# AI大模型应用开发·实战营

彭靖田

Google Developer Experts
LangChain Developer

# 课程安排

# AI 大模型应用开发·学习路径

# AI 大模型应用开发·技术栈

# AI大模型应用开发 ·实战

## OpenAI-Translator

English | 中文

*All the code and documentation are entirely generated by OpenAI's GPT-4 Model*

### Introduction

OpenAI Translator is an AI-powered translation tool designed to translate English PDF books to Chinese. The tool leverages large language models (LLMs) like ChatGLM and OpenAI's GPT-3 and GPT-3.5 Turbo for translation. It's built in Python and has a flexible, modular, and object-oriented design.

### Why this project

In the current landscape, there's a lack of non-commercial yet efficient PDF translation tools. Many users have PDF documents with sensitive data that they prefer not to upload to public commercial service websites due to privacy concerns. This project was developed to address this gap, providing a solution for users who need to translate their PDFs while maintaining data privacy.

### Sample Results

The OpenAI Translator is still in its early stages of development, and I'm actively working on adding more features and improving its performance. We appreciate any feedback or contributions!

### Features

- [x] Translation of English PDF books to Chinese using LLMs.
- [x] Support for both ChatGLM and OpenAI models.
- [x] Flexible configuration through a YAML file or command-line arguments.
- [x] Timeouts and error handling for robust translation operations.
- [x] Modular and object-oriented design for easy customization and extension.
- [ ] Implement a graphical user interface (GUI) for easier use.
- [ ] Add support for batch processing of multiple PDF files.
- [ ] Create a web service or API to enable usage in web applications.
- [ ] Add support for other languages and translation directions.
- [ ] Add support for preserving the original layout and formatting of the source PDF.
- [ ] Improve translation quality by using custom-trained translation models.

### Getting Started

#### Environment Setup

1.Clone the repository `git clone git@github.com:DjangoPeng/openai-translator.git`.

2.The `OpenAI-Translator` requires Python 3.6 or later. Install the dependencies with `pip install -r requirements.txt`.

3.Set up your OpenAI API key( `$OPENAI_API_KEY` ) or ChatGLM Model URL( `$GLM_MODEL_URL` ). You can either add it to your environment variables or specify it in the config.yaml file.

#### Usage

You can use OpenAI-Translator either by specifying a configuration file or by providing command-line arguments.

**Using a configuration file:**

Adapt `config.yaml` file with your settings:

```yaml
OpenAIModel:
  model: "gpt-3.5-turbo"
  api_key: "your_openai_api_key"

GLMModel:
  model_url: "your_chatglm_model_url"
  timeout: 300
```

https://github.com/DjangoPeng/openai-translator

| 基础篇 | | |
|---|---|---|
| 大模型基础：理论与技术的演进 | • 初探大模型：起源与发展<br>• GPT 模型家族：从始至今 | 7 月 12 日（周三）<br>20:00–22:00 |
| | • 大模型的未来：开源力量与生态繁荣<br>• 提示学习（Prompt Learning） | 7 月 16 日（周日）<br>19:00–22:00 |
| 大模型开发基础 | • Embedding<br>• OpenAI 大模型选择指南<br>• OpenAI 大模型二次开发指南 | 7 月 19 日（周三）<br>20:00–22:00 |
| 大模型应用最佳实践 & 硬件选型 | • 大模型应用最佳实践<br>• 大模型核心硬件选型指南 | 7 月 23 日（周日）<br>19:00–22:00 |
| 基于 ChatGPT 开发智能翻译助手 | • OpenAI–Translator 市场需求分析<br>• OpenAI–Translator 产品定义与功能规划<br>• OpenAI–Translator 技术方案与架构设计<br>• OpenAI–Translator 实战 | 7 月 26 日（周三）<br>20:00–22:00 |
| 动手开发第一个 ChatGPT Plugin | • ChatGPT Plugin 开发指南<br>• 将 OpenAI–Translator 改造成 ChatGPT Plugin | 7 月 30 日（周日）<br>19:00–22:00 |

| | 生态篇 | |
|---|---|---|
| 大模型开源生态 | • 大模型排行榜与基准测试<br>• 大模型横向对比<br>• 大模型时代的 Github：Hugging Face<br>• 使用 HuggingFace 托管你的 LLMs 应用 | 8 月 27 日（周日）<br>19:00–22:00 |
| 国产大模型 ChatGLM 应用开发 | • ChatGLM 与 VisualGLM 家族<br>• ChatGLM–6B 介绍与快速入门<br>• ChatGLM–6B fine–tuning<br>• ChatGLM–6B 私有化部署<br>• 基于 ChatGLM 打造你的私有化 "ChatGPT" | 8 月 30 日（周三）<br>20:00–22:00 |
| 结课直播 | 课程总结 + 答疑 + 加餐知识点 | 9 月 3 日（周日）<br>19:00–22:00 |

