翻訳に使用される深層学習モデルです。自己注意機構を利用して、入力データの異なる部分間の関係を学習します。

Transformerへのアーキテクチャー



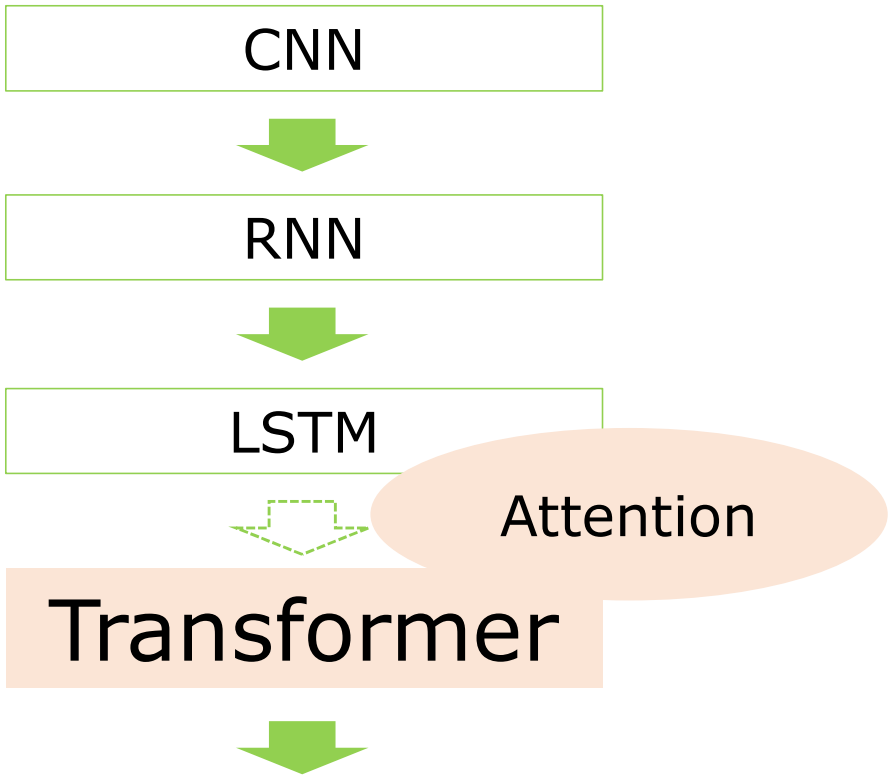
GPT：Generative Pre-trained Transformer

Transformerへのアーキテクチャー



GPT：Generative Pre-trained Transformer

Transformerへのアーキテクチャー

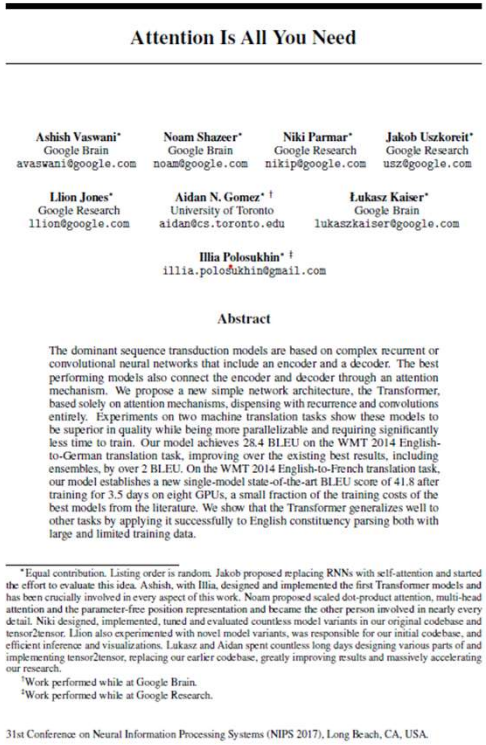


GPT：Generative Pre-trained Transformer

Transformer

arXiv “Attention Is All You Need” (Vaswani et al, 2017)

https://arxiv.org/pdf/1706.03762



Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

†Work performed while at Google Brain.

