A Survey on GPT Technical Applications

Chat GPT & DaKES

【Abstract】Generative Pre-trained Transformer (GPT) models have led to significant advancements in the field of natural language processing (NLP) and artificial intelligence (AI). This paper presents a comprehensive survey on the GPT technical applications, including text generation, sentiment analysis, machine translation, and multimodal AI. Then we probe into three extendable clues: GPT+X, X+GPT and Auto-GPT.

1. GPT Technology: A brief overview

Generative Pre-trained Transformer (GPT) models are a class of language models that have significantly advanced the field of natural language processing (NLP) and artificial intelligence (AI). These models leverage the Transformer architecture and adopt pre-training and fine-tuning strategies to achieve state-of-the-art performance in a wide range of NLP tasks.

2.1 Transformer Architecture (Vaswani et al., 2017)

The Transformer architecture, introduced by Vaswani et al. (2017), is the foundation of GPT models. It is designed to address the limitations of recurrent neural networks (RNNs) and convolutional neural networks (CNNs) in handling long-range dependencies and parallelization. The core component of the Transformer architecture is the self-attention mechanism, which allows the model to weigh the importance of each input token with respect to other tokens in the sequence.

A Transformer model consists of an encoder and a decoder, both composed of multiple layers of self-attention and feed-forward neural networks. The encoder processes the input sequence, while the decoder generates the output sequence. Each layer in the encoder and decoder is comprised of multi-head self-attention, followed by layer normalization, position-wise feed-forward networks, and another layer normalization.

One of the key innovations in the Transformer architecture is the incorporation of positional encoding, which enables the model to capture the order of tokens in the input sequence. This is achieved by adding a sinusoidal function to the input embeddings, allowing the model to discern the relative positions of tokens in the sequence.

2.2. Pre-training and Fine-tuning Strategies (Radford et al., 2018)

GPT models employ pre-training and fine-tuning strategies to learn language representations and adapt them to specific tasks. The pre-training phase involves training the model on a large corpus of unannotated text, allowing it to learn general language patterns and representations. During this phase, GPT models are trained using a masked language modeling objective, where a portion of the input tokens is masked, and the model must predict the masked tokens based on the context provided by the unmasked tokens.

After the pre-training phase, the model is fine-tuned on smaller, task-specific datasets to adapt its learned representations to the target task. Fine-tuning involves updating the model's parameters with supervised learning, using labeled data from the target task. This process allows GPT models to achieve high performance on various NLP tasks with relatively small amounts of labeled data, as the pre-training phase provides a strong foundation of language understanding.

Radford et al. (2018) introduced the first GPT model, which demonstrated the effectiveness of the Transformer architecture and the pre-training and fine-tuning strategy. Subsequent GPT models (GPT-2, GPT-3, and GPT-4) built on these foundations, achieving state-of-the-art performance in a wide array of NLP tasks by scaling the model size and leveraging ever-larger pre-training datasets.

2. GPT Technical Applications: Some examples

Below are some examples of GPT technical applications across various domains, such as text generation, sentiment analysis, machine translation, and multimodal AI.

2.1 Text Generation

Content summarization: GPT models can automatically generate summaries of long articles, books, or documents, condensing the main ideas into a concise format. This capability can be particularly useful for news agencies, researchers, and professionals who need to process large amounts of textual information quickly.

Email drafting and auto-completion: GPT models can be utilized to draft emails or provide context-aware auto-completion suggestions. By predicting the most likely words or phrases to follow a given input, GPT models can help users save time and effort when composing messages.

2.2 Sentiment Analysis

Social media monitoring: Companies can use GPT models to analyze user-generated content on social media platforms, such as tweets, comments, and reviews, to understand public sentiment about their products, services, or brand image. This information can be valuable for marketing, public relations, and customer service teams.

Financial market analysis: GPT models can be employed to analyze news articles and financial reports to predict market sentiment and potential stock price movements. This can help investors and financial institutions make more informed decisions and manage risk more effectively.