| | 进阶篇 | |
|---|---|---|
| 大模型应用开发框架 LangChain | • LangChain 项目介绍<br>• LangChain 开发上手指南<br>• 标准化的大模型抽象：Models<br>• 大模型的输入抽象：Prompts<br>• 大模型应用的最佳实践：Chains<br>• 实现复杂应用的代理系统：Agents | 8 月 2 日（周三）<br>20:00–22:00 |
| | • 检索非结构化数据的利器：Indexes<br>• 赋予应用记忆的能力：Memory<br>• 大模型原生的回调系统：Callbacks<br>• LangChain 典型应用剖析 | 8 月 6 日（周日）<br>19:00–22:00 |
| 实战 3:<br>使用 Langchain 重新实现智能翻译助手 | • 接入 LangChain 后的架构设计<br>• 基于 LangChain 版本重构的方案调整<br>• 实战 LangChain 版 OpenAI–Translator | 8 月 9 日（周三）<br>20:00–22:00 |
| 实战 4:<br>手把手带你实现网红项目 AutoGPT | • AutoGPT 原始版本定位与功能解读<br>• LangChain 版 AutoGPT 技术方案与架构设计<br>• 实战 LangChain 版 AutoGPT | 8 月 13 日（周日）<br>19:00–22:00 |
| 实战 5:<br>基于知识库的销售顾问 Sales–Consultant | • Sales–Consultant 市场需求分析<br>• Sales–Consultant 产品定义与功能规划 | 8 月 16 日（周三）<br>20:00–22:00 |
| | • Sales–Consultant 技术方案与架构设计<br>• 实战 LangChain 版 Sales–Consultant | 8 月 20 日（周日）<br>19:00–22:00 |
| 大模型应用落地与数据隐私 | • 大模型时代的开源与数据协议<br>• 数据安全性和保护措施<br>• 模型的可解释性和可审计性<br>• 伦理和道德考虑<br>• 法规合规性<br>• 数据隐私的量化评估 | 8 月 23 日（周三）<br>20:00–22:00 |

# 热点问题回应

问：是否必须使用 **GPT-4 API**？
答：不是。

问：除了 **GPT** 还会讲别的大模型吗？
答：会。

问：是否需要科学上网？
答：

问：是否会讲基于私有数据的开发？
答：是。

问：学完之后可以能独立开发大模型应用吗？
答：可以

# AI 大模型应用开发·内容安排

大模型应用实战（Practical Application of LLM）

- Prompt Enginering: ChatGPT, LLaMA
- LLM Developer: Apps based on LangChain

元知识（Meta-knowledge）

- Knowing how to learn
- How to analyze and solve problems

合作共赢（Cooperation and Win-Win）

- Human-AI Collaboration
- Multi-person Cooperation

创新思维（Innovative thinking）

- Tracking new technology trends
- Innovative ideas and practices.

# 大模型基础：理论与技术的演进

彭靖田

Google Developer Experts
LangChain Developer

# 目录


- 初探大模型：起源与发展
  - 预热篇：解码注意力机制（Attention）
  - 变革里程碑：Transformer的崛起
  - 走向不同：GPT与BERT的选择
- GPT 模型家族：从始至今
  - 从GPT-1到GPT-3.5：一路的风云变幻
  - ChatGPT：赢在哪里
  - GPT-4：一个新的开始

# 初探大模型：起源与发展

# AI 历史上的四次大发展

Activation model of the neuron (McCulloch, 1943)
Macy Conference
ONR funds Rosenblatt's Perceptron (1957-1962)
Backpropagation algorithm (Rumelhart, 1986)
Convolution on zip codes (Lecun, 1989)
Support Vector Machines - SVM (Vapnik, 1995)
DARPA funds Minsky and McCarthy AI group at MIT (1963-1974)
Connectionist
Symbolic
Dartmouth workshop
Invention of the term "Artificial Intelligence"
Resolution by logic (Robinson, 1965)
Criticism of the Perceptron (Minsky, 1969)
1st AI winter
Expert systems development
LISP machines market collapse
Ratio connectionist / symbolic (log)
1.5
1.25
1.0
0.8
0.66
1940
1950
1960
1970
1980
1990
2000
Publication year of cited references (1935-2005)

Cited between 1980 and 1989
Cited between 1990 and 1999
Cited between 2000 and 2009
Cited between 2010 and 2018
% of WoS

Connectionist
Zadeh
Dubois
Yager
Kosko
Mamdani
Holland
Goldberg
Zhang
Karaboga
Michalewicz
Garey
Kisi
Li
Mellit
Kennedy
Dorigo
Kirkpatrick
Jang
Hornik
Chau
Wang
Koza
Glover
Haykin
Hagan
Jain
Hopfield
Kohonen
Kim
Vapnik
Bishop
Han
Rumelhart
Breiman
Lecun
Witten
Hinton
Sutton
Quinlan
Pawlak
Duda
Mitchell
Michalski
Kloopman
Aamodta
Kolodner
Heckerman

Symbolic
Prakken
Walton
Russell
Pearl
Dechter
Russell Stuart
McDermott
Fox
Wooldridge
Kowalski
Reiter
Shoham
Nilsson
Jennings
Fikes
Allen
McCarthy
Rich
Charniak
Genesereth
Lenat
Cohen
Maes
Brachman
Kuipers
Winston
Sowa
Davis
Forbus
Schank
Buchanan
Dekleer
Feigenbaum
Shortliffe
Chandrasekaran
Shannon
Clancey
Wiener
Minsky
Winograd
Simon
Brooks
Marr
Pfeifer
Boden
Fodor
Anderson
Searle
Dennett
Newell
Turing
Laird
Dreyfus

# IMAGENET 数据集与竞赛

Data

Compute

# 数据和算力增长推动深度学习的繁荣发展（2010’s）

Jeff Dean, Google Brain Team, Building Intelligent Systems with Large Scale Deep Learning

## The blessings of scale

### AI training runs, estimated computing resources used

Floating-point operations, selected systems, by type, log scale

Sources: "Compute trends across three eras of machine learning", by J. Sevilla et al., arXiv, 2022; Our World in Data

## Quick learners

The speed at which artificial intelligence models master benchmarks and surpass human baselines is accelerating. But they often fall short in the real world.

