Recipe for the model behind ChatGPT

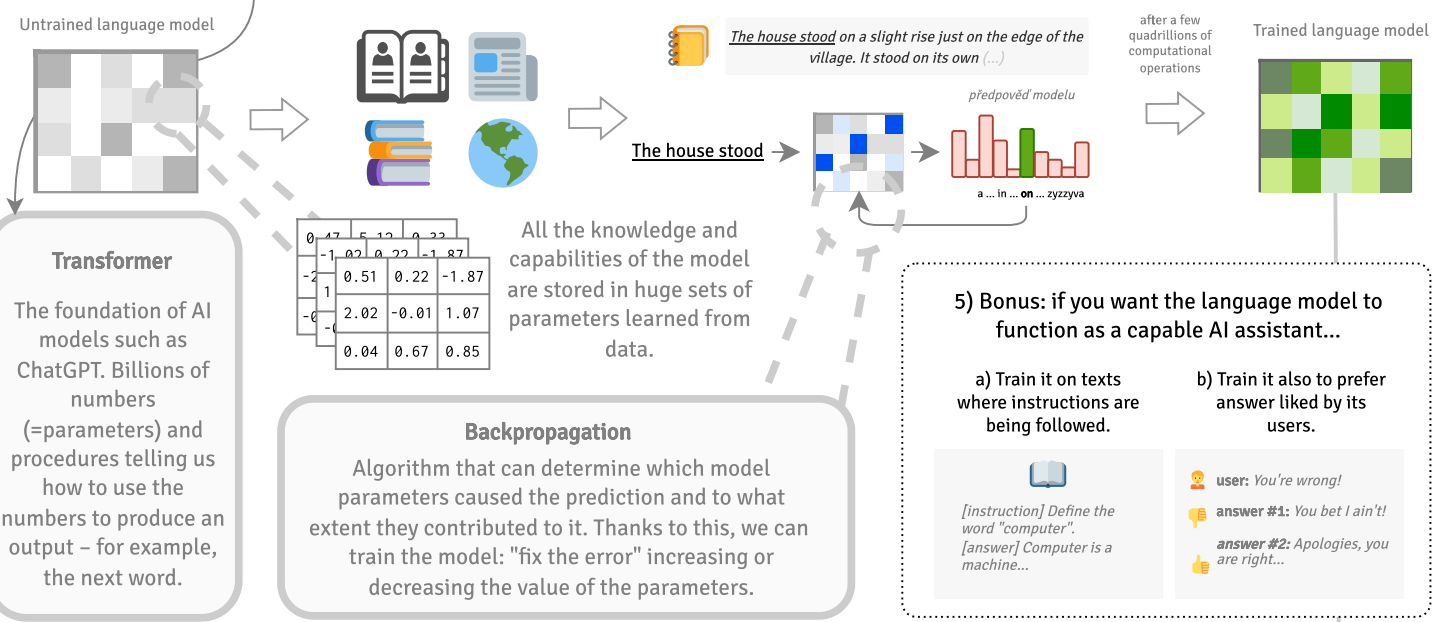
Univerzita Karlova · ÚFAL

1) Take a *Transformer* and set its parameters randomly. (If you want a very capable model, take a very large *Transformer*.)

2) Collect lots of texts: internet, documents, books. Having more texts of higher quality leads to better outputs.

3) For each text prefix, ask the model which word should follow. When the model doesn't guess the word correctly, slightly adjust the parameters so that it predicts this word for the same prefix with higher probability.

4) Done! Our model can decently predict the next word in a sentence on the texts it was trained on.



Transformer

The foundation of AI models such as ChatGPT. Billions of numbers (=parameters) and procedures telling us how to use the numbers to produce an output – for example, the next word.

All the knowledge and capabilities of the model are stored in huge sets of parameters learned from data.

Backpropagation

Algorithm that can determine which model parameters caused the prediction and to what extent they contributed to it. Thanks to this, we can train the model: "fix the error" increasing or decreasing the value of the parameters.

5) Bonus: if you want the language model to function as a capable AI assistant...

a) Train it on texts where instructions are being followed.

[instruction] Define the word "computer". [answer] Computer is a machine...

b) Train it also to prefer answer liked by its users.

user: *You're wrong!*

answer #1: *You bet I ain't!*

answer #2: *Apologies, you are right...*

Let's generate some text!

