Faster and Stronger: from Transformers to GPTs

Irene Li

April 24, 2023

What are Large Language Models?

GPT-3.5
GPT-4

The definition of Large Language Models?

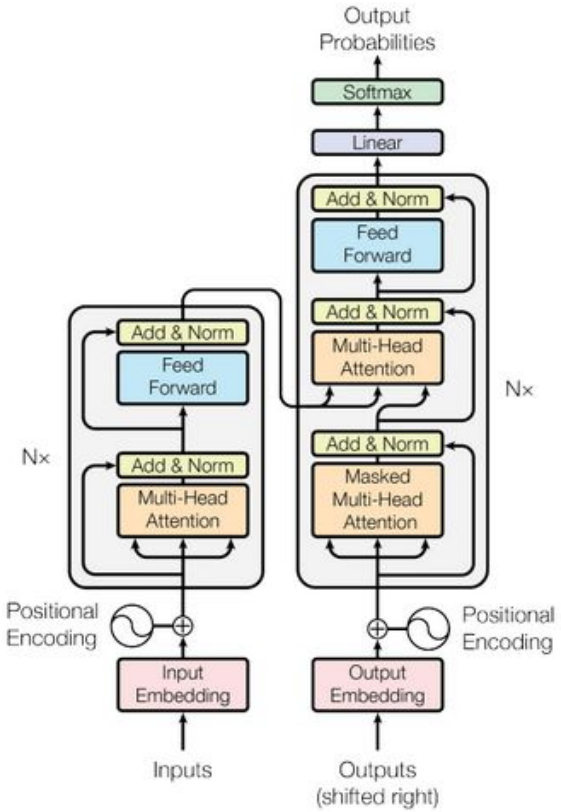
Large Language Models (LLMs) are a type of artificial intelligence (AI) models that are capable of processing and generating natural language text on a large scale. They use deep learning techniques to analyze vast amounts of text data, and learn patterns and relationships within the data to generate new text that appears to be written by a human. LLMs are often used in applications such as language translation, text summarization, and text completion. Examples of LLMs include GPT-3, BERT, and T5.

Transformers

A seq2seq model based on attention self-attention mechanism;

Foundational architecture for later strong models.

BERT, RoBERTa, XLM, …



Vaswani, Ashish, et al. "Attention is All You Need." 2017

GPT Family

Generative Pretrained Transformers:

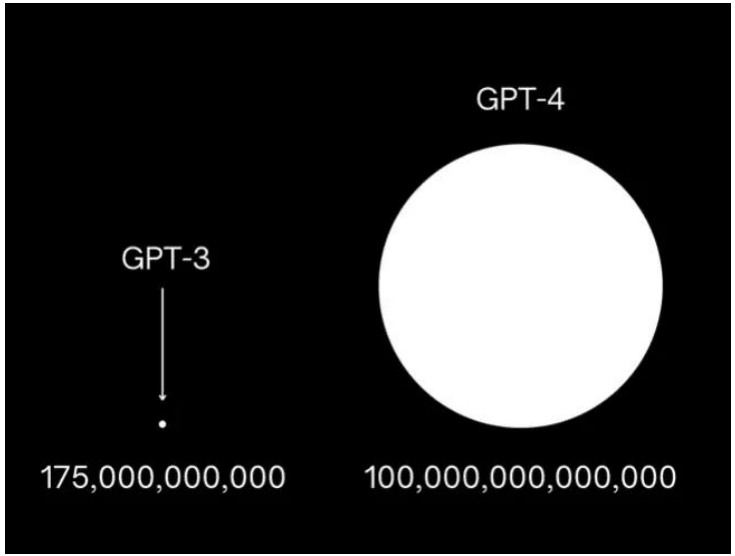
More data and more complex structure, stronger ability for longer inputs;

GPT-3: 175B parameters, 800GB to store. (100x GPT-2).

GPT-4: multi-modality, longer inputs/outputs

parameters ?

