

Text generation with Large Language Models (1/2)

with focus on ChatGPT

October 13, 2023

Generative AI 2023: a short but thorough and hands-on introduction
Bo Kang, Thomas Demeester, Tijl De Bie

TH

Write a tweet about my upcoming lecture on ChatGPT in the UGent Generative AI course.



Excited to be a part of the UGent Generative AI course! 🚀 Join me for an upcoming lecture on ChatGPT, where we'll delve into the world of generative AI and explore how ChatGPT is reshaping human-computer interactions. 🤖🌟 Don't miss this opportunity to learn about cutting-edge AI! #UGentGenerativeAI #ChatGPT #AIEducation



<https://chat.openai.com/>

ChatGPT (GPT-3.5), 2023-10-13

⚡ GPT-3.5

🌐 GPT-4

Our most capable model, great for tasks that require creativity and advanced reasoning.

Available exclusively to Plus users

GPT-4 currently has a cap of 25 messages every 3 hours.

⊕ Default

🌐 Browse with  Bing Beta



⊕ Plugins Beta

🖼 DALL-E 3 Beta

TH

This afternoon, Thomas Demeester will do a lecture on LLMs, for the Generative AI course at UGent. Write a tweet as announcement.



Finished browsing

I was unable to find specific information regarding Thomas Demeester's lecture on LLMs for the Generative AI course at UGent this afternoon. However, based on the information available, I can help you draft a tweet announcement for the lecture:



Excited for this afternoon's lecture on Large Language Models (LLMs) by prof. dr. ir. Thomas Demeester at #UGent! Dive into the fascinating world of #GenerativeAI with us in the Generative AI course. 🎉🧠 Don't miss it!
#AI #MachineLearning #LanguageModels

ChatGPT (GPT-4 with Bing),
2023-10-13

TH

Write a tweet about the upcoming guest lecture of Thomas Demeester at UGent, for the Generative AI course, on recent big evolutions in his research field.

hinting at LLMs



Finished browsing

Based on the recent evolutions in Thomas Demeester's research, it's evident that he has been actively engaged in multiple projects spanning from biomedical NLP to educational applications. Here's a draft tweet regarding his upcoming guest lecture at UGent for the Generative AI course:

"Excited for Prof. Thomas Demeester's guest lecture at #UGent's Generative AI course! Dive deep into the realms of #NLP with his insights on large-scale biomedical term glossaries, adaptive learning through partially annotated data, and novel methods in distractor generation for educational exercises. Don't miss the chance to learn from his recent groundbreaking evolutions in the field! 📚💻 #GenerativeAI #AIResearch #EducationalAI"

nicely written, but
based on just some
randomly chosen papers

ChatGPT (GPT-4 with Bing),
2023-10-13

⚡ GPT-3.5

++ GPT-4

Our most capable model, great for tasks that require creativity and advanced reasoning.

Home > Software Apps

ChatGPT pauses Bing integration to stop people from bypassing paywalls

By [Kevin Okemwa](#) published 7 days ago

ChatGPT aided users aiming to bypass paywalls, but OpenAI put a stop to it.



Comments (0)

early July 2023

Choose Microsoft Edge
for AI-powered search



What's new in Microsoft Edge

Microsoft Edge introduces exciting new features every month. Check out the latest features here.

[See new features](#)

Enhance your browsing
capabilities with AI

Get quick access to AI-powered tools, apps, and more right within Microsoft Edge's sidebar. This includes Bing Chat where you can ask questions, get answers, refine search, summarize, and create content—all without switching tabs or breaking your flow.

[Learn more](#)

Microsoft Edge

write a tweet to announce the first lecture of the UGent Generative AI Course 2023, on LLMs.



Your personal and company data are protected in this chat

The UGent Generative AI Course 2023 is starting soon! Don't miss the first lecture on LLMs, the state-of-the-art models for natural language generation. Learn how LLMs work, what they can do, and how to use them in your own projects. The lecture will be on **Monday, October 16, 2023 at 10:00 AM in Auditorium A1**. See you there! #UGent #GenerativeAI #LLMs

pure hallucination

```
import os
import openai
```

```
openai.organization = "*****"
openai.api_key = os.getenv("OPENAI_API_KEY")
```

```
prompt = "Write a tweet about the upcoming guest lecture on ChatGPT by Thomas Demeester for the course 'Generative AI'.
```

```
def readout_chatgpt(pr, T=0):
    resp = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[{"role": "user", "content": pr}],
        temperature=T, max_tokens=300
    )
    return resp[ "choices" ][0][ "message" ][ "content" ]
```

```
print ('\n'.join([f'T={T}:\n{readout_chatgpt(prompt,T)}\n' for T in [0.0, 0.3, 1.0]]))
```

