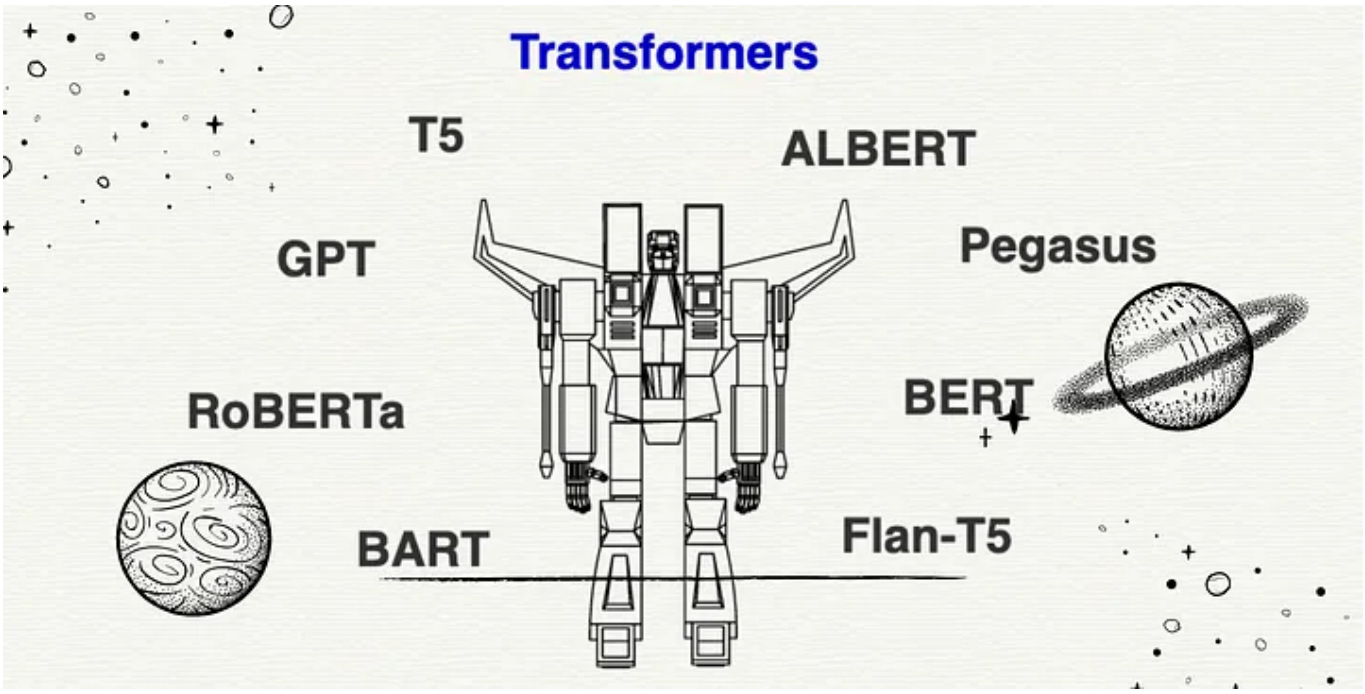
Member-only story

ChatGPT and Other Transformers: How to Select Large Language Model for Your NLP Projects

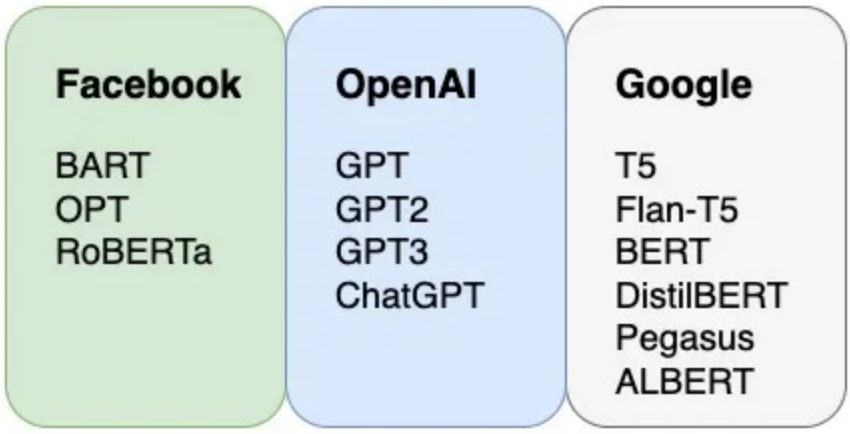
Three types of transformers: Encoder model, decoder model, and sequence-to-sequence model

Alina Zhang · Follow
5 min read · Feb 21



All images in the article are created by author

We are in the golden age of natural language processing. Since 2018, the born of GPT and BERT, a gang of transformers has emerged and gradually become the workhorse in industry. The star players of large language models are shown in the following figure:



Star players of large language models.