Only rumors..~ 1 trillion



https://neuroflash.com/blog/gpt-4-parameters-rumors-and-forecasts/

GPT-4: images and texts

Based on the provided ingredients, here are five food items that can be created:

1. Fruit Salad:
   Combine diced apples, oranges, and bananas in a large bowl. Optionally, you can add a squeeze of lemon juice to prevent browning and a drizzle of honey for extra sweetness. Mix well and serve chilled.
2. Cheese Omelette:
   Beat 2-3 eggs in a bowl with a pinch of salt and pepper. Heat a non-stick pan with a little oil or butter, pour in the egg mixture, and cook until it begins to set. Add grated cheese on one side, fold the omelette over, and cook until the cheese melts. Serve with your choice of bread or toast.
3. Ham and Cheese Sandwich:
   Layer slices of lunch meat and cheese on a slice of bread. Add lettuce, tomato, or other desired toppings, then spread mustard or mayonnaise on another slice of bread and place it on top. Enjoy the sandwich as is or grill it for a warm, toasty meal.
4. Fruit Smoothie:
   Blend a mix of apples, oranges, and bananas with a cup of milk (adjust the amount to achieve desired consistency) until smooth. For added flavor, add a spoonful of honey or a few ice cubes for a cold, refreshing drink.

https://twitter.com/sudu_cb/status/1636080774834257920?lang=en

THE GPT-3 FAMILY: 50+ MODELS (MAR/2023)

base (May/2020)

- GPT-3 Small 125M
- GPT-3 Large 760M
- GPT-3 2.7B
- GPT-3 13B
- GPT-3 Medium 350M *ada*
- GPT-3 XL 1.3B *babbage*
- GPT-3 6.7B *curie*
- **GPT-3 175B *davinci***

special (Dec/2021)

- WebGPT

Key
Model name / Parameters / (Instruction type)
API name
Open via API
Research only
Popular

embeddings (Jan/2022)

- ***text-embedding-ada-002***
- ➤ text similarity
- *text-similarity-ada-001*
- *text-similarity-babbage-001*
- *text-similarity-curie-001*
- *text-similarity-davinci-001*
- ➤ text search
- *text-search-ada-doc-001*
- *text-search-ada-query-001*
- *text-search-babbage-doc-001*
- *text-search-babbage-query-001*
- *text-search-curie-doc-001*
- *text-search-curie-query-001*
- *text-search-davinci-doc-001*
- *text-search-davinci-query-001*
- ➤ code search
- *code-search-ada-code-001*
- *code-search-ada-text-001*
- *code-search-babbage-code-001*
- *code-search-babbage-text-001*

instruct (old)

- *curie-instruct-beta* 6.7B
- InstructGPT-3 175B (SFT) *davinci-instruct-beta*
- *text-ada-001* 350M
- *text-babbage-001* 1.3B (FeedME)
- *text-curie-001* 6.7B (FeedME)
- *text-davinci-001* 175B (FeedME)

code (Jul/2021)

- Codex 12B *code-cushman-001*
- Codex 175B *code-davinci-001*

pretrain + cpt (Mar/2022)

- GPT-3 1.3B pretrain
- GPT-3 2.7B pretrain
- GPT-3 6.7B pretrain
- GPT-3 unsupervised cpt-text 1.2B

insert + edit (Mar/2022)

- *text-davinci-insert-001*
- *text-davinci-insert-002*
- *text-davinci-edit-001*
- *code-davinci-edit-001*

GPT-3.5 (2022-2023)

- **Codex 175B (no instruct) *code-davinci-002***
- *text-davinci-002* 175B (FeedME)
- ***text-davinci-003* 175B (PPO)**
- **ChatGPT 175B (PPO) *gpt-3.5-turbo***
- Microsoft Bing Chat 175B (Proprietary, GPT-4?)

➤ Miscellaneous API

1. cushman:2020-05-03
2. ada:2020-05-03
3. babbage:2020-05-03
4. curie:2020-05-03
5. davinci:2020-05-03

6. if-curie-v2
7. if-davinci-v2
8. if-davinci:3.0.0
9. davinci-if:3.0.0

10.davinci-instruct-beta:2.0.0 (SFT)

11.text-ada:001
12.text-babbage:001
13.text-curie:001
14.text-davinci:001

15.audio-transcribe-deprecated

16.text-chat-davinci-002-20221122
17.gpt-3.5-turbo-0301

Total models shown in this viz = **64**

Not to scale. Selected highlights only. Alan D. Thompson. March 2023. https://lifearchitect.ai/gpt-3 Sources: OpenAI 1, 2 and papers, API duplicates removed.

