

# 読んだ論文の共有

tax\_free

東京工業大学 情報理工学院 数理・計算科学系 学士課程 3 年

November 22, 2024

- 1 The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery [1]
- 2 Attention Is All You Need [3]

# The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery [1]

## 論文情報

著者	Chris Lu and Cong Lu and Robert Tjarko Lange and Jakob Foerster and Jeff Clune and David Ha
雑誌	arXiv
url	<a href="https://arxiv.org/abs/2408.06292">https://arxiv.org/abs/2408.06292</a>

### ■ 背景とモチベーション

- 自律的な科学的発見は大きな挑戦で、AI が一部を支援するものはあるが完全な自律研究は実現できていなかった
- LLM の発展によってアイデア生成、実験設計、実行、結果の可視化、論文執筆、査読の全プロセスを自動化できるようになった。

### ■ 貢献していること

- 機械学習分野における完全自動化研究プロセス「The AI Scientist」を初めて導入し、全体のフレームワークを構築
- 人間レベルの査読精度を持つ自動査読プロセスを開発し、AI 生成論文の品質評価を実装

### ■ キーワード

- { 自動・自律実験, LLM, 自動査読 }

# Model Architecture

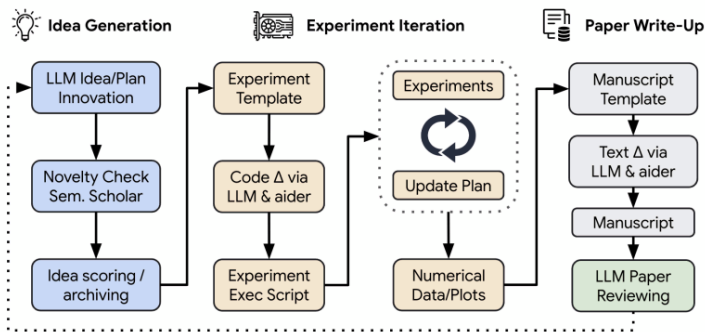


Figure 1 | Conceptual illustration of THE AI SCIENTIST, an end-to-end LLM-driven scientific discovery process. THE AI SCIENTIST first invents and assesses the novelty of a set of ideas. It then determines how to test the hypotheses, including writing the necessary code by editing a codebase powered by recent advances in automated code generation. Afterward, the experiments are automatically executed to collect a set of results consisting of both numerical scores and visual summaries (e.g. plots or tables). The results are motivated, explained, and summarized in a LaTeX report. Finally, THE AI SCIENTIST generates an automated review, according to current practice at standard machine learning conferences. The review can be used to either improve the project or as feedback to future generations for open-ended scientific discovery.

## Model Architecture - Idea Generation

### アイデア生成のポイント

- 与えられたテンプレートを種に, LLM を mutation operator として進化計算をしてアイデアを成長させる.
- 各アイデアに含まれるスコア (面白さ, 新規性, 実現可能性など) が含まれる.
- 複数回の CoT と self-reflection を行ってアイデアを洗練する.
- アイデアを生成した後に Semantic Scholar API と web 検索を使用して過去の研究と類似したものをフィルタする.

### self-reflection とは [2]

- 従来の Reinforcement learning で LLM を微調整するのに非常に大きなコストがかかる.
- 重みを変更するのではなく LLM の memory に verbal な reflection を入れることでモデルの重みを変更することなくフィードバックできるようにした.

## Model Architecture - Experiment Iteration

以下の自動実験のプロセスに従って実験を行い，結果を踏まえて最大 5 回まで実行する。

### 自動実験のプロセス

- Aider を使って実行する実験のリストを計画して実験リストの計画を順番に実行する。
- 実験のメモ，プロットを保存する。
- 失敗やタイムアウトした場合はエラーを返し，Aider がコードを修正して最大 4 回まで再試行する。

strict なテンプレートを使わないことで新たな指標を発見するなどの独創性が生まれる。

### Aider とは

- LLM を活用したコード支援を行うチャットボットのような開発支援ツール
- コードの最適化，修正，ドキュメントの改善などを LLM が提案してくれる

# Model Architecture - Paper Write-up

進捗状況を標準的な ML 系の学会の LaTeX フォーマットで出力する。執筆は熟練した研究者でも時間を要することがあるので以下のプロセスでサポートしている。

## 論文執筆のプロセス

- セクションごとのテキスト生成:
  - 実験の過程で保存したメモとプロットを基に Aider がテンプレートに沿って結果を"入力"する (hallucination を防ぐために根拠を要求する)。
  - self-reflection を一回して精緻化する。
- references の web 検索:
  - Semantic Scholar API を最大で 20 回まで実行して深堀りする。
  - 検索結果を BibTex に保存して信頼性を確保する。
- 精緻化
  - 上記の 2 段階だけだと冗長で繰り返しが多い場合があるので、重複表現を除去して一貫性を持たせるためにセクションごとに self-reflection を行う。

# Attention Is All You Need [3]

## 論文情報

著者	Ashish Vaswani and Noam Shazeer and Niki Parmar and Jakob Uszkoreit and Llion Jones and Aidan N. Gomez and Lukasz Kaiser and Illia Polosukhin
雑誌	arXiv
url	<a href="https://arxiv.org/abs/1706.03762">https://arxiv.org/abs/1706.03762</a>

### ■ 背景とモチベーション

- 当時 ( 2017 年 ) の言語処理系の問題では RNN, 特に LSTM が主流だった.
- 工夫して緩和されているけど, recurrent network の構造的な問題で RNN や LSTM は並列化が難しく, また, sequence 長が大きくなるとメモリの制約によってバッチ処理が制限されて困る.
- Attention だけを用いた Transformer というモデルを使うことで recurrent の部分を排除して上の課題を解決できる.