‡Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

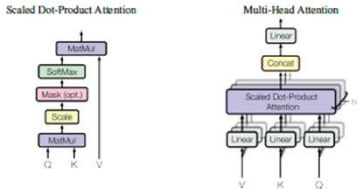


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

3.2.1 Scaled Dot-Product Attention

We call our particular attention "Scaled Dot-Product Attention" (Figure 2). The input consists of queries and keys of dimension d_k, and values of dimension d_v. We compute the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values.

In practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V. We compute the matrix of outputs as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

The two most commonly used attention functions are additive attention [2], and dot-product (multiplicative) attention. Dot-product attention is identical to our algorithm, except for the scaling factor of $\frac{1}{\sqrt{d_k}}$. Additive attention computes the compatibility function using a feed-forward network with a single hidden layer. While the two are similar in theoretical complexity, dot-product attention is much faster and more space-efficient in practice, since it can be implemented using highly optimized matrix multiplication code.

While for small values of d_k the two mechanisms perform similarly, additive attention outperforms dot product attention without scaling for larger values of d_k [3]. We suspect that for large values of d_k, the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients [4]. To counteract this effect, we scale the dot products by $\frac{1}{\sqrt{d_k}}$.

3.2.2 Multi-Head Attention

Instead of performing a single attention function with d_{model}-dimensional keys, values and queries, we found it beneficial to linearly project the queries, keys and values h times with different, learned linear projections to d_k, d_k and d_v dimensions, respectively. On each of these projected versions of queries, keys and values we then perform the attention function in parallel, yielding d_v-dimensional

[4] To illustrate why the dot products get large, assume that the components of q and k are independent random variables with mean 0 and variance 1. Then their dot product, $q \cdot k = \sum_{i=1}^{d_k} q_i k_i$, has mean 0 and variance d_k.

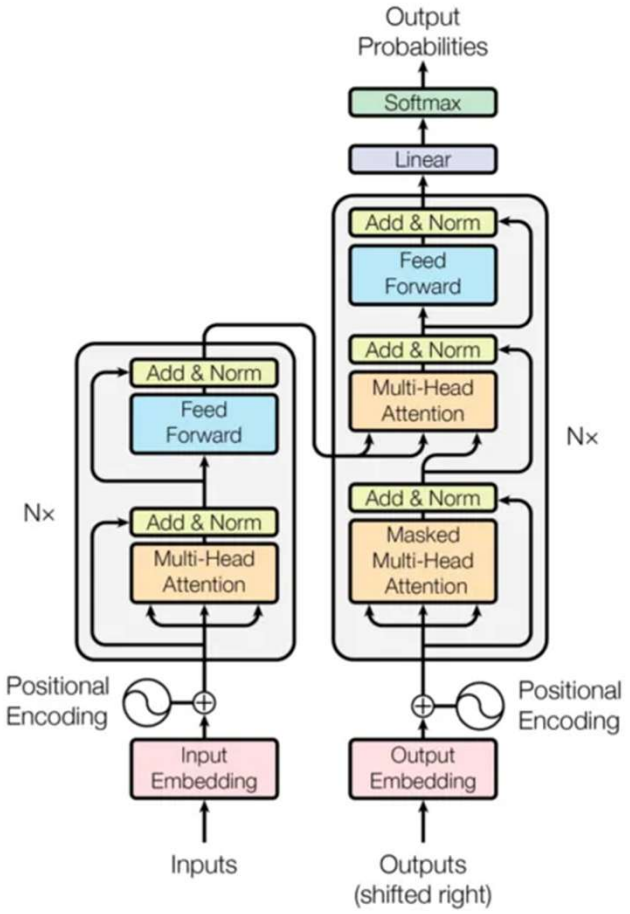
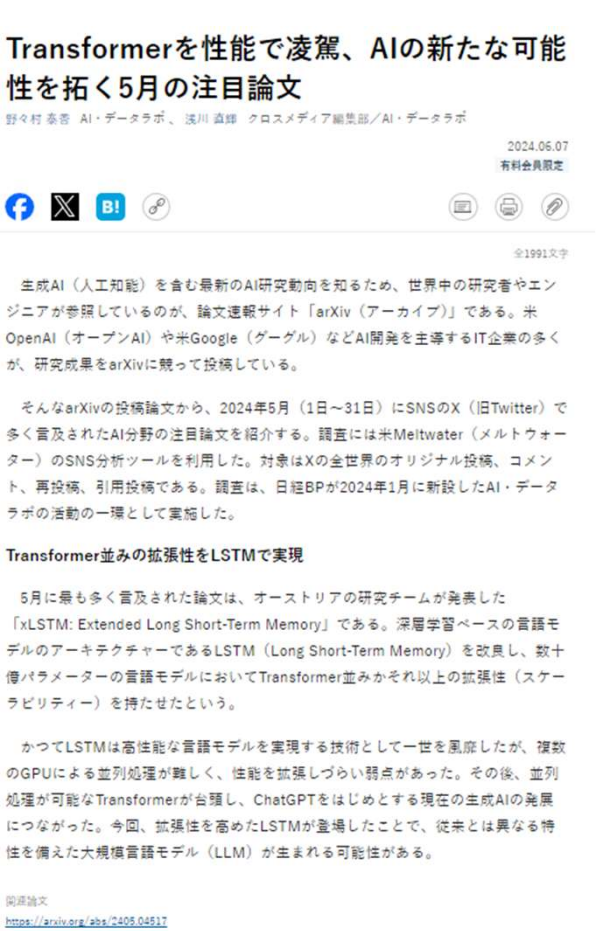