2.3 Machine Translation

Real-time translation: GPT models can be used for real-time translation of text or speech in various languages, enabling seamless communication between people who speak different languages. This can have applications in international business, diplomacy, travel, and customer support.

Language preservation: GPT models can be employed to develop translation systems for low-resource languages, helping to preserve linguistic diversity and support communication within and between minority language communities.

2.4 Multimodal AI

Image captioning: GPT models can be combined with computer vision techniques to generate natural language descriptions of images. This can be useful for accessibility (e.g., creating alternative text for visually impaired users), content management, or image search applications.

Video summarization: By integrating GPT models with video processing algorithms, AI systems can automatically generate summaries or transcripts of video content, making it easier for users to navigate and understand large video collections.

3. Three Extendable Paths

At present, there are three obvious paths for extending the applications of GPT technology, as follows.

3.1 GPT+X

In the GPT+X mode, X represents the applications that GPT lacks but already exist in reality. For example: in order to solve the problem that ChatGPT cannot connect to the Internet, traditional search engines are added to it, such as ChatGPT+Bing, Bard+Google, Wenxinyiyan+Baidu, etc.; due to the limited transmission of Token and context, vertical It can be customized in different domains, such as PDF+GPT (ChatPDF), Blog+GPT (Glasp), Web+GPT (Glarity), etc.; ChatGPT has no long-term memory, and if you forget it after chatting once, add an external storage, such as Vector databases (Milvus, Pinecone), graph databases (Data Graph), etc.

GPT+X is essentially improving the ability of GPT to "chat", making it more accurate, longer, and more valuable. It hopes to use the basic interaction mode of human-machine chat to complete more application processing.

3.2 X+GPT

The simple case of X+GPT mode is to add GPT to existing applications to make the original capabilities more powerful. Such as Office 365 Copliot in the office field, Notion.AI and Miro.AI in the note and drawing field, Copilot X in the programming IDE, etc.

The more complicated X+GPT mode integrates ChatGPT and existing applications or interfaces through traditional workflow engines, including RPA or API Platform, to form a "task splitter + GPT". For example, Uipath/Automation Anywhere + ChatGPT, ChatGPT Plugin + Zapier, etc., due to the strong integration capabilities of X itself, X+GPT can play a greater role.

Another X+GPT mode is to use GPT as a language engine to use its own language bonding effect to play a role. At present, the industry has developed two typical operation and integration frameworks:

(1) HuggingGPT

HuggingGPT is a system designed to solve AI tasks, using language as an interface to connect large language models (LLM) with various AI models. The main idea is to use LLM as a controller to manage the AI model and utilize models from communities like Hugging Face to solve different user requests. It first applies LLM to understand user requests and decomposes them into small tasks, then assigns these tasks to different functional models for completion, and finally uses LLM again to aggregate the results into the final output. Its workflow is:

Task planning: LLM understands user requests and breaks them down into subtasks as needed.

Model selection: LLM selects the most appropriate expert model according to the description of each subtask.

Task Execution: HuggingGPT performs a task using selected models, integrating their results to provide a comprehensive response.

Generating responses: LLM integrates the calculation results of all models to generate responses for users.

Microsoft has released JARVIS based on HuggingGPT, which operates as an AI controller system and can simultaneously control multiple AI models to complete multi-modal and multi-task.

(2) BabyAGI

BabyAGI is a simplified version of task-driven autonomous agent based on the open source framework LongChain. Task-driven autonomous agent is an intelligent system that uses OpenAI's GPT-4 language model, Pinecone vector search and LangChain wide range of tasks in the field. There are four main modules, including completing tasks, generating new tasks based on completed results, prioritizing tasks in real time, historical tasks, and result storage, all within constraints and contexts. Its workflow is:

The user gives the goal requirement and the first task, which is also the task to be performed currently

Putting tasks into a safe agent helps ensure that the inputs and outputs generated by the system are ethical and safe, reducing the risk of unintended consequences

The security agent returns the result of security compliance

Use the ability of ChatGPT as a task generator Task Generator to generate a new task New Task and put it into the task list Tasklist

Arrange task priorities according to rules or external requirements Task Prioritization

Generate the current task to be executed in order of priority Current Task

HuggingGPT and BabyAGI are somewhat similar. Jarvis is like a task decomposer, while LangChain is more like a task executor. They both need to solve the problem of accuracy and error rate in task execution. The deviation and superposition of error rates of different tasks will give the results with disastrous consequences. This application mode can be understood as "GPT + task manager + task executor". GPT is no longer just a simple interactive Chat tool, but an AI "router" for task judgment, task decision-making, task routing, and task inspection.