Our Focus…

Transformer Efficiency

Transformers Explainability

Observations On GPT-3.5

Transformer Efficiency

Diffuser: Efficient Transformers with Multi-hop Attention Diffusion for Long Sequences

The 37th AAAI Conference on Artificial Intelligence

Aosong Feng, Irene Li, Yuang Jiang, and Rex Ying

Scaling to long sequences

- Popular transformers are designed for relatively **short sequence**:
  - BERT/GPT take **512** tokens as the input sequence.
  - ViT take **197** tokens in the input sequence (with 16×16 pixels as a patch)
- Long sequence scenario:
  - The median length of a research paper is **4,133** words (data from PubMed)
  - Byte-level NLP task
  - Finer-grained image patch: 8×8 patch->**785** tokens; 4×4 patch-> **3137** tokens
- Trouble from **quadratic complexity**

BERT input length	512	1024	2048	3072	4096
GPU memory (GB)	1.9	4.5	12.5	23.6	44.2
Per-iteration time (s)	0.12	0.20	0.24	0.79	1.98

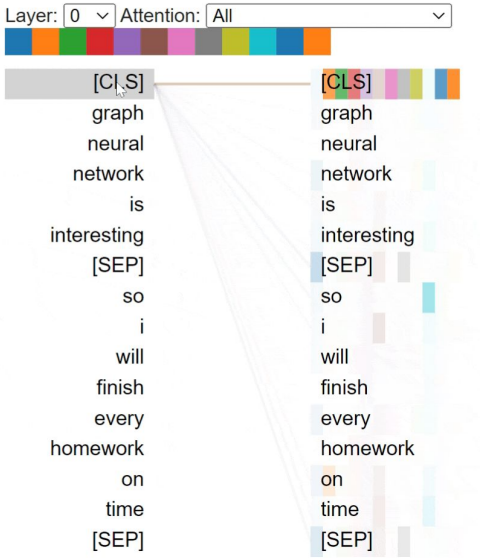
~ 20×

Full attention is not necessary

- **Observation 1**: Although every attention is calculated, most of them are close to 0, the resulting attention maps are usually **sparse**.
- **Observation 2**: non-zero attention mostly appear between the node and its **local neighbors**. (**local attention**).
- **Observation 3**: some **key words** like “so” almost attend to every token in the sentence. (**global attention**)

We can simplify the self-attention (full-attention) with appropriate sparse pattern

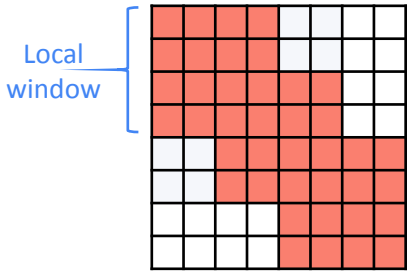
A sentence encoded by pretrained BERT



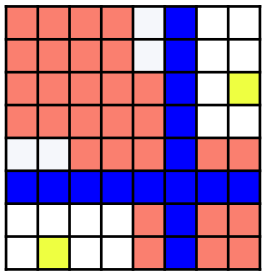
Attention between pairwise tokens

Existing Sparse Transformers

- Masking attention (with sparse pattern) can reduce complexity from $\boldsymbol{O(n^2)}$ to $\boldsymbol{O(n)}$.
- Previous sparse transformers design **attention patterns** to approximate the full attention.
- **Attention Patterns (Masks)**



Longformer [Beltagy et al., 2020]



BigBird [Zaheer et al., 2020]

No attention

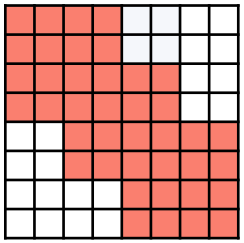
Local attention: tokens attend within a local window (size = 4 in the figure)

Global attention: one global token attend to all tokens

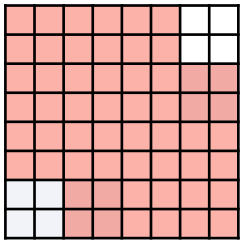
Random attention: randomly select attentions

Attention Diffusion

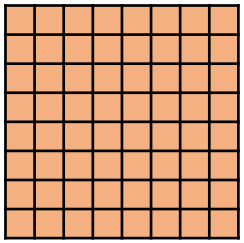
- We propose to **augment sparse transformers** with **attention diffusion** targeting the limitations (inaccurate approximation, slow propagation, bad robustness).
- Calculated attention scores between **directly connected** pairs diffuse to **indirectly connected** pairs, through multiple diffusion steps, **within one layer**.