But how to know which language model would perform best on your task? How to choose the best model for your NLP project?

The common NLP projects are text classification, text summarization, question answering, text generation, named entity recognition, sentiment analysis, language modeling, and translation. To answer which model would be the best candidate, we need to understand three problems first:

1. Why transformers are so powerful?
2. What are the three types of transformers?
3. What are the advantages of different types of large language models?

Why transformers like GPT, BERT, and BART are so powerful — standing on the shoulders of giants

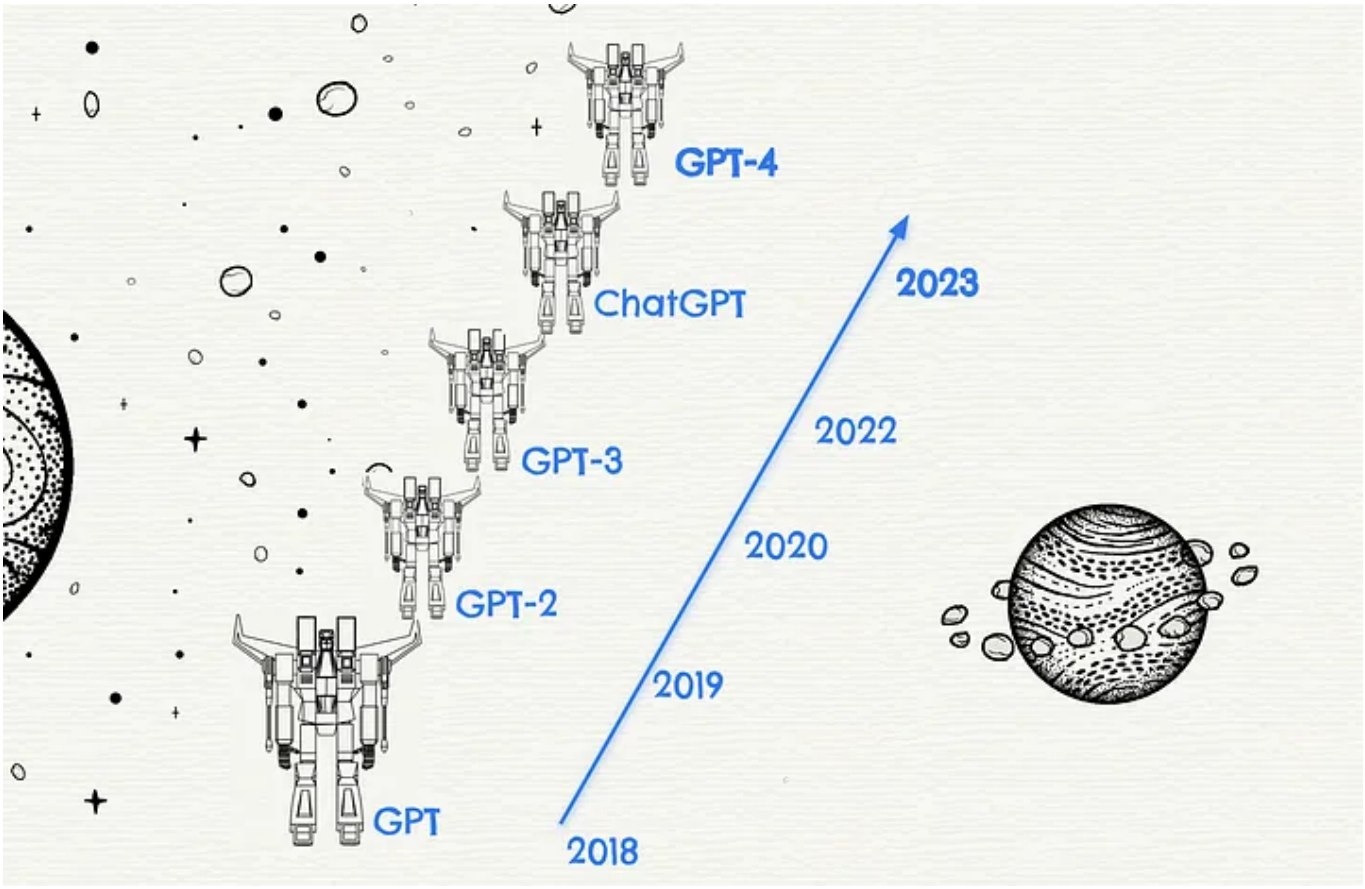
Whether your father is a scientist or he dropped out of high school, you have to start your learning from scratch. Because you cannot inherit knowledge directly from your father or another brain.