(GRAPHIC) K. FRANKLIN/*SCIENCE*; (DATA) D. KIELA *ET AL.*, DYNABENCH: RETHINKING BENCHMARKING IN NLP, DOI:10.48550/ARXIV.2104.14337

# 预热篇：解码注意力机制（Attention Mechanism）

# What is Attention?

说重点！！！

# 问：“我”去了几次咖啡店？

"昨天，我在一个繁忙的一天结束后，决定去我最喜欢的咖啡店放松一下。我走进咖啡店，点了一杯拿铁，然后找了一个靠窗的位置坐下。我喝着咖啡，看着窗外的人们匆匆忙忙，感觉非常惬意。然后，我从咖啡店出来，回到了家中。"

# 问：“我”去了几次咖啡店？

"昨天，我在一个繁忙的一天结束后，决定去我最喜欢的**咖啡店**放松一下。我走进**咖啡店**，点了一杯拿铁，然后找了一个靠窗的位置坐下。我喝着咖啡，看着窗外的人们匆匆忙忙，感觉非常惬意。然后，我从**咖啡店**出来，回到了家中。"

# 高频 ≠ 重点

"昨天，我在一个繁忙的一天结束后，决定去我最喜欢的**咖啡店**放松一下。我走进**咖啡店**，点了一杯拿铁，然后找了一个靠窗的位置坐下。我喝着咖啡，看着窗外的人们匆匆忙忙，感觉非常惬意。然后，我从**咖啡店**出来，回到了家中。"

在这个例子中，"咖啡店"出现了三次，但并非每次都是关键信息。

- 当我们理解"我走进咖啡店，点了一杯拿铁"时，"走进"和"点了一杯拿铁"是关键动作，而"咖啡店"的信息已经被我们知道了。
- 同样，当理解"我从咖啡店出来，回到了家中"时，"从咖啡店出来"和"回到了家中"是关键信息，而"咖啡店"的信息仍然是冗余的。

注意力机制就是帮助模型在处理这样的句子时，能够更好地**关注到关键的信息**，而忽略冗余的信息。

# But How?

# Encoder-decoder architecture

# Encoder-decoder Architecture with Attention Model

| Function | Traditional Encoder-Decoder | Encoder-Decoder with Attention |
|---|---|---|
| Encode | $h_i = f(x_i, h_{i-1})$ | $h_i = f(x_i, h_{i-1})$ |
| Context | $c = h_T$ | $c_j = \sum_{i=1}^{T} \alpha_{ij} h_i$ |
| | | $\alpha_{ij} = p(e_{ij})$ |
| | | $e_{ij} = a(s_{j-1}, h_i)$ |
| Decode | $s_j = f(s_{j-1}, y_{j-1}, c)$ | $s_j = f(s_{j-1}, y_{j-1}, c_j)$ |
| Generate | $y_j = g(y_{j-1}, s_j, c)$ | $y_j = g(y_{j-1}, s_j, c_j)$ |

$x = (x_1, ..., x_T)$: input sequence, $T$: length of input sequence, $h_i$: hidden states of encoder, $c$: context vector, $\alpha_{ij}$: attention weights over input, $s_j$: decoder hidden state, $y_j$: output token, $f, g$: non-linear functions, $a$: alignment function, $p$: distribution function

Table 1. Encoder-decoder architecture: traditional and with attention model

Fig. 2. Encoder-decoder architecture: (a) traditional (b) with attention model

Chaudhari, S., Mithal, V., Polatkan, G. and Ramanath, R., 2021. An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), pp.1-32.

# Encoder-decoder Architecture with Attention Model

| Function | Traditional Encoder-Decoder | Encoder-Decoder with Attention |
|---|---|---|
| Encode | $h_i = f(x_i, h_{i-1})$ | $h_i = f(x_i, h_{i-1})$ |
| Context | $c = h_T$ | $c_j = \sum_{i=1}^{T} \alpha_{ij} h_i$ |
| | | $\alpha_{ij} = p(e_{ij})$ |
| | | $e_{ij} = a(s_{j-1}, h_i)$ |
| Decode | $s_j = f(s_{j-1}, y_{j-1}, c)$ | $s_j = f(s_{j-1}, y_{j-1}, c_j)$ |
| Generate | $y_j = g(y_{j-1}, s_j, c)$ | $y_j = g(y_{j-1}, s_j, c_j)$ |

$x = (x_1, ..., x_T)$: input sequence, $T$: length of input sequence, $h_i$: hidden states of encoder, $c$: context vector, $\alpha_{ij}$: attention weights over input, $s_j$: decoder hidden state, $y_j$: output token, $f, g$: non-linear functions, $a$: alignment function, $p$: distribution function

Table 1. Encoder-decoder architecture: traditional and with attention model

Fig. 2. Encoder-decoder architecture: (a) traditional (b) with attention model

Chaudhari, S., Mithal, V., Polatkan, G. and Ramanath, R., 2021. An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), pp.1-32.

# Encoder-decoder Architecture with Attention Model

| Function | Traditional Encoder-Decoder | Encoder-Decoder with Attention |
|---|---|---|
| Encode | $h_i = f(x_i, h_{i-1})$ | $h_i = f(x_i, h_{i-1})$ |
| Context | $c = h_T$ | $c_j = \sum_{i=1}^{T} \alpha_{ij} h_i$ |
| | | $\boxed{\alpha_{ij} = p(e_{ij})}$ |
| | | $e_{ij} = a(s_{j-1}, h_i)$ |
| Decode | $s_j = f(s_{j-1}, y_{j-1}, c)$ | $s_j = f(s_{j-1}, y_{j-1}, c_j)$ |
| Generate | $y_j = g(y_{j-1}, s_j, c)$ | $y_j = g(y_{j-1}, s_j, c_j)$ |

$x = (x_1, ..., x_T)$: input sequence, $T$: length of input sequence, $h_i$: hidden states of encoder, $c$: context vector, $\alpha_{ij}$: attention weights over input, $s_j$: decoder hidden state, $y_j$: output token, $f, g$: non-linear functions, $a$: alignment function, $p$: distribution function

Table 1. Encoder-decoder architecture: traditional and with attention model

Attention weights $\alpha$
Capture the relevance between h and $s$

↓

sigmoid & softmax

Chaudhari, S., Mithal, V., Polatkan, G. and Ramanath, R., 2021. An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), pp.1-32.