3.3 Auto-GPT

Auto-GPT is an experimental open source application that is powered by GPT-4 and can autonomously achieve any goal set by the user, so Auto-GPT is an experimental autonomous application built on the GPT-4 language model.

Auto-GPT aims to demonstrate the potential of GPT-4 by providing various functions, such as Internet access for search and information gathering, long-term and short-term memory management, text generation using GPT-4 instances, access to popular websites and platforms, and using GPT-3.5 for file storage and aggregation. And these AI-Agents will automatically run and complete tasks by themselves, and will perform recursive debugging, development and self-improvement.

GitHub address: https://github.com/torantulino/auto-gpt

Auto-GPT is equivalent to giving a GPT-based model a memory and a body, so a task can be handed over to the AI-Agent, allowing it to propose a plan autonomously and then execute the plan. It also features internet access, long- and short-term memory management, GPT-4 instances for text generation, and GPT-3.5 for file storage and summarization. But there are also the following problems:

(1) High cost:

Since Auto-GPT's task is accomplished through a chain of ideas, each step involves invoking the expensive GPT-4 model, which typically maximizes tokens to provide better reasoning and hints.

The GPT-4 token is not cheap. It is initially estimated that the total cost of each step of GPT-4 is about 0.288 US dollars. Auto-GPT requires an average of 50 steps to complete a small task. The cost of completing a task is about $14.4.

(2) Development and production are not separated

Because Auto-GPT's process of decomposing and generating tasks is mixed with the process of executing tasks, it means that it costs $14.4 per run. In fact, task decomposition is more like the design and definition of a process, which belongs to the category of process design

Auto-GPT has no way to "serialize" the tasks on the operation chain as a reusable function. Therefore, every time the user wants to solve a problem, he has to develop this task chain from scratch.

(3) Trapped in a self-circulation

Perhaps the above two problems are easy to solve, but Auto-GPT will be stuck in a loop, which is the key to restricting its development. Many users have reported that Auto-GPT often gets stuck in a loop that prevents it from solving real problems rather than accomplishing the goals requested by users.

4. Discussion

Other issues include:

(1) GPT in Multimodal Applications

In addition to text-based applications, GPT technology has been adapted to incorporate multimodal data, such as images and audio. It is necessary to explore the integration of GPT models with other AI and ML techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to enable more comprehensive understanding and synthesis of multimodal data.

(2) Ensuring Robustness and Security in GPT Applications

As GPT models become increasingly integrated into critical systems, ensuring their robustness and security is essential. It is valuable to investigate the techniques used to improve the robustness of GPT models against adversarial attacks and explore methods for ensuring secure deployment in sensitive applications.

5. Conclusion

To sum up, the three paths of GPT+X, X+GPT and Auto-GPT to extend the applications of GPT technology have their own advantages and disadvantages, and they are expected to be perfected in the subsequent development, so as to prosper the applications of GPT technology.

References

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 33.

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

OpenAI. (2022). GPT-4: The State-of-the-Art in Natural Language Processing. OpenAI Blog.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. OpenAI Blog.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30, 5998-6008.