### ■ 貢献していること

- 当時の英独・英仏翻訳タスクで SoTA を達成
- 流行りの? LLM の基になっている Transformer を提案

### ■ キーワード

- { Machine Learning, NLP, Transformer, Sequence Transduction }



# Model Architecture

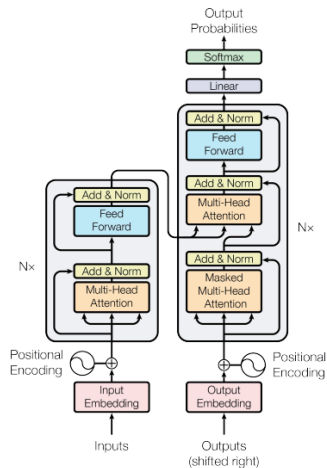


Figure 1: The Transformer - model architecture.

## Model Architecture - Encoder and Decoder Stacks

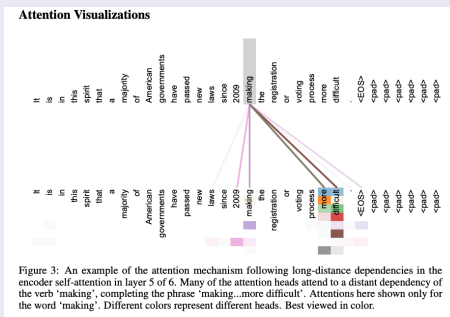
Transformer アーキテクチャは、主に Encoder と Decoder から構成されている。それぞれのスタックは、同一の層が積み重なった構造で、入力処理して表現を生成する Encoder と、その表現を元に出力を生成する Decoder の 2 つの役割を担う。

- Encoder: 各 Encoder layer は、Multi-Head Self-Attention, position-wise fully connected Feed Forward (FFN) から構成される。
- Decoder: 各 Decoder layer は、Masked Multi-Head Attention, Multi-Head Self-Attention, position-wise fully connected Feed Forward (FFN) の 3 つのサブ層を持つ。

Encoder と Decoder のスタックは、タスクに応じて層の数を変更することができ、柔軟に設計できる。

## Attention

- Attention は、入力シーケンスの異なる部分に焦点を当て、関連性の高い情報に重みを付けて処理する。これにより、長距離依存関係を効率的に捉えることができる。
- Transformer では、Scaled Dot-Product Attention を用いて、query と key の類似度に基づき重みを計算し、Multi-Head Attention で並列処理を行う。



# Model Architecture

## Scaled Dot-Product Attention

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

- Q: query
- K: key
- $d_k$ : key の次元

各単語が query, key という特徴量を持っている。スケールしているのは,  $d_k$  が大きいと性能が下がってしまうのを防ぐため。

# Model Architecture

## Multi-Head Attention

複数の Attention を並列で計算し，潜在表現を持てるようにする．各ヘッ드의出力を結合し，最終的に線形変換を行う．

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O$$

## Training

Table: Training Data and Batching

項目	詳細
データセット (英独)	WMT 2014 English-German, 4.5 百万ペアの文
ボキャブラリサイズ (英独)	約 37,000 トークン (Byte-Pair Encoding 使用)
データセット (英仏)	WMT 2014 English-French, 3,600 万文
ボキャブラリサイズ (英仏)	約 32,000 トークン (Word-Piece 使用)
バッチサイズ	約 25,000 ソーストークンと 25,000 ターゲットトークン




Table: Hardware and Schedule

項目	詳細
ハードウェア	8 台の NVIDIA P100 GPU
訓練ステップ時間 (ベースモデル)	1 ステップあたり約 0.4 秒
訓練時間 (ベースモデル)	100,000 ステップ, 12 時間
訓練ステップ時間 (ビッグモデル)	1 ステップあたり約 1.0 秒
訓練時間 (ビッグモデル)	300,000 ステップ, 3.5 日間

## Result

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	<b>41.29</b>	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	<b><math>3.3 \cdot 10^{18}</math></b>	
Transformer (big)	<b>28.4</b>	<b>41.8</b>	$2.3 \cdot 10^{19}$	

-  Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha.  
The ai scientist: Towards fully automated open-ended scientific discovery, 2024.
-  Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao.  
Reflexion: Language agents with verbal reinforcement learning, 2023.
-  Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin.  
Attention is all you need, 2023.