T=0.0:

Excited for the upcoming guest lecture on ChatGPT by Thomas Demeester for our 'Generative AI' course! 😊 Can't wait to dive into the world of conversational AI and learn from the best. Join us for an insightful session on ChatGPT's capabilities and its impact on the future of AI. Don't miss out! #GenerativeAI #ChatGPT #GuestLecture

T=0.3:

Excited for the upcoming guest lecture on ChatGPT by Thomas Demeester for our 'Generative AI' course! Can't wait to dive deeper into the world of AI and explore the potential of ChatGPT. Join us for an insightful session and get ready to be amazed! #GenerativeAI #ChatGPT #AIinEducation

T=1.0:

"🌟 Excited to announce the upcoming guest lecture on ChatGPT by Thomas Demeester for our 'Generative AI' course! 🎉 Get ready to dive into the world of cutting-edge language models and their applications. Don't miss out on this incredible opportunity to gain insights from an expert in the field! #AI #GenerativeAI #ChatGPT"

Write a short outline for a one-hour lecture on ChatGPT for the hands-on UGent course on generative AI. Use markdown.



Lecture Outline: ChatGPT - Hands-on with Generative AI

I. Introduction to ChatGPT

- **Time:** 5 minutes
 - Welcome and course overview
 - Introduction to ChatGPT and its role in generative AI
 - Importance of generative AI in various applications

II. Understanding ChatGPT's Architecture

- **Time:** 15 minutes
 - Deep dive into the architecture of ChatGPT
 - Explanation of neural networks and transformer models
 - Discuss the pre-training and fine-tuning process

III. Building a ChatGPT Environment

- **Time:** 10 minutes
 - Setting up the development environment
 - Installation and configuration of necessary libraries and tools
 - Accessing ChatGPT API (if available)

IV. Fine-tuning ChatGPT

- **Time:** 10 minutes
 - Explanation of fine-tuning and its importance
 - Demonstrating how to fine-tune ChatGPT for specific tasks or domains
 - Discussing data requirements and best practices

good luck
with that!

V. ChatGPT Applications

- **Time:** 10 minutes
 - Real-world use cases of ChatGPT
 - Examples of applications in customer support, content generation, and more
 - Showcasing how ChatGPT can solve specific problems

VI. Hands-On Workshop

- **Time:** 20 minutes
 - Guided practical session:
 - Participants experiment with pre-trained ChatGPT models
 - Fine-tune ChatGPT for a sample task or dataset
 - Create chatbots or generate text using ChatGPT

VII. Challenges and Ethical Considerations

- **Time:** 10 minutes
 - Discussing challenges in using generative AI like ChatGPT
 - Addressing ethical concerns, biases, and responsible AI usage
 - Encouraging responsible AI development and deployment

Introduction to language models

ChatGPT

Prompting strategies

Q&A

A short history of neural language models

Why do we want to know whether a sentence is “likely” or not?

E.g. ambiguity in speech recognition:

$$P(\text{Why do dogs wag } \underline{\text{there}} \text{ tales}) < P(\text{Why do dogs wag } \underline{\text{their}} \text{ tails?})$$

Why difficult?

“Curse of dimensionality”: new sentence likely different from any sentence ever seen!

How to reach **generalization**?

- Classical n-gram based models:
generalization by concatenating very short overlapping sequences seen during training.
- Neural language models with **learned distributed word representations**:
generalization from fact that new sequences are expressed with similar representations as sequences seen during training.

A short history of neural language models

Core idea: the model jointly learns

- (1) a distributed representation for each word, and
- (2) the probability function for word sequences

total context
up to 32k tokens
in GPT-4

$$P(y^{<1>} , y^{<2>} , \dots , y^{<T_y>}) = \prod_{t=1}^{T_y} P(y^{<t>} | y^{<1>} , \dots , y^{<t-1>})$$