But what if the knowledge from your father was transferred to you a hundred percent when you were born? What if you are a 1-month-old scientist? This would significantly save lots of time humans spend on education and boost the development of science and technology.

A transformer is a deep learning model with a large number of layers. The combination of attention mechanism, parallelizable computation, and transfer learning makes transformer a powerful tool. But the most unique part of its architecture is the transfer learning.

In transfer learning, we can borrow the first several layers of a pre-trained model which has already learned some useful representations of the data. Instead of having to learn these representations from scratch, we can just transfer the knowledge directly from the pre-trained model and then train the subsequent layers of the neural network on our specific tasks.

The original GPT was trained on a diverse corpus that included over 40GB of web pages, books, and other sources of text data. Its descendants such as GPT-2, GPT-3, ChatGPT, and the coming GPT-4 just need to be fine-tuned on additional text data instead of starting from zero. Transfer learning is like standing on the shoulders of giants.



Transfer learning: Standing on the shoulders of giants

What are the three types of transformers

There are three types of transformers:

- Encoder models
- Decoder models
- Encoder-decoder models / Sequence-to-sequence models

Encoder models use only the encoder of a transformer model. The encoder model takes an input text sequence and transforms it into a fixed-length representation which is a vector of values for each word of the initial sequence. The numerical representation is affected by the context so that

the same word has different representations when the words around it are different.

For example, 'She walked along the river bank' and 'He works for Citi bank' both contain word *bank*. But the vector representation of *bank* is different in these two sentences.

At each stage during training, **the attention layers can access all the words in the initial sentence**. For example, "She moved to San Francisco this year", all words in the sentences are visible for the word *Francisco* during training.

Decoder models use only the decoder of a transformer model. At each stage, **for a given word the attention layers can only access the words before it in the sentence.** For example, with the initial sequence "She moved to San Francisco this year", only 'She moved to San' is visible for the word *Francisco* during training.

Sequence-to-sequence models, also called encoder-decoder models, use both parts of the transformer architecture. The encoder component takes care of understanding the sequence while the decoder is in charge of generating a sequence according to the understanding of the encoder. More specifically, the decoder component uses representations from the encoder and the word it just generated to generate the next word.

What are the advantages of different types of large language models

Encoder Model	text classification named entity recognition sentiment analysis extractive question answering	BERT DistilBERT RoBERTa ALBERT
Decoder Model	casual language modeling generating sequences	GPT families CTRL Transformer XL
Sequence-to-sequence Model	text summarization translation generative question answering	Pegasus T5 FLAN-T5 BART mBART

The three types of transformers and their best-suited tasks

How to select the ideal model for your project?

If you are working on **text classification, named entity recognition, sentiment analysis, and extractive question answering**, encoder models would be your best choice. Encoder models, with bi-directional attention, are good at understanding the full sequence and the interdependence between words. The prominent encoder models are BERT, DistilBERT, RoBERTa, ALBERT, and so on.

Decoders, with uni-directional attention, are best suited for **casual language modeling and generating sequences.** Representatives of decoder models include GPT families, CTRL, and Transformer XL.

Sequence-to-sequence models are good at generating new sentences depending on a given input, such as text summarization, translation, or

generative question answering. The state-of-the-art sequence-to-sequence models include Pegasus, T5, FLAN-T5, BART, and mBART.