1-step diffusion



2-step diffusion



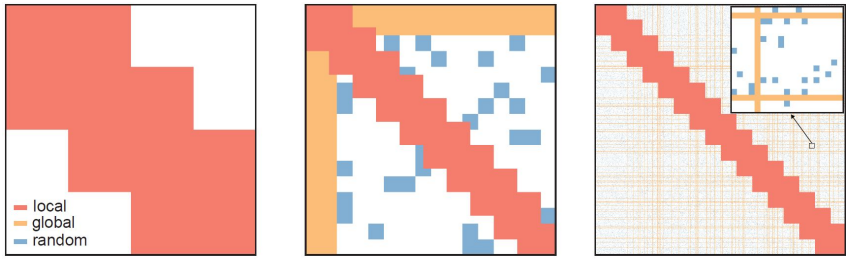
3-step diffusion

Attention Diffusion — a graph view

- Relationship to **graph diffusion**: By treating the attention matrix A ($A_{i,j}$ is the attention between token i and j) as the adjacency matrix of a graph $\boldsymbol{G}$, attention diffusion is equivalent to diffusion on $\boldsymbol{G}$

Diffuser — sparse pattern

- **Element-wise** attention mask
  - Longformer and BigBird use **block-wise** attention mask (64 tokens as a **block**), for the sake of computation



(a) Longformer (b) BigBird (c) Diffuser

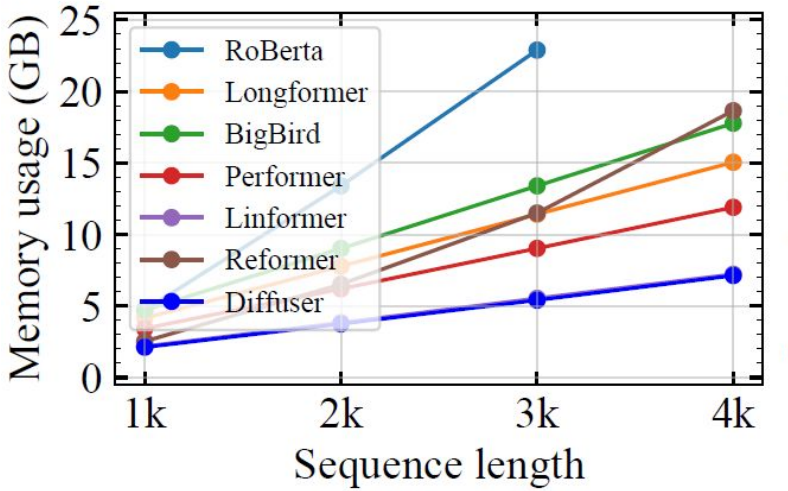
- **Sparser pattern**

Length	Longformer	BigBird	Diffuser			
			tot	loc	glob	rand
1024	62.5	55.7	**24.0**	18.0	4.2	1.9
2048	34.4	32.5	**15.5**	9.2	4.2	2.1
4096	18.0	16.9	**11.2**	4.6	4.3	2.2

> > >

Experiments — Efficiency

- Diffuser is implemented with DGL (Deep Graph Library) packages for efficient sparse attention calculations (under message passing framework).
- Diffuser achieves **1.67× memory savings** in average of input lengths, with comparable running time compared to baselines.



Experiments — Performance

- Language **finetuning** tasks

Text classification

	HYP	20NG	IMDB	A-512	A-2048	Avg.
95pt.	2,030	1,229	771	1,696	5,216	-
BERT	85.7	85.3	91.3	59.2	50.3	74.36
RoBERTa	87.4	85.7	95.3	65.0	57.9	78.26
BigBird	92.2	82.3	95.2	67.4	63.6	80.14
Longformer	93.8	86.3	**95.7**	67.3	61.2	80.86
BigBird_D	93.1	84.5	95.0	**68.2**	63.4	80.84
Longformer_D	93.5	**87.3**	95.4	67.0	62.5	81.24
Diffuser	**94.4**	86.8	95.2	67.8	**64.8**	**81.80**

Question answering

Model	WikiHop	TriviaQA	
Metric	Acc	F1	EM
RoBERTa	71.82	74.02	66.87
Longformer	75.30	74.82	67.24
BigBird	74.54	73.16	68.26
Diffuser	**75.80**	**75.84**	**70.20**

  - Diffuser outperforms baselines on different downstream tasks, with better performance on **longer sequences** as in HYP, A(Amazon)-2048.