# Encoder-decoder Architecture with Attention Model

| Function | Traditional Encoder-Decoder | Encoder-Decoder with Attention |
|---|---|---|
| Encode | $h_i = f(x_i, h_{i-1})$ | $h_i = f(x_i, h_{i-1})$ |
| Context | $c = h_T$ | $c_j = \sum_{i=1}^{T} \alpha_{ij} h_i$ |
| | | $\alpha_{ij} = p(e_{ij})$ |
| | | $e_{ij} = \boxed{a(s_{j-1}, h_i)}$ |
| Decode | $s_j = f(s_{j-1}, y_{j-1}, c)$ | $s_j = f(s_{j-1}, y_{j-1}, c_j)$ |
| Generate | $y_j = g(y_{j-1}, s_j, c)$ | $y_j = g(y_{j-1}, s_j, c_j)$ |

$x = (x_1, ..., x_T)$: input sequence, $T$: length of input sequence, $h_i$: hidden states of encoder, $c$: context vector, $\alpha_{ij}$: attention weights over input, $s_j$: decoder hidden state, $y_j$: output token, $f, g$: non-linear functions, $a$: alignment function, $p$: distribution function

Table 1. Encoder-decoder architecture: traditional and with attention model

**Learning attention weights**

↓

**a: alignment function**

based on a notion of comparing **query** representations with **key** representations

***e* = *a*(*K*,*q*) and *α* = *p*(*e*)**

↓

e.g. softmax

$$A(q, K, V) = \sum_i p(a(k_i, q)) * v_i$$

$k_i = v_i = h_i$ In some cases, there is also **additional input of values *V***

Chaudhari, S., Mithal, V., Polatkan, G. and Ramanath, R., 2021. An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), pp.1-32.

# Attention Mechanism

$$A(q, K, V) = \sum_i p(a(k_i, q)) * v_i$$

Bahdanau, D., Cho, K. and Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

# Generalized Attention Model

$$A(q, K, V) = \sum_i p(\boxed{a(k_i, q)}) * v_i$$

| Function | Equation | References |
|---|---|---|
| similarity | $a(k_i, q) = sim(k_i, q)$ | [Graves et al. 2014a] |
| dot product | $a(k_i, q) = q^T k_i$ | [Luong et al. 2015a] |
| scaled dot product | $a(k_i, q) = \frac{q^T k_i}{\sqrt{d_k}}$ | [Vaswani et al. 2017] |
| general | $a(k_i, q) = q^T W k_i$ | [Luong et al. 2015a] |
| biased general | $a(k_i, q) = k_i(Wq + b)$ | [Sordoni et al. 2016] |
| activated general | $a(k_i, q) = act(q^T W k_i + b)$ | [Ma et al. 2017b] |
| generalized kernel | $a(k_i, q) = \phi(q)^T \phi(k_i)$ | [Choromanski et al. 2021] |
| concat | $a(k_i, q) = w_{imp}^T act(W[q; k_i] + b)$ | [Luong et al. 2015a] |
| additive | $a(k_i, q) = w_{imp}^T act(W_1 q + W_2 k_i + b)$ | [Bahdanau et al. 2015] |
| deep | $a(k_i, q) = w_{imp}^T E^{(L-1)} + b^L$ | [Pavlopoulos et al. 2017] |
| | $E^{(l)} = act(W_l E^{(l-1)} + b^l$ | |
| | $E^{(1)} = act(W_1 k_i + W_0 q) + b^l$ | |
| location-based | $a(k_i, q) = a(q)$ | [Luong et al. 2015a] |
| feature-based | $a(k_i, q) = w_{imp}^T act(W_1 \phi_1(K) + W_2 \phi_2(K) + b)$ | [Li et al. 2019a] |

$a(k_i, q)$: alignment function for query $q$ and key $k_i$, sim : similarity functions such as cosine, $d_k$: length of input, $(W, w_{imp}, W_0, W_1, W_2)$: trainable parameters, b: trainable bias term, act: activation function.

Table 2. Summary of Alignment Functions

Chaudhari, S., Mithal, V., Polatkan, G. and Ramanath, R., 2021. An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), pp.1-32.

# Generalized Attention Model

$$A(q, K, V) = \sum_i p(\boxed{a(k_i, q)}) * v_i$$

| Function | Equation | References |
|---|---|---|
| similarity | $a(k_i, q) = sim(k_i, q)$ | [Graves et al. 2014a] |
| dot product | $a(k_i, q) = q^T k_i$ | [Luong et al. 2015a] |
| scaled dot product | $a(k_i, q) = \frac{q^T k_i}{\sqrt{d_k}}$ | [Vaswani et al. 2017] |
| general | $a(k_i, q) = q^T W k_i$ | [Luong et al. 2015a] |
| biased general | $a(k_i, q) = k_i(Wq + b)$ | [Sordoni et al. 2016] |
| activated general | $a(k_i, q) = act(q^T W k_i + b)$ | [Ma et al. 2017b] |
| generalized kernel | $a(k_i, q) = \phi(q)^T \phi(k_i)$ | [Choromanski et al. 2021] |
| concat | $a(k_i, q) = w_{imp}^T act(W[q; k_i] + b)$ | [Luong et al. 2015a] |
| additive | $a(k_i, q) = w_{imp}^T act(W_1 q + W_2 k_i + b)$ | [Bahdanau et al. 2015] |
| deep | $a(k_i, q) = w_{imp}^T E^{(L-1)} + b^L$ | [Pavlopoulos et al. 2017] |
| | $E^{(l)} = act(W_l E^{(l-1)} + b^l$ | |
| | $E^{(1)} = act(W_1 k_i + W_0 q) + b^l$ | |
| location-based | $a(k_i, q) = a(q)$ | [Luong et al. 2015a] |
| feature-based | $a(k_i, q) = w_{imp}^T act(W_1 \phi_1(K) + W_2 \phi_2(K) + b)$ | [Li et al. 2019a] |