Figure 1: The Transformer - model architecture.

LSTM（Long Short-Term Memory）の拡張版



Transformerを性能で凌駕、AIの新たな可能性を拓く5月の注目論文

野々村 泰志 AI・データラボ、浅川 直輝 クロスメディア編集部／AI・データラボ

2024.06.07
有料会員限定

全1991文字

生成AI（人工知能）を含む最新のAI研究動向を知るため、世界中の研究者やエンジニアが参照しているのが、論文速報サイト「arXiv（アーカイブ）」である。米OpenAI（オープンAI）や米Google（グーグル）などAI開発を主導するIT企業の多くが、研究成果をarXivに競って投稿している。

そんなarXivの投稿論文から、2024年5月（1日～31日）にSNSのX（旧Twitter）で多く言及されたAI分野の注目論文を紹介する。調査には米Meltwater（メルトウォーター）のSNS分析ツールを利用した。対象はXの全世界のオリジナル投稿、コメント、再投稿、引用投稿である。調査は、日経BPが2024年1月に新設したAI・データラボの活動の一環として実施した。

Transformer並みの拡張性をLSTMで実現

5月に最も多く言及された論文は、オーストリアの研究チームが発表した「xLSTM: Extended Long Short-Term Memory」である。深層学習ベースの言語モデルのアーキテクチャーであるLSTM（Long Short-Term Memory）を改良し、数十億パラメーターの言語モデルにおいてTransformer並みかそれ以上の拡張性（スケーラビリティー）を持たせたという。

かつてLSTMは高性能な言語モデルを実現する技術として一世を風靡したが、複数のGPUによる並列処理が難しく、性能を拡張しづらい弱点があった。その後、並列処理が可能なTransformerが台頭し、ChatGPTをはじめとする現在の生成AIの発展につながった。今回、拡張性を高めたLSTMが登場したことで、従来とは異なる特性を備えた大規模言語モデル（LLM）が生まれる可能性がある。