- Diffuser shows strong performance as well in **image generative** and **Long Range Arena (LRA)** benchmarks.

Explainable Transformers: from a linguistic perspective

Large Language Models:

How can we trust them?

Scenarios in special domains.

Explainable Models:

Provide transparent and interpretable insights about the decisions

Why the model make such predictions?

What are the relationships between the output and input?

Our focus:

Explain Transformers (i.e., the attention heads)

But from a linguistic point of view.

Explainable Transformers: from a linguistic perspective

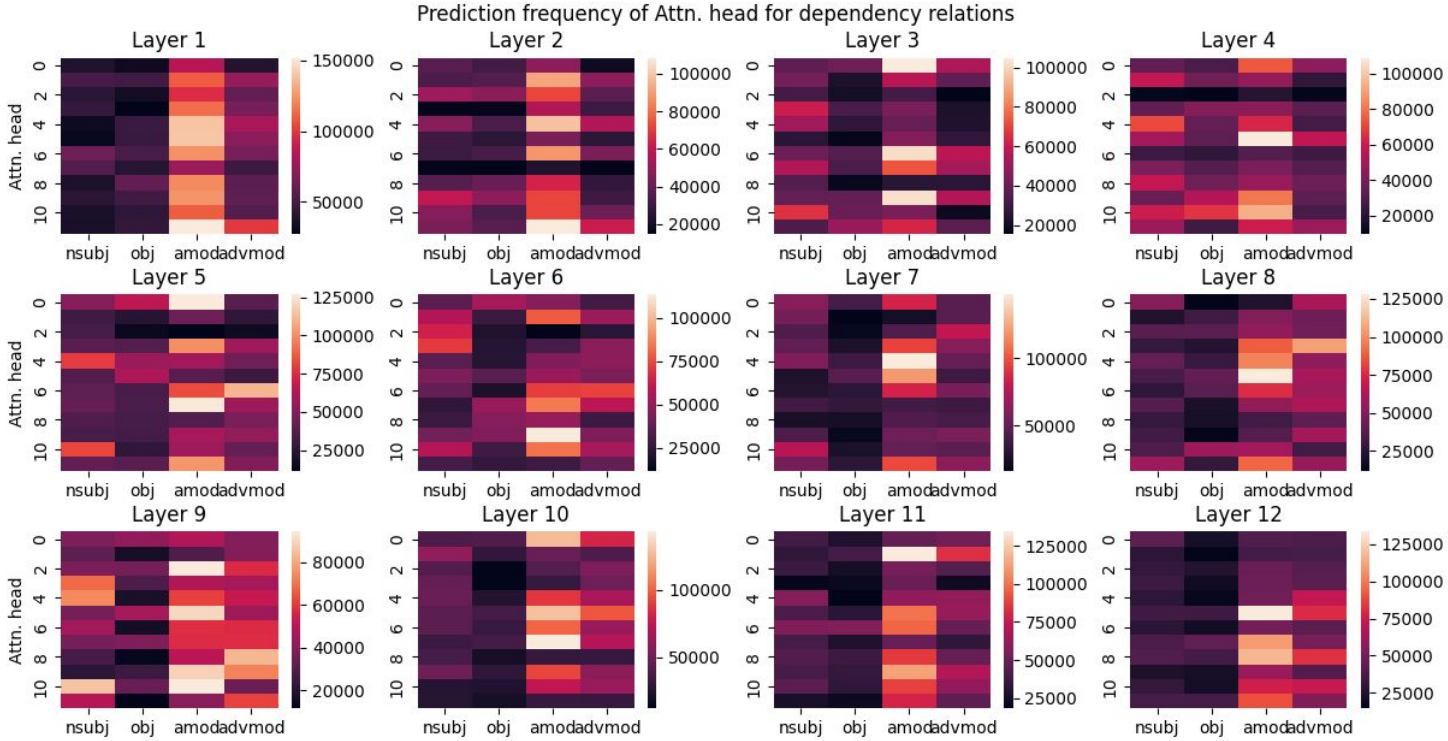
We summarized 4 core dependency relation in English from SST-2:

- **nsubj**: nominal subject
- **obj**: direct object
- **amod**: adjectival modifier
- **advmod**: adverbial modifier

She likes eating delicious sushi rolls with chopsticks.