$a(k_i, q)$: alignment function for query $q$ and key $k_i$, sim : similarity functions such as cosine, $d_k$: length of input, $(W, w_{imp}, W_0, W_1, W_2)$: trainable parameters, b: trainable bias term, act: activation function.

Table 2. Summary of Alignment Functions

Chaudhari, S., Mithal, V., Polatkan, G. and Ramanath, R., 2021. An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), pp.1-32.

# Why Attention?

# 注意力机制的特点和优势

1. 注意力机制有助于克服循环神经网络（**RNNs**）的一些挑战，例如输入序列长度增加时性能下降和顺序处理输入导致的计算效率低下。

2. 在自然语言处理（**NLP**）、计算机视觉（**Computer Vision**）、跨模态任务和推荐系统等多个领域中，注意力机制已成为多项任务中的最先进模型，取得了显著的性能提升。

3. 注意力机制不仅可以提高主要任务的性能，还具有其他优势。它们被广泛用于提高神经网络的可解释性，帮助解释模型的决策过程，使得原本被认为是黑盒模型的神经网络变得更易解释。这对于人们对机器学习模型的公平性、可追溯性和透明度的关注具有重要意义。

# Overcoming challenges with RNNs

解决传统编码器-解码器模型的挑战，避免信息损失和无法建模输入输出对齐的问题。

允许解码器访问整个编码的输入序列，通过注意力权重选择性地关注相关信息。

自动学习注意力权重，捕捉编码器和解码器之间的相关性。

构建上下文向量，使解码器能够全面访问输入序列并重点关注相关部分。

提高模型性能，改善输出质量，并提供更好的解释性。

# SOTA for Various Tasks

GPT-4 (no date). Available at: https://openai.com/research/gpt-4.

# SOTA for Various Tasks

| | estimated percentile | estimated percentile | estimated percentile |
|---|---|---|---|
| Uniform Bar Exam (MBE+MEE+MPT)[1] | 298/400<br>~90th | 298/400<br>~90th | 213/400<br>~10th |
| LSAT | 163<br>~88th | 161<br>~83rd | 149<br>~40th |
| SAT Evidence-Based Reading & Writing | 710/800<br>~93rd | 710/800<br>~93rd | 670/800<br>~87th |
| SAT Math | 700/800<br>~89th | 690/800<br>~89th | 590/800<br>~70th |
| Graduate Record Examination (GRE) Quantitative | 163/170<br>~80th | 157/170<br>~62nd | 147/170<br>~25th |
| Graduate Record Examination (GRE) Verbal | 169/170<br>~99th | 165/170<br>~96th | 154/170<br>~63rd |
| Graduate Record Examination (GRE) Writing | 4/6<br>~54th | 4/6<br>~54th | 4/6<br>~54th |
| USABO Semifinal Exam 2020 | 87/150<br>99th–100th | 87/150<br>99th–100th | 43/150<br>31st–33rd |
| USNCO Local Section Exam 2022 | 36/60 | 38/60 | 24/60 |
| Medical Knowledge Self-Assessment Program | 75% | 75% | 53% |
| Codeforces Rating | 392<br>below 5th | 392<br>below 5th | 260<br>below 5th |
| AP Art History | 5<br>86th–100th | 5<br>86th–100th | 5<br>86th–100th |
| AP Biology | 5<br>85th–100th | 5<br>85th–100th | 4<br>62nd–85th |
| AP Calculus BC | 4<br>43rd–59th | 4<br>43rd–59th | 1<br>0th–7th |
| AP Chemistry | 4<br>71st–88th | 4<br>71st–88th | 2<br>22nd–46th |
| AP English Language and Composition | 2 | 2 | 2 |

GPT-4 (no date). Available at: https://openai.com/research/gpt-4.

# Characterisation & interpretability

Guo, M.H., Xu, T.X., Liu, J.J., Liu, Z.N., Jiang, P.T., Mu, T.J., Zhang, S.H., Martin, R.R., Cheng, M.M. and Hu, S.M., 2022. Attention mechanisms in computer vision: A survey. *Computational visual media*, *8*(3), pp.331-368.

(a)

(b)

(c)

(d)

# Characterisation & interpretability

# 变革里程碑：Transformer的崛起

Neural Machine Translation by Jointly Learning to Align and Translate

Prior works | Derivative works | Sponsored by DagsHub

Search... Expand

**Origin paper**
Neural Machine Translation by Jointly Learning to Align and Translate
Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio 2014

Sequence to Sequence Learning with Neural Networks
Ilya Sutskever, Oriol Vinyals, Quoc V. Le 2014

Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation
Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre,... 2014

Attention is All you Need
Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob... 2017

On the Properties of Neural Machine Translation: Encoder–Decoder Approaches
Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau,... 2014

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention
Ke Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C.... 2015

On Using Monolingual Corpora in Neural Machine Translation
Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc... 2015

Bleu: a Method for Automatic Evaluation of Machine Translation
K. Papineni, S. Roukos, T. Ward, Wei-Jing Zhu 2002

Long Short-Term Memory
S. Hochreiter, J. Schmidhuber 1997

Neural Machine Translation of Rare Words with Subword Units
Rico Sennrich, B. Haddow, Alexandra Birch 2015

Universal Vector Neural Machine Translation With Effective Attention
Satish Mylapore, Ryan Quincy Paul, Joshua Yi, Robert D. Slater 2020

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova 2019

Deep Visual-Semantic Alignments for Generating Image Descriptions
A. Karpathy, Li Fei-Fei 2014

**Attention Mechanism**
**打破局面（2014）**

**Transformer 问世（2017）**

Created on May 25 2023

1997 2020

Neural Machine Translation by Jointly Learning to Align and Translate

Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio

2014, International Conference on Learning Representations

22734 Citations Save

Open in:

Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder-decoders and consists of an encoder that encodes a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder-decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. Furthermore, qualitative analysis reveals that the (soft-)alignments found by the model agree well with our intuition.