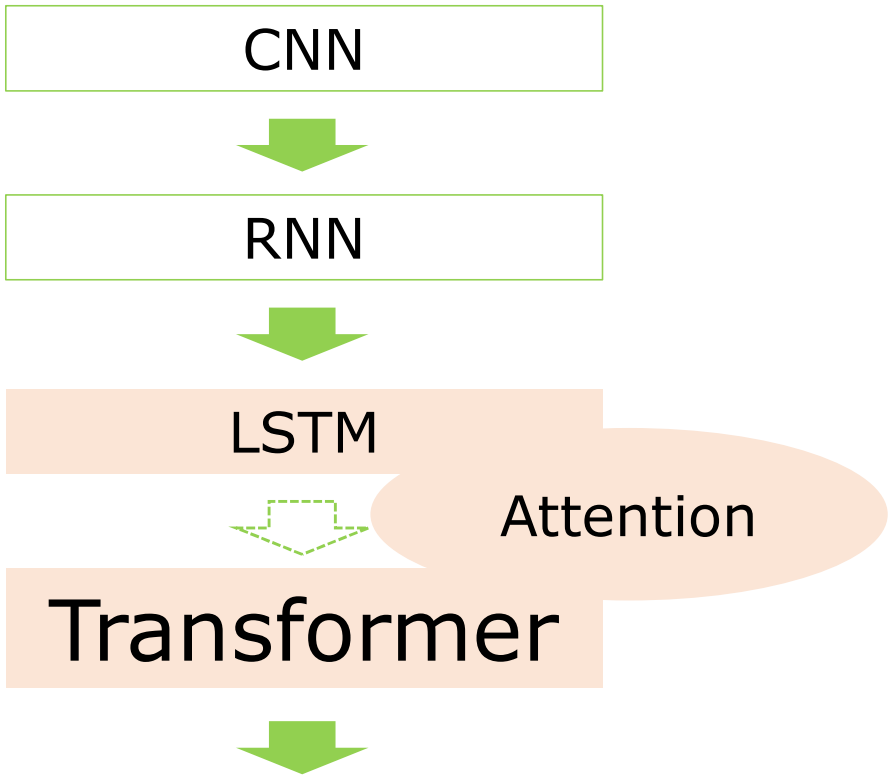
関連論文
https://arxiv.org/abs/2405.04517

日経XTECH
https://xtech.nikkei.com/atcl/nxt/column/18/02801/060400005/

arXiv
https://arxiv.org/abs/2405.04517

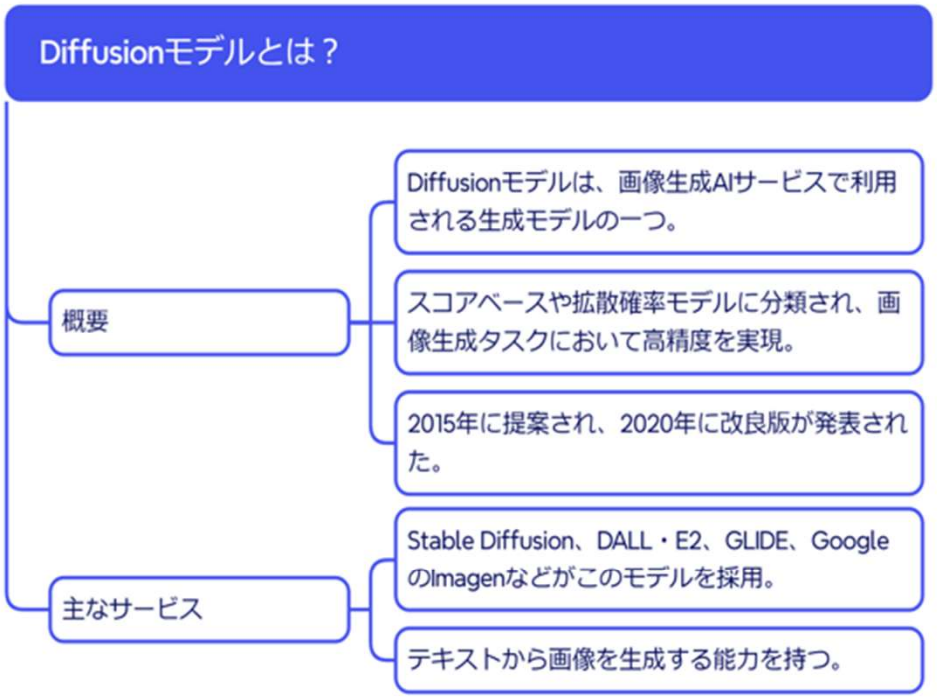
- 深層学習ベースの言語モデルのアーキテクチャーであるLSTM（Long Short-Term Memory）を改良。（2024年6月現在）
- 数十億パラメーターの言語モデルにおいてTransformer並みかそれ以上の拡張性（スケーラビリティー）を持たせた。
- LSTMは高性能な言語モデルを実現する技術として一世を風靡したが、複数のGPUによる並列処理が難しく、性能を拡張しづらい弱点があった。
- その後、並列処理が可能なTransformerが台頭、ChatGPTをはじめとする現在の生成AIの発展につながった。
- 今回、拡張性を高めたLSTMが登場したことで、従来とは異なる特性を備えた大規模言語モデル（LLM）が生まれる可能性がある。