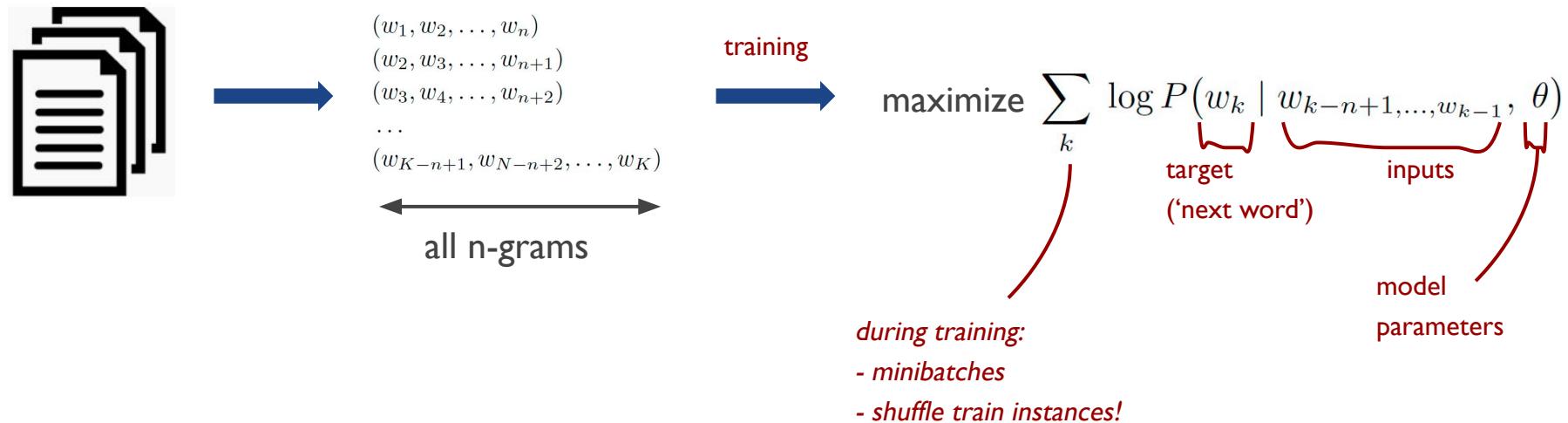
joint prob. of sequence

$$\approx \prod_{t=1}^{T_y} P(y^{<t>} | y^{<t-n+1>} , \dots , y^{<t-1>})$$

only n-1 last words
(as in n-gram model)

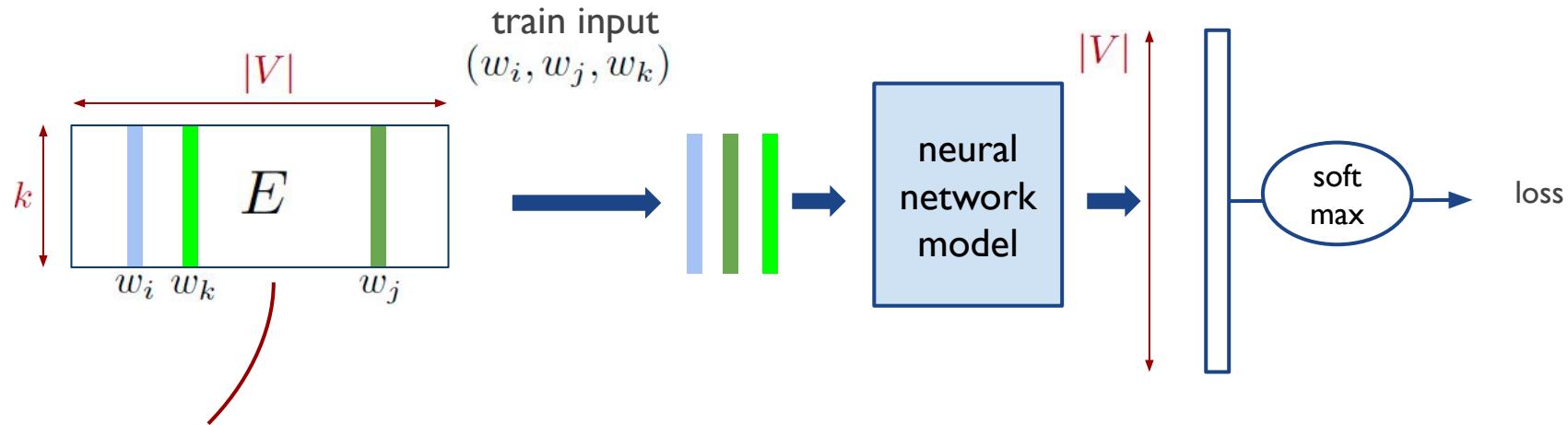
A short history of neural language models

Find a model that maximizes the probability of training sequences



A short history of neural language models

Model architecture?