- **nsubj:** She
- **obj:** sushi rolls
- **amod:** delicious
- **advmod:** with chopsticks

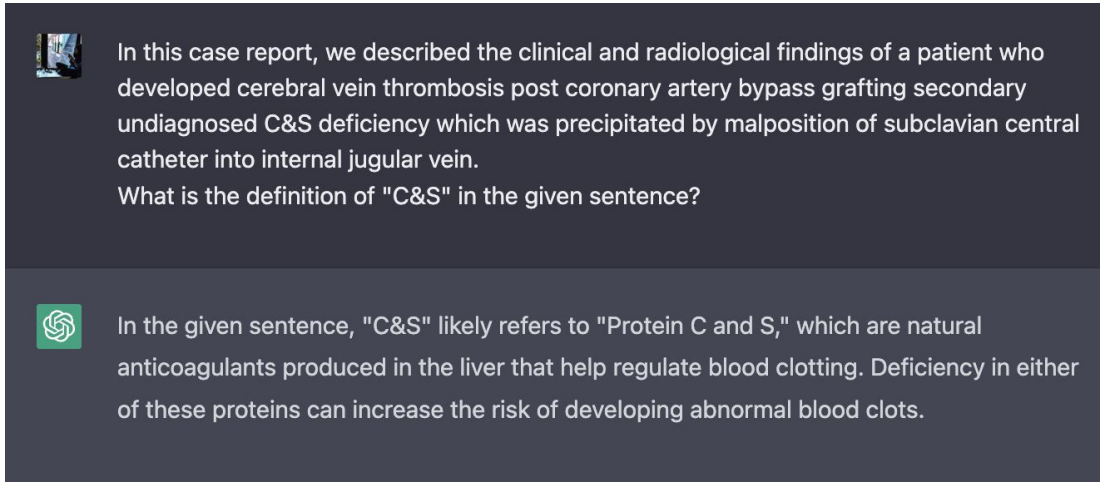
12-layer Transformers



The primary syntactic focus of each attention head in the Transformer model. A lighter color signifies a higher concentration.

Observations on GPT-3.5: zero-shot inference

Lexical Semantics through Zero-shot Definition Generation:
given a text, generate the definition of a target word or phrase.



Corresponding sub-tasks:

- Definition Modeling/Generation
- Word Sense Disambiguation
- Word in Context

Observations on GPT-3.5: zero-shot inference

Our focus: zero-shot prompting GPT-3.5/4 for DM can lead to better performance than the *generationary* approach in English.

Prompt:

> %SENTENCE%
>
> Given the example, what is the definition of %WORD% in one sentence?

Observations on GPT-3.5: zero-shot inference

WiC dataset

Sentence-level contextualised embeddings	Implementation	Accuracy %
SenseBERT-large†	Levine et al (2019)	72.1
KnowBERT-W+W†	Peters et al (2019)	70.9
RoBERTa	Liu et al (2019)	69.9
BERT-large	Wang et al (2019)	69.6
Ensemble	Gari Soler et al (2019)	66.7
ELMo-weighted	Ansell et al (2019)	61.2

GPT-3.5 Zero-shot with prompt: 70.5

Next…

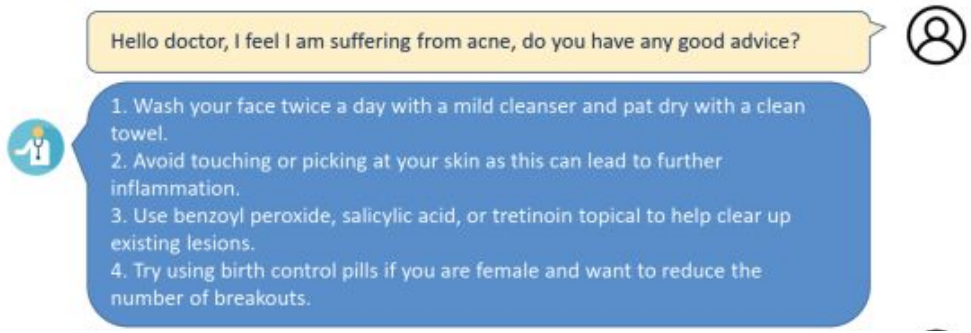
Transformer Efficiency

Transformers Explainability

Observations On GPT-3.5

(LLM) ChatGPT-like models for medical applications:

- Utilizing specific medical domain knowledge
- Possibility to be open source



ChatDoctor