GPT技术应用概览

Chat GPT & DaKES

【摘要】生成式预训练 Transformer (GPT) 模型在自然语言处理 (NLP) 和人工智能 (AI) 领域取得了重大进展。 本文对 GPT 技术应用进行了全面的调查，包括文本生成、情感分析、机器翻译和多模态 AI。然后我们探讨三个可扩展的路径：GPT+X、X+GPT和Auto-GPT。【中文版非英文版的完全对应】

1. GPT 技术：简要概述

生成式预训练 Transformer (GPT) 模型是一类语言模型，极大地推动了自然语言处理 (NLP) 和人工智能 (AI) 领域的发展。 这些模型利用 Transformer 架构并采用预训练和微调策略，以在广泛的 NLP 任务中实现最先进的性能。

1.1 Transformer 架构（Vaswani 等人，2017 年）

Transformer 架构，由 Vaswani 等人介绍。 (2017)，是 GPT 模型的基础。 它旨在解决循环神经网络 (RNN) 和卷积神经网络 (CNN) 在处理远程依赖性和并行化方面的局限性。 Transformer 架构的核心组件是自注意力机制，它允许模型权衡每个输入标记相对于序列中其他标记的重要性。

Transformer 模型由编码器和解码器组成，两者均由多层自注意力和前馈神经网络组成。 编码器处理输入序列，而解码器生成输出序列。 编码器和解码器中的每一层都由多头自注意力、层归一化、位置前馈网络和另一层归一化组成。 Transformer 架构的一项关键创新是结合了位置编码，这使模型能够捕获输入序列中标记的顺序。 这是通过向输入嵌入添加正弦函数来实现的，允许模型识别序列中标记的相对位置。

1.2 预训练和微调策略（Radford 等人，2018）

GPT 模型采用预训练和微调策略来学习语言表示并使其适应特定任务。 预训练阶段涉及在大量未注释文本的语料库上训练模型，使其能够学习一般语言模式和表示。 在此阶段，GPT 模型使用掩蔽语言建模目标进行训练，其中一部分输入标记被掩蔽，并且模型必须根据未掩蔽标记提供的上下文预测掩蔽标记。

预训练阶段之后，模型在较小的、特定于任务的数据集上进行微调，以使其学习的表示适应目标任务。 微调涉及通过监督学习更新模型的参数，使用来自目标任务的标记数据。 此过程使 GPT 模型能够以相对少量的标记数据在各种 NLP 任务上实现高性能，因为预训练阶段为语言理解提供了坚实的基础。

雷德福等人。 (2018) 介绍了第一个 GPT 模型，证明了 Transformer 架构的有效性以及预训练和微调策略。 随后的 GPT 模型（GPT-2、GPT-3 和 GPT-4）建立在这些基础上，通过扩展模型大小和利用越来越大的预置模型，在广泛的 NLP 任务中实现了最先进的性能。 训练数据集。

2. GPT 技术应用：一些例子

以下是 GPT 技术在各个领域的一些应用示例，例如文本生成、情感分析、机器翻译和多模式 AI。

2.1 文本生成

文本摘要：GPT 模型可以自动生成长文章、书籍或文档的摘要，将主要思想浓缩成简洁的格式。 此功能对于需要快速处理大量文本信息的新闻机构、研究人员和专业人员特别有用。

电子邮件起草和自动完成：GPT 模型可用于起草电子邮件或提供上下文感知的自动完成建议。 通过预测最有可能跟随给定输入的单词或短语，GPT 模型可以帮助用户在撰写消息时节省时间和精力。

2.2 情感分析

社交媒体监控：公司可以使用 GPT 模型分析社交媒体平台上用户生成的内容，例如推文、评论和评论，以了解公众对其产品、服务或品牌形象的看法。 此信息对于营销、公共关系和客户服务团队可能很有价值。

金融市场分析：GPT 模型可用于分析新闻文章和财务报告，以预测市场情绪和潜在的股价走势。 这可以帮助投资者和金融机构做出更明智的决策并更有效地管理风险。

2.3 机器翻译

实时翻译：GPT 模型可用于各种语言的文本或语音的实时翻译，实现不同语言的人之间的无缝交流。 这可以应用于国际商务、外交、旅游和客户支持。

语言保护：GPT 模型可用于开发低资源语言的翻译系统，有助于保护语言多样性并支持少数民族语言社区内部和之间的交流。

2.4 多模态人工智能

Image captioning：GPT 模型可以与计算机视觉技术相结合，生成图像的自然语言描述。 这对于可访问性（例如，为视障用户创建替代文本）、内容管理或图像搜索应用程序很有用。