*matrix with distributed word
representations (efficient look-up)
randomly initialized
(to be trained jointly
with the model!)*

Bengio et al., "A neural probabilistic language model",
NIPS 2000

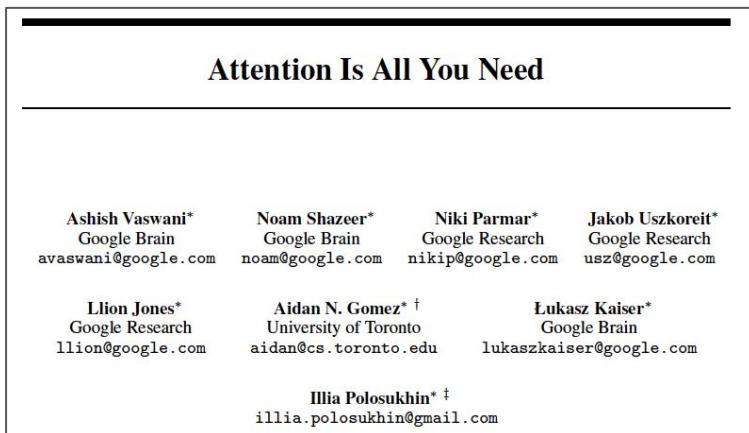
Merity et al., "Regularizing and Optimizing LSTM Language Models",
ICLR 2018

A short history of neural language models

Model architecture?

Many years: **Recurrent Neural Networks** (such as LSTM)

2017: introduction of the **Transformer** by Google



> 90k citations today...

2019: introduction of **BERT** (again Google)



> 78k citations today...

Transformer-based language models

3 broad classes:

Auto-regressive models

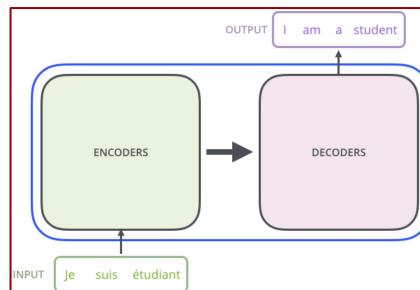
*like the original LM (Bengio):
(predict the next word given the past)*

Auto-encoding models

*like BERT: masked language model
(mask some words, and learn to predict them from bidirectional context)*

Sequence-to-sequence models

*Encoder-decoder models
(like the 'original transformer')*



Transformer-based language models

Auto-regressive models

Training objective (“causal LM objective”):

for a given sequence $\mathbf{x} = [x^{<1>} , x^{<2>} , \dots , x^{<T>}]$ find parameters θ
to maximize the log-likelihood

$$\log p_\theta(\mathbf{x}) = \sum_{t=1}^T \log p_\theta(x^{<t>} | \mathbf{x}^{<1:t-1>})$$

joint prob. written as forward autoregressive factorization

embedding of token t in the vocab.
(row t of linear output layer from hidden to vocab space)

neural representation of the previous part of the seq.
(e.g., hidden RNN state,
or transformer output with correct att. masking)

$$= \sum_{t=1}^T \log \frac{\exp(h_\theta(\mathbf{x}^{<1:t-1>})^T e(x^{<t>}))}{\sum_{x'} \exp(h_\theta(\mathbf{x}^{<1:t-1>})^T e(x'^{<t>}))}$$

denominator of softmax over vocab space

Can be used as pre-training strategy in transformers, e.g. for GPT models.

Similar to the decoder in the original Transformer.

Transformer-based language models

Auto-encoding models

BERT: based on ‘denoising auto-encoding’ objective:

First create a corrupted version $\hat{\mathbf{x}}$ of \mathbf{x} (e.g., replace 15% of tokens in \mathbf{x} by [MASK]).

Say $\bar{\mathbf{x}}$ represents the masked tokens,

and the model is a transformer that maps a sequence \mathbf{x}

into a sequence of representations $H_\theta(\mathbf{x}) = [H_\theta(x)^{<1>}, H_\theta(x)^{<2>}, \dots, H_\theta(x)^{<T>}]$

then the training objective is to reconstruct $\bar{\mathbf{x}}$ from $\hat{\mathbf{x}}$ by maximizing

$$\log p_\theta(\bar{\mathbf{x}}|\hat{\mathbf{x}}) \approx \sum_{t=1}^T m_t \log p_\theta(x_t|\hat{\mathbf{x}})$$
$$\approx \sum_{t=1}^T m_t \log \frac{\exp(H_\theta(\hat{\mathbf{x}})^{<t>} e(x^{<t>}))}{\sum_{x'} \exp(H_\theta(\hat{\mathbf{x}})^{<t>} e(x'^{<t>}))}$$

*Approximation!
(assumes indep.
of masked tokens)*

$m_t = 1$ indicates token $x^{<t>}$ is masked

Transformer-based language models