# Key Components of Attention Modeling Techniques

Chaudhari, S., Mithal, V., Polatkan, G. and Ramanath, R., 2021. An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), pp.1-32.

# Key Components of Attention Modeling Techniques

Chaudhari, S., Mithal, V., Polatkan, G. and Ramanath, R., 2021. An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), pp.1-32.

# 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.

# 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.

# 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.

# 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.

# Transformer : Game Changer

Encoder-decoder architecture with Attention Network

The Transformer - model architecture.

# Transformer : Game Changer

seq-aligned RNNs → self-attention

Encoder-decoder architecture with Attention Network

The Transformer - model architecture.

The
agreement
on
the
European
Economic
Area
was
signed
in
August
1992
.
<end>
L'
accord
sur
la
zone
économique
européenne
a
été
signé
en
août
1992
.
<end>

The
Law
will
never
be
perfect
,
but
its
application
should
be
just
-
this
is
what
we
are
missing
,
in
my
opinion
.
<EOS>
<pad>

# Self-attention

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

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing*

# Self-attention

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

Scaled Dot-Product Attention

# Self-attention

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

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$

$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Scaled Dot-Product Attention

Multi-Head Attention

# Self-attention

# Self-attention

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

# Transformer : Encoder - Decoder

Alammar, J. (no date) The Illustrated Transformer. Available at: http://jalammar.github.io/illustrated-transformer/.

# Transformer : A High-Level Look

Alammar, J. (no date) The Illustrated Transformer. Available at: http://jalammar.github.io/illustrated-transformer/.

# Transformer : A Specific-Level Look

Alammar, J. (no date) The Illustrated Transformer. Available at: http://jalammar.github.io/illustrated-transformer/.

# Transformer : A Specific-Level Look

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

Alammar, J. (no date) The Illustrated Transformer. Available at: http://jalammar.github.io/illustrated-transformer/.

# Transformer : A Specific-Level Look

Alammar, J. (no date) The Illustrated Transformer. Available at: http://jalammar.github.io/illustrated-transformer/.

# Transformer : Multi-Head Attention Visualizaitons

# Transformer : Beat Previous SOTA

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 | **41.29** | $7.7 \cdot 10^{19}$ | $1.2 \cdot 10^{21}$ |
| Transformer (base model) | 27.3 | 38.1 | $\mathbf{3.3 \cdot 10^{18}}$ | |
| Transformer (big) | **28.4** | **41.8** | $2.3 \cdot 10^{19}$ | |

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing*

# Why Transformer Important

**Attention is all you need:**
The first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence- aligned RNNs or convolution

**GPU-friendly parallel computation:**
These layers are highly parallelizable, resulting in lower computational costs

**Sentence-level representations:**
Combined with positional encoding, the Transformer can capture long-range dependencies by considering the relative positions of tokens.

# Open A Foundation Model Era

ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
LARGE LANGUAGE MODEL
LLM has embarked on the path of AGI)
1950's
1960's
1970's
1980's
1990's
2000's
2010's
2020's

Machine Learning
Deep Learning
Foundation Models
Emergence of...
Homogenization of...
"how"
learning algorithms
features
architectures
functionalities
models

# 走向不同：GPT与BERT的选择

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

Prior works | Derivative works | Sponsored by DagsHub

Search... Expand

**Google 团队**
**发布 BERT(2019)**

**OpenAI团队**
**发布GPT-1(2018)**

Origin paper
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova 2019

Attention is All you Need
Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones,... 2017

Deep Contextualized Word Representations
Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark,... 2018

Improving Language Understanding by Generative Pre-Training
Alec Radford, Karthik Narasimhan 2018

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer
Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang,... 2019

GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding
Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R.... 2018

GloVe: Global Vectors for Word Representation
Jeffrey Pennington, R. Socher, Christopher D. Manning 2014

XLNet: Generalized Autoregressive Pretraining for Language Understanding
Zhilin Yang, Zihang Dai, Yiming Yang, J. Carbonell, R. Salakhutdinov, Quoc V. Le 2019

ERNIE: Enhanced Language Representation with Informative Entities
Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, Qun Liu 2019

Neural Machine Translation by Jointly Learning to Align and Translate
Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio 2014

ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators
Kevin Clark, Minh-Thang Luong, Quoc V. Le, Christopher D. Manning 2020

Investigating Meta-Learning Algorithms for Low-Resource Natural Language Understanding Tasks
Zi-Yi Dou, Keyi Yu, Antonios Anastasopoulos 2019

Weighted Transformer Network for Machine Translation
Karim Ahmed, N. Keskar, R. Socher 2017

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

You've used 2/2 free graphs this month. Log in to get 3 more.