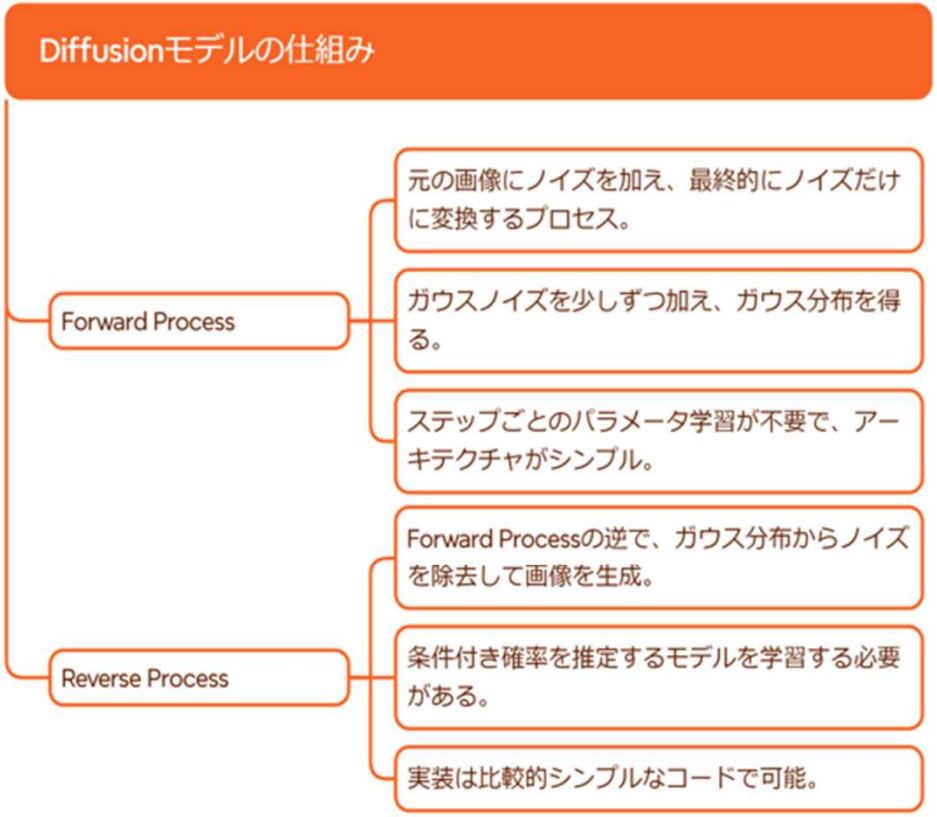
Transformerへのアーキテクチャー



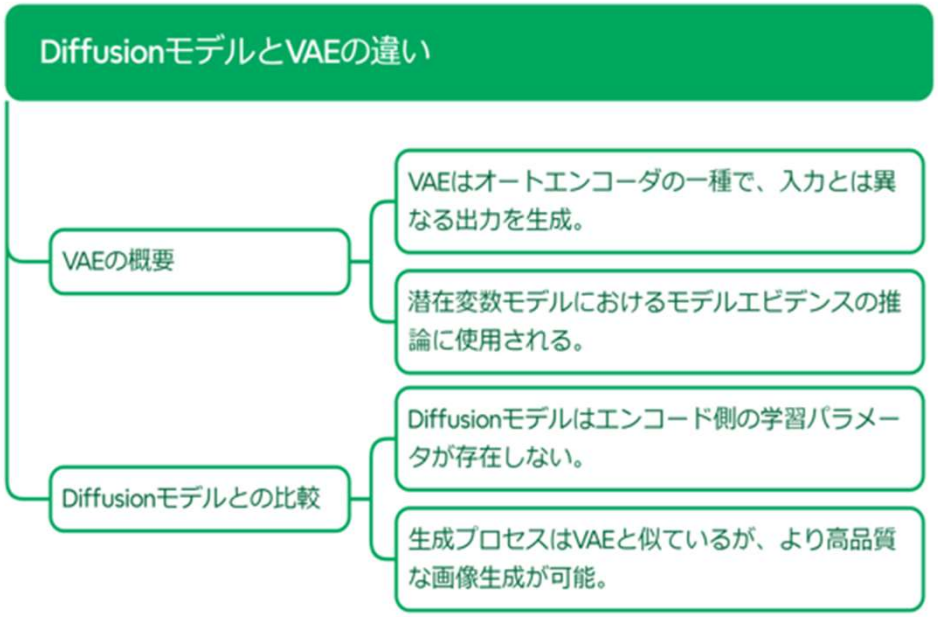
GPT：Generative Pre-trained Transformer