视频摘要：通过将 GPT 模型与视频处理算法相结合，AI 系统可以自动生成视频内容的摘要或抄本，使用户更容易浏览和理解大型视频集。

3. 扩展 GPT 技术应用的三条路径

目前明显的可以扩展 GPT 技术应用的三条路径如下。

3.1 GPT+X

在 GPT+X 模式中，X 代表 GPT 所缺、而现实已有的应用。实例如：为了解决 ChatGPT 不能连接互联网的问题，就给它加上传统的搜索引擎，诸如 ChatGPT+Bing，Bard+Google，文心一言+百度等；由于传输 Token 和上下文受限，就按照垂直领域给它来个性化定制，例如 PDF+GPT（ChatPDF），Blog+GPT（Glasp），Web+GPT（Glarity）等；ChatGPT 没有长期记忆，聊过一次就忘，那就加个外部存储，例如向量数据库（Milvus，Pinecone），图数据库看（Data Graph）等。

GPT+X 本质上是在提升 GPT"聊"的能力，让它聊的更准确、更长久、更有价值，希望利用人与机器聊天这个基本交互模式，完成更多的应用处理。

3.2 X+GPT

X+GPT 模式的简单情形是在已有应用上添加 GPT，让原能力变得更强大。如办公领域的 Office 365 Copliot，笔记和绘图领域的 Notion.AI 和 Miro.AI，编程 IDE 的 Copilot X 等。

复杂一些的 X+GPT 模式是通过传统的工作流引擎，也包括 RPA 或者 API Platform 来集成 ChatGPT 和已有的应用或接口，构成"任务分解器+GPT"。例如 Uipath/Automation Anywhere + ChatGPT，ChatGPT Plugin + Zapier 等，由于 X 本身的强大集成能力，可让 X+GPT 发挥更大作用。

另一种 X+GPT 模式是通过 GPT 作为语言引擎，利用其自身具备的语言粘合效应发挥作用。目前业界研发了两个典型的运行和集成框架：

(1) HuggingGPT

HuggingGPT 是一个旨在解决 AI 任务的系统，它使用语言作为接口，将大型语言模型 (LLM) 与各种 AI 模型连接起来。主要思想是使用 LLM 作为控制器来管理 AI 模型，并利用来自 Hugging Face 等社区的模型来解决不同的用户请求。它首先应用 LLM 来理解用户请求并将其分解成小任务，然后将这些任务分配给不同的功能模型来完成，最后再次使用 LLM 将结果汇总为最终输出。其工作流程为：

任务规划：LLM 了解用户的请求，并根据需要将其分解为子任务。

模型选择：LLM 根据每个子任务的描述选择最合适的专家模型。

任务执行：HuggingGPT 使用选定的模型执行任务，整合它们的结果以提供全面的响应。

生成回复：LLM 整合所有模型的计算结果，为用户生成回答。

微软已基于 HuggingGPT 发布 JARVIS，作为 AI 控制器系统运行，可以同时控制多个 AI 模型以完成多模态多任务。

(2) BabyAGI

BabyAGI 是基于开源框架 LongChain 的 task-driven autonomous agent 简化版, Task-driven autonomous agent 任务驱动的自主代理是一个智能系统，它利用 OpenAI 的 GPT-4 语言模型、Pinecone 矢量搜索和 LangChain 框架来执行跨不同领域的广泛任务。主要有四个模块，包括完成任务、根据完成的结果生成新任务、实时确定任务的优先级、历史任务和结果存储，所有这些都在约束和上下文中进行。其工作流程为：