Resources for training transformers

There are two popular resources for training transformers: Hugging Face and Fast.ai. Hugging Face provides APIs for fine-tuning the pre-trained models on your own datasets, as well as tools for evaluating and sharing custom models.

Fast.ai is a more general-purpose library for building deep learning models, such as computer vision and NLP while Hugging Face is focusing on NLP tasks. Fast.ai offers a simplified interface for developers to work with PyTorch and TensorFlow to fine-tune pre-trained models or build custom models.

Congratulations! You have learned about the types of transformers and how to select the best language model for a wide range of NLP tasks. Natural language processing with transformers is the next big thing. 80% of the world's data is unstructured and the percentage is keeping increasing. We expect to see the explosive growth of technologies similar to GPT and their commercial applications. Keep learning to stay ahead of the curve. 👍🤣🌊

References:

- Transformers for Natural Language Processing — Second Edition by Denis Rothman
- Hugging Face official transformer course
- Natural Language Processing with Transformers, Revised Edition by Lewis Tunstall, Leandro von Werra, Thomas Wolf

Machine Learning | Data Science | NLP | Deep Learning | Python

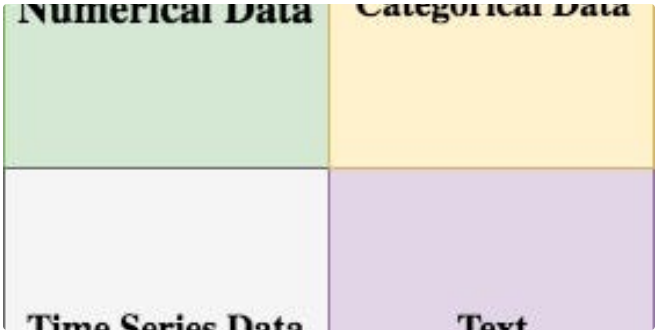
Written by Alina Zhang

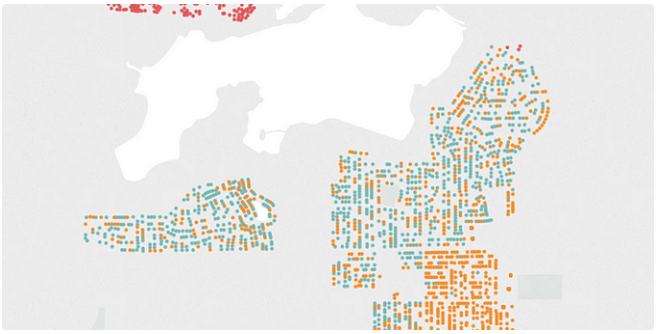
Follow

1.1K Followers

Data Scientist: Keep it simple. https://lnkd.in/gjwc233a

More from Alina Zhang





Alina Zhang in Towards Data Science

Data Types From A Machine Learning Perspective With...

Almost anything can be turned into DATA. Building a deep understanding of the differe...

4 min read · Aug 17, 2018

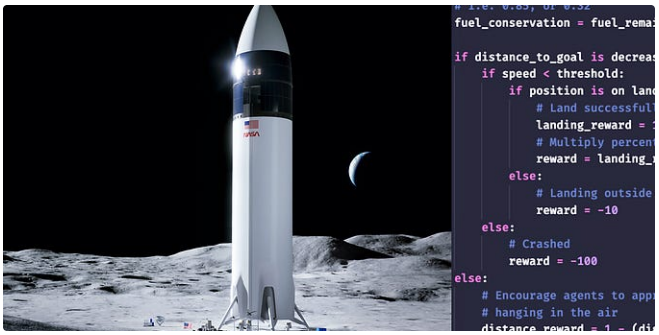
197 6

Alina Zhang in Towards Data Science

Visualize Geographic Data Using Longitude and Latitude Values in...

3 min read · Sep 5, 2018

52 1



Alina Zhang in Towards Data Science

How to Design a Reinforcement Learning Reward Function for a...

Imagine aliens attacked and you were trying to land a Lander on the Moon, what...

3 min read · Aug 18, 2021

36 2

Alina Zhang in Towards Data Science

How to Flatten Deeply Nested JSON Objects in Non-Recursive Elegant...