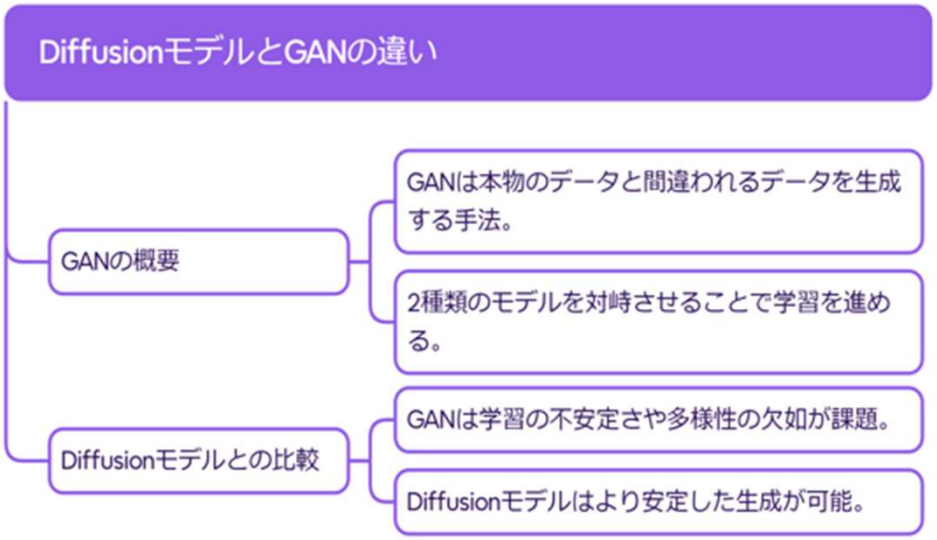
Diffusion model（拡散モデル）



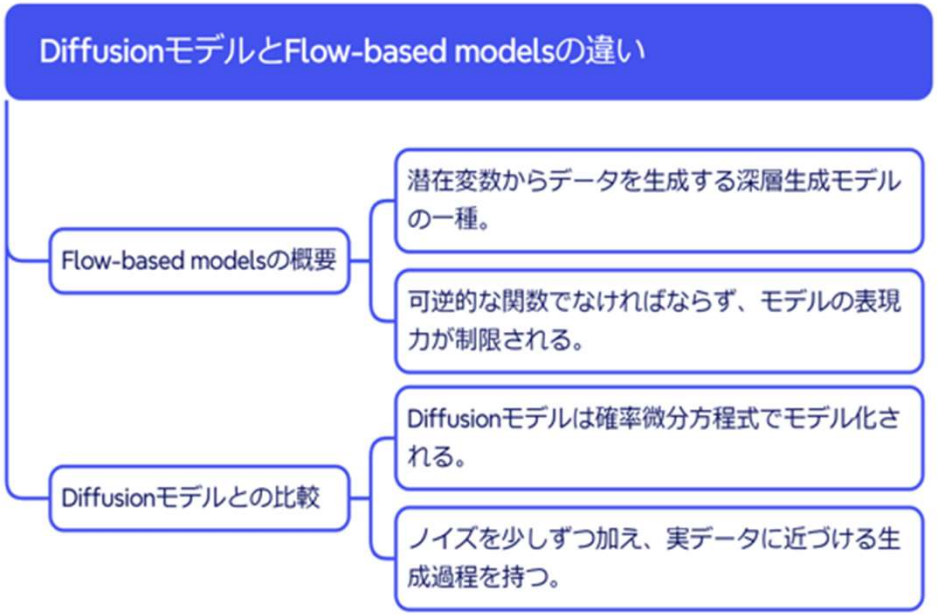


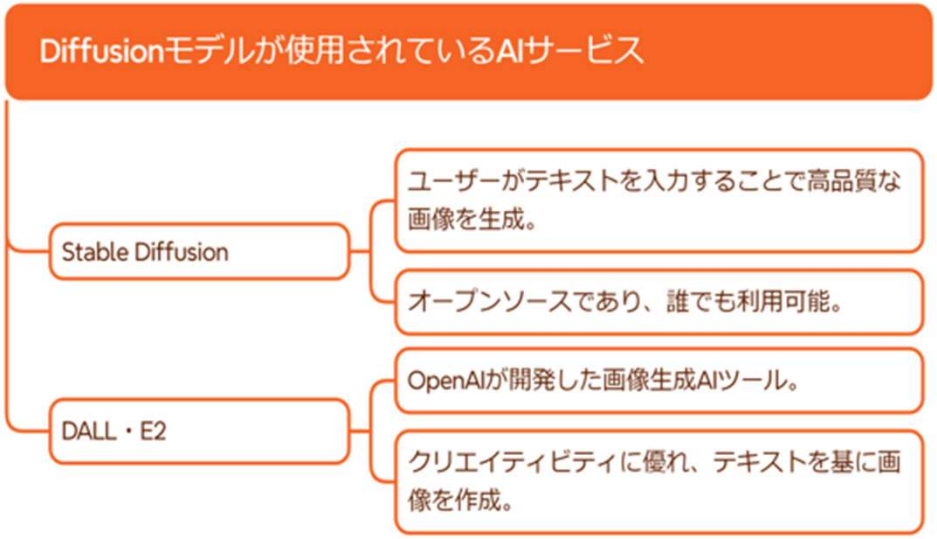