用户给出目标要求和第一个任务，也是当前要执行的任务

将任务放入一个安全代理，安全代理有助于确保系统生成的输入和输出符合道德和安全准则，降低意外后果的风险

安全代理返回安全合规的结果

利用 ChatGPT 的能力作为任务生成器 Task Generator，生成新的任务 New Task，放入任务清单 Tasklist

依据规则或者外部要求，安排任务优先级 Task Prioritization

按照优先级顺序生成当前要执行的任务 Current Task

HuggingGPT 和 BabyAGI 有些相似，Jarvis 像是个任务分解器，而 LangChain 更像是任务执行器，它们都需要解决任务执行中的准确度和错误率的问题，不同任务的错误率的偏差和叠加，都会给结果带来灾难性的后果。这种应用模式可理解为“GPT+任务管理器+任务执行器”，GPT 不再只是提供简单交互的 Chat 工具，而成为了任务判断、任务决策，、任务路由、任务检查的 AI“路由器”。

3.3 Auto-GPT

Auto-GPT 是一个实验性开源应用程序，该程序由 GPT-4 驱动，可以自主实现用户设定的任何目标，因此 Auto-GPT 是一个基于 GPT-4 语言模型构建的实验性自治应用程序。

Auto-GPT 旨在通过提供各种功能来展示 GPT-4 的潜力，例如用于搜索和信息收集的互联网访问、长期和短期记忆管理、使用 GPT-4 实例生成文本、访问流行的网站和平台，以及使用 GPT-3.5 进行文件存储和汇总。而这些 AI-Agent 会自行自动运行并完成任务, 会进行递归调试、开发和自我改进。

GitHub 地址：https://github.com/torantulino/auto-gpt

Auto-GPT 相当于给基于 GPT 的模型一个内存和一个身体，于是可以把一项任务交给 AI-Agent，让它自主地提出一个计划，然后执行计划。此外其还具有互联网访问、长期和短期内存管理、用于文本生成 GPT-4 实例以及使用 GPT-3.5 进行文件存储和生成摘要等功能。但也存在以下问题：

(1) 高昂的成本：

由于 Auto-GPT 的任务是通过一连串的想法完成的，每一步都需要调用昂贵的 GPT-4 模型，该模型通常最大化令牌，以提供更好的推理和提示。

而 GPT-4 令牌并不便宜，初步估算 GPT-4 的每步执行的总成本约 0.288 美元，Auto-GPT 平均需要 50 个步骤来完成一个小任务，为此完成一项任务的成本约是 14.4 美元。

(2) 开发和生产未分离

因为 Auto-GPT 分解和产生任务的过程和任务的执行过程混合在了一起，也就意味着每次运行它都需要14.4美元。事实上，任务分解更像是个流程的设计和定义，属于流程设计的范畴

Auto-GPT 没有办法将操作链上的任务“序列化”作为可重用的功能。因此，用户每次想要解决问题时，都必须从头开始开发这个任务链。

(3) 陷入自我循环

或许以上两个问题还容易解决，但 Auto-GPT 会循环卡死，这才是制约它发展的关键。许多用户报告说，Auto-GPT 经常陷入循环，使其无法解决实际问题，而无法完成用户要求的目标。

4. 讨论

其它问题有：

(1) GPT 的多模态应用

除了基于文本的应用，GPT 技术已经适应了整合多模态数据，如图像和音频。有必要探讨将 GPT 模型与其他 AI 和 ML 技术（如卷积神经网络（CNN）和循环神经网络（RNN））相结合，以实现对多模态数据更全面的理解和综合。

(2) 确保 GPT 应用的鲁棒性和安全性

随着 GPT 模型日益融入关键系统，确保其鲁棒性和安全性至关重要。值得研究提高 GPT 模型抵抗对抗攻击的鲁棒性的技术，并探讨了确保敏感应用中安全部署的方法。

5. 小结

综上，扩展 GPT 技术应用的 GPT+X、X+GPT 和 Auto-GPT 三条路径各有优劣，可望在后续发展中各自完善，从而繁荣 GPT 技术应用。

参考文献

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 33.

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

OpenAI. (2022). GPT-4: The State-of-the-Art in Natural Language Processing. OpenAI Blog.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. OpenAI Blog.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30, 5998-6008.