Created on Jun 14 2023

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

Jacob Devlin + 2 authors Kristina Toutanova

2019, North American Chapter of the Association for Computational Linguistics

53385 Citations Save

Open in:

We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT 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 BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5 (7.7 point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

**Alec Radford** · 3rd
Head of Research at indico

**Ilya Sutskever** · 3rd
Co-Founder and Chief Scientist at OpenAI

**Jacob Devlin** · 2nd
Software Engineer at Google

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

Prior works | Derivative works | Sponsored by DagsHub

Search... Expand

**Origin paper**
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova 2019

Attention is All you Need
Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones,... 2017

Deep Contextualized Word Representations
Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark,... 2018

Improving Language Understanding by Generative Pre-Training
Alec Radford, Karthik Narasimhan 2018

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer
Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang,... 2019

GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding
Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R.... 2018

GloVe: Global Vectors for Word Representation
Jeffrey Pennington, R. Socher, Christopher D. Manning 2014

XLNet: Generalized Autoregressive Pretraining for Language Understanding
Zhilin Yang, Zihang Dai, Yiming Yang, J. Carbonell, R. Salakhutdinov, Quoc V. Le 2019

ERNIE: Enhanced Language Representation with Informative Entities
Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, Qun Liu 2019

Neural Machine Translation by Jointly Learning to Align and Translate
Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio 2014

ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators
Kevin Clark, Minh-Thang Luong, Quoc V. Le, Christopher D. Manning 2020

Investigating Meta-Learning Algorithms for Low-Resource Natural Language Understanding Tasks
Zi-Yi Dou, Keyi Yu, Antonios Anastasopoulos 2019

Weighted Transformer Network for Machine Translation
Karim Ahmed, N. Keskar, R. Socher 2017

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

You've used 2/2 free graphs this month. Log in to get 3 more.

**Google 团队**
**发布 BERT(2019)**

**OpenAI团队**
**发布GPT-1(2018)**

Created on Jun 14 2023



**Attention Mechanism**

**BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**
Jacob Devlin + 2 authors Kristina Toutanova
2019, North American Chapter of the Association for Computational Linguistics
53385 Citations Save
Open in:

We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT 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 BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5 (7.7 point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

**Alec Radford** · 3rd
Head of Research at indico

**Ilya Sutskever** · 3rd
Co-Founder and Chief Scientist at OpenAI

**Jacob Devlin** · 2nd
Software Engineer at Google

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

Prior works | Derivative works | Sponsored by DagsHub

Search... Expand

**Origin paper**
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova 2019

Attention is All you Need
Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones,... 2017

Deep Contextualized Word Representations
Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark,... 2018

Improving Language Understanding by Generative Pre-Training
Alec Radford, Karthik Narasimhan 2018

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer
Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang,... 2019

GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding
Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R.... 2018

GloVe: Global Vectors for Word Representation
Jeffrey Pennington, R. Socher, Christopher D. Manning 2014

XLNet: Generalized Autoregressive Pretraining for Language Understanding
Zhilin Yang, Zihang Dai, Yiming Yang, J. Carbonell, R. Salakhutdinov, Quoc V. Le 2019

ERNIE: Enhanced Language Representation with Informative Entities
Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, Qun Liu 2019

Neural Machine Translation by Jointly Learning to Align and Translate
Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio 2014

ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators
Kevin Clark, Minh-Thang Luong, Quoc V. Le, Christopher D. Manning 2020

Investigating Meta-Learning Algorithms for Low-Resource Natural Language Understanding Tasks
Zi-Yi Dou, Keyi Yu, Antonios Anastasopoulos 2019

Weighted Transformer Network for Machine Translation
Karim Ahmed, N. Keskar, R. Socher 2017

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

You've used 2/2 free graphs this month. Log in to get 3 more.

**Google 团队**
**发布 BERT(2019)**

**OpenAI团队**
**发布GPT-1(2018)**

Created on Jun 14 2023

**BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**
Jacob Devlin + 2 authors Kristina Toutanova
2019, North American Chapter of the Association for Computational Linguistics
53385 Citations Save
Open in:

We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT 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 BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5 (7.7 point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

**Alec Radford** · 3rd
Head of Research at indico

**Ilya Sutskever** · 3rd
Co-Founder and Chief Scientist at OpenAI

**Jacob Devlin** · 2nd
Software Engineer at Google

Nerual Network
LSTM
Bi-LSTM
Embeding
Word2vec
GloVe
Attention Mechanism
Transformer (Self-attention)
GPT-1
BERT
...
GPT-4

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

**Jacob Devlin** **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

## Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT 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 BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

We argue that current techniques restrict the power of the pre-trained representations, especially for the fine-tuning approaches. The major limitation is that standard language models are unidirectional, and this limits the choice of architectures that can be used during pre-training. For example, in OpenAI GPT, the authors use a left-to-right architecture, where every token can only attend to previous tokens in the self-attention layers

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

**Jacob Devlin** **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

## Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT 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 BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

We argue that current techniques restrict the power of the pre-trained representations, especially for the fine-tuning approaches. The major limitation is that standard language models are unidirectional, and this limits the choice of architectures that can be used during pre-training. For example, in OpenAI GPT, the authors use a left-to-right architecture, where every token can only attend to previous tokens in the self-attention layers

| 基准测试 | 描述 |
| --- | --- |
| GLUE Benchmark | 一个包含九个自然语言理解任务的基准测试，例如：文本蕴含识别、语义角色标注、情感分析等。BERT在多项任务中刷新了记录。 |
| SQuAD v1.1 | 斯坦福问答数据集（Stanford Question Answering Dataset），是一个阅读理解型问答任务的基准测试。BERT在此任务中超过了人类的表现。 |
| SQuAD v2.0 | 与v1.1版本类似，但增加了一部分问题并没有答案的情况，这对模型的理解能力提出了更高的要求。BERT也在此任务中取得了最优成绩。 |
| SWAG | 一个多选问答任务，需要根据前文预测下一句话的内容。BERT在这个任务中也刷新了最好的成绩。 |
| CoNLL-2003 NER | 命名实体识别（Named Entity Recognition）的基准测试，BERT在此任务上表现优异。 |