Generating the next word in a sentence

Embedding

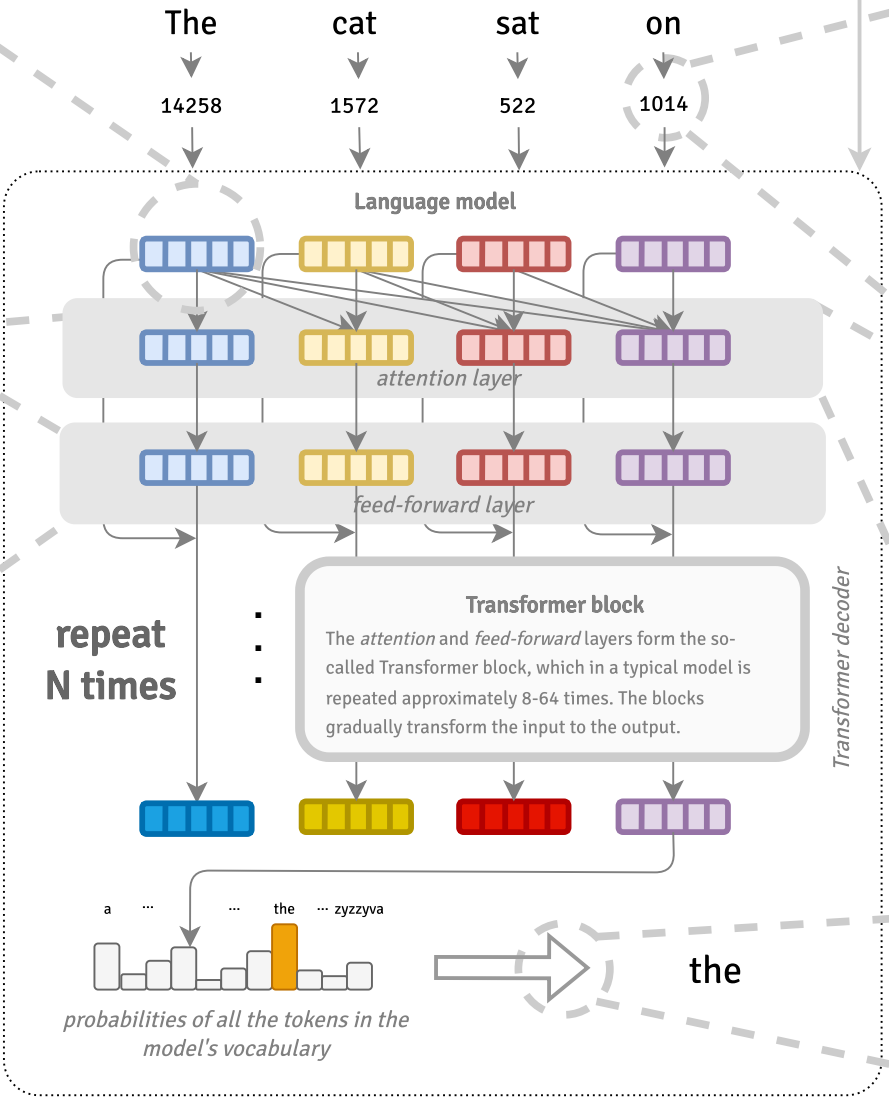
Several hundred or thousands of numbers that represent a token (= a word or group of characters).

Feed-forward layer

A layer where the model processes the representation of each token independently.

Computational operations

The arrows symbolize mathematical operations. Within Transformer blocks, these are primarily multiplication and addition. These operations combine the input and the model parameters.



Token id

```
...
1013: let
1014: on
1015: $$$
...
```

The vocabulary of a typical model contains several tens of thousands of tokens. Each token has its unique ID.

Attention layer

A layer that takes information from other tokens and mixes it into the token's representation.

Decoding algorithm

Decides which token to generate based on the probabilities assigned to individual tokens in the model's vocabulary.
What do you think: do we always want to generate the token with the highest probability? 😉

NEW: See it all animated at `https://animatedllm.github.io`!

Frequently asked questions about language models

Can language models also look up answers on the internet?

Short answer: They can't! (But someone can do it for them.)

Long answer: A language model as such cannot search the internet. It had to learn everything it can do during training.

Nowadays, however, commercial language models typically have an external search module. This module searches for information related to the query and appends it to the user input. The model then works with this information the same way as with the rest of the text. The quality of the search results can influence the generated text both positively and negatively.

Does the language model have a list of all words in all languages?

Short answer: No – but of their parts, yes!

Long answer: There are indeed too many words in all the languages combined (think of all the Czech declinations!).

Instead, we use so-called "subwords": words and their parts from which we can assemble the rest by "gluing" the parts together. We have the more frequent words in the vocabulary directly and we assemble the less common ones from multiple parts. Breaking text into subwords and then re-assembling it is the task of a specialized algorithm, the so-called tokenizer, which works independently of the language model.

Why don't models know other languages as well as English?

Short answer: Because English is the language of the internet.

Long answer: We still can't teach models language nearly as efficiently as, for example, children. Not even all of Wikipedia is enough for a model to learn the language perfectly. The model's capabilities therefore grow with the amount of texts it was trained on. And there is really a *lot* of English texts online!

Surprisingly, there is quite a lot texts on the internet even in Czech. But in Irish or Telugu, for example, the model will lag behind.

How energy-intensive is it to train a model? And how much energy does one query consume?

Short answer: Training is demanding, generating less so.

Long answer: It depends on the size of the model, but it's estimated that *training* a model with 175 billion parameters costs 1.2 GWh of energy, which corresponds to the annual consumption of 120 American households. Fortunately, models are not trained that often. Generating text from a trained model is much more efficient: an average ChatGPT response consumes approximately 0.3 Wh, which corresponds to, for example, an electric kettle turned on for 1 second.

[1] https://blog.samaltman.com/the-gentle-singularity
[2] https://adasci.org/how-much-energy-do-llms-consume-unveiling-the-power-behind-ai/
[3] https://andymasley.substack.com/p/a-cheat-sheet-for-conversations-about

What makes large language models so intelligent?

Short answer: Such a tricky question...!

Long answer: Models can to some extent repeat what they've seen in the training data. Their smart answers might be therefore just "parroted" from someone on the internet. But to some extent, the models can also combine learned patterns in an original answer. It is because with a large amount of data, it might be simpler to learn to generalize than to memorize everything.

And here come further questions: What makes a human intelligent? And what actually is this "intelligence"?

Where can I learn more about language models?

Short answer: Here at ÚFAL!

Long answer: We deal with similar topics at the Institute of Formal and Applied Linguistics (ÚFAL), which is part of the Faculty of Mathematics and Physics at Charles University. At our department, we connect knowledge about language with knowledge from machine learning.

We teach bachelor's and master's courses about it, such as Deep Learning or Large Language Models.

vytvořil Zdeněk Kasner, MFF UK

https://kasnerz.github.io/assets/materials/llm_cheatsheet.pdf