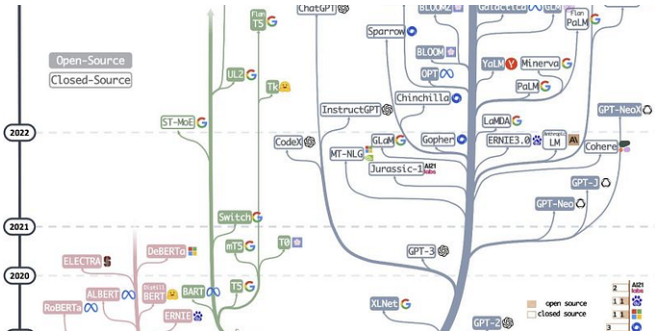
It is dangerous to flatten deeply nested JSON objects with a recursive python solution....

3 min read · Dec 2, 2018

598 7

See all from Alina Zhang

Recommended from Medium



Haifeng Li

A Tutorial on LLM

Generative artificial intelligence (GenAI), especially ChatGPT, captures everyone's...

15 min read · Sep 14

372

David Shapiro

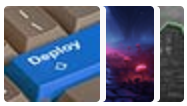
A Pro's Guide to Finetuning LLMs

Large language models (LLMs) like GPT-3 and Llama have shown immense promise for...

12 min read · Sep 23

283 6

Lists



Predictive Modeling w/ Python

20 stories · 452 saves

Practical Guides to Machine Learning

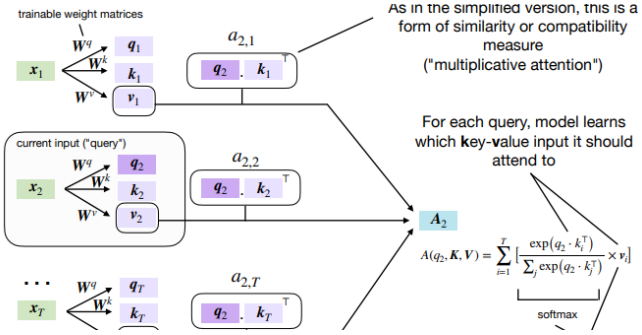
10 stories · 519 saves

Natural Language Processing

669 stories · 283 saves

Coding & Development

11 stories · 200 saves

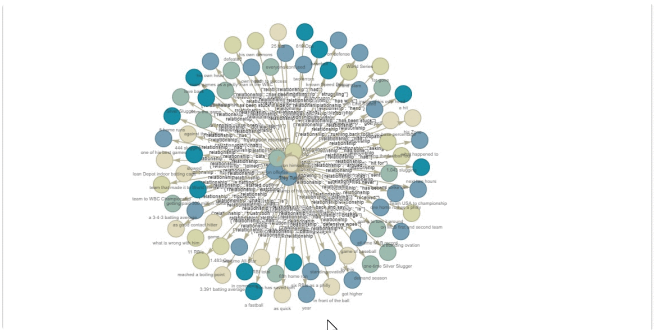


Zain ul Abideen

Attention Is All You Need: The Core Idea of the Transformer

An overview of the Transformer model and its key components.

6 min read · Jun 26



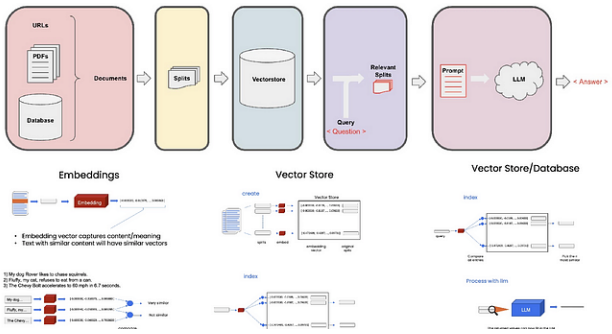
Wenqi Glantz in Better Programming

7 Query Strategies for Navigating Knowledge Graphs With...

Exploring NebulaGraph RAG Pipeline with the Philadelphia Phillies

· 17 min read · 4 days ago

144 501 4



Ryan Nguyen in Towards AI

So, You Want To Improve Your RAG Pipeline

Ways to go from prototype to production with LlamaIndex

· 9 min read · Sep 27

176 2

TeeTracker

Chat with your PDF (Streamlit Demo)

Conversation with specific files

4 min read · Sep 15

56

See more recommendations