# BERT : Pre-training + Fine-Tuning Paradigm

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

# BERT 独特价值

**1.全方位上下文理解**：与以前的模型（例如GPT）相比，BERT能够双向理解上下文，即同时考虑一个词的左边和右边的上下文。这种全方位的上下文理解使得BERT能够更好地理解语言，特别是在理解词义、消歧等复杂任务上有明显优势。

**2.预训练+微调（Pre-training + Fine-tuning）的策略**：BERT模型先在大规模无标签文本数据上进行预训练，学习语言的一般性模式，然后在具体任务的标签数据上进行微调。这种策略让BERT能够在少量标签数据上取得很好的效果，大大提高了在各种NLP任务上的表现。

**3.跨任务泛化能力**：BERT通过微调可以应用到多种NLP任务中，包括但不限于文本分类、命名实体识别、问答系统、情感分析等。它的出现极大地简化了复杂的NLP任务，使得只需一种模型就能处理多种任务。

**4.多语言支持**：BERT提供了多语言版本（Multilingual BERT），可以支持多种语言，包括但不限于英语、中文、德语、法语等，使得NLP任务能够覆盖更广的语言和区域。

**5.性能优异**：自BERT模型提出以来，它在多项NLP基准测试中取得了优异的成绩，甚至超过了人类的表现。它的出现标志着NLP领域进入了预训练模型的新时代。

**6.开源和可接入性**：BERT模型和预训练权重由Google公开发布，让更多的研究者和开发者可以利用BERT模型进行相关研究和应用开发，推动了整个NLP领域的发展。

# Pre-training in NLP

- Word embeddings are the basis of deep learning for NLP

- Word embeddings (`word2vec`, `GloVe`) are often *pre-trained* on text corpus from co-occurrence statistics

# History of Contextual Representations

- *Semi-Supervised Sequence Learning*, Google, 2015

**Train LSTM Language Model**

**Fine-tune on Classification Task**

# History of Contextual Representations

- *ELMo: Deep Contextual Word Embeddings*, AI2 & University of Washington, 2017

**Train Separate Left-to-Right and Right-to-Left LMs**

**Apply as "Pre-trained Embeddings"**

# History of Contextual Representations

- *GPT-1: Improving Language Understanding by Generative Pre-Training*, OpenAI, 2018

**Train Deep (12-layer) Transformer LM**

**Fine-tune on Classification Task**

# Problem with Previous Methods

- **Problem**: Language models only use left context *or* right context, but language understanding is bidirectional.
- Why are LMs unidirectional?
- Reason 1: Directionality is needed to generate a well-formed probability distribution.
  - We don't care about this.
- Reason 2: Words can "see themselves" in a bidirectional encoder.

# Unidirectional vs. Bidirectional Models

**Unidirectional context**

Build representation incrementally

**Bidirectional context**

Words can “see themselves”

# Masked LM

- **Solution**: Mask out *k*% of the input words, and then predict the masked words
  - We always use $k = 15\%$

```
                          store            gallon
                            ↑                ↑
the man went to the [MASK] to buy a [MASK] of milk
```

- Too little masking: Too expensive to train
- Too much masking: Not enough context

# Masked LM

- Problem: Mask token never seen at fine-tuning
- Solution: 15% of the words to predict, but don't replace with `[MASK]` 100% of the time.

  Instead:
- 80% of the time, replace with `[MASK]`

  `went to the store → went to the [MASK]`
- 10% of the time, replace random word

  `went to the store → went to the running`
- 10% of the time, keep same

  `went to the store → went to the store`

# Next Sentence Prediction

- To learn *relationships* between sentences, predict whether Sentence B is actual sentence that proceeds Sentence A, or a random sentence

**Sentence A** = `The man went to the store.`
**Sentence B** = `He bought a gallon of milk.`
**Label** = `IsNextSentence`

**Sentence A** = `The man went to the store.`
**Sentence B** = `Penguins are flightless.`
**Label** = `NotNextSentence`

# BERT vs GPT
# 共识

| | |
|---|---|
| 模型架构 | **Transformer** |
| 数据预处理 | 都需要对数据进行**Tokenization**，一般使用词片方法 **(Subword Tokenization)** |
| 模型训练 | 均使用了大量的无标签数据进行预训练 |
| 任务迁移 | 都可以通过**Fine-tuning**方式进行任务迁移 |
| 训练目标 | 都试图通过预训练理解语言的一般模式，如语法、语义、上下文关系等 |
| 多语言支持 | 均支持多语言模型训练 |

# BERT vs GPT 差异

| 特性 | BERT | GPT |
| --- | --- | --- |
| 训练方式 | 自编码（Autoencoding） | 自回归（Autoregressive） |
| 预测目标 | 给定上下文，预测其中的一个或多个缺失单词 | 在给定前面的单词时，预测下一个单词 |
| 输入处理 | 双向，可以同时考虑一个词的左右上下文 | 单向（从左到右或者从右到左） |
| 适用场景 | 适合理解上下文，有助于信息提取、问答系统、情感分析等 | 适合生成式任务，如文章生成、诗歌创作等 |
| 架构 | 基于Transformer的编码器 | 基于Transformer的解码器 |
| 语言模型 | 判别式（Discriminative） | 生成式（Generative） |
| 优点 | 对上下文理解能力较强 | 预测的连贯性较强 |
| 缺点 | 生成的文本连贯性较弱 | 对上下文理解能力相对较弱 |

# THANKS

极客时间 | 训练营