Diffusion model（拡散モデル）



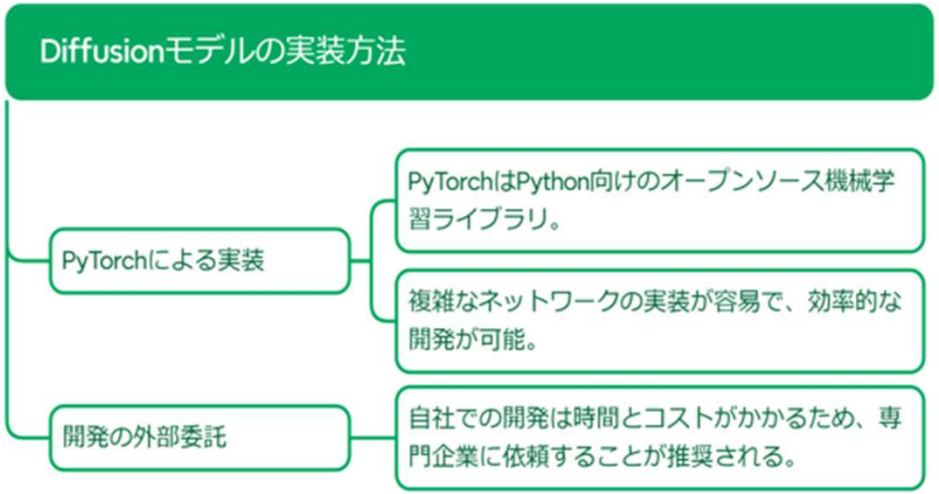


Diffusion model（拡散モデル）





Diffusion model（拡散モデル）



LSTM (Long Short-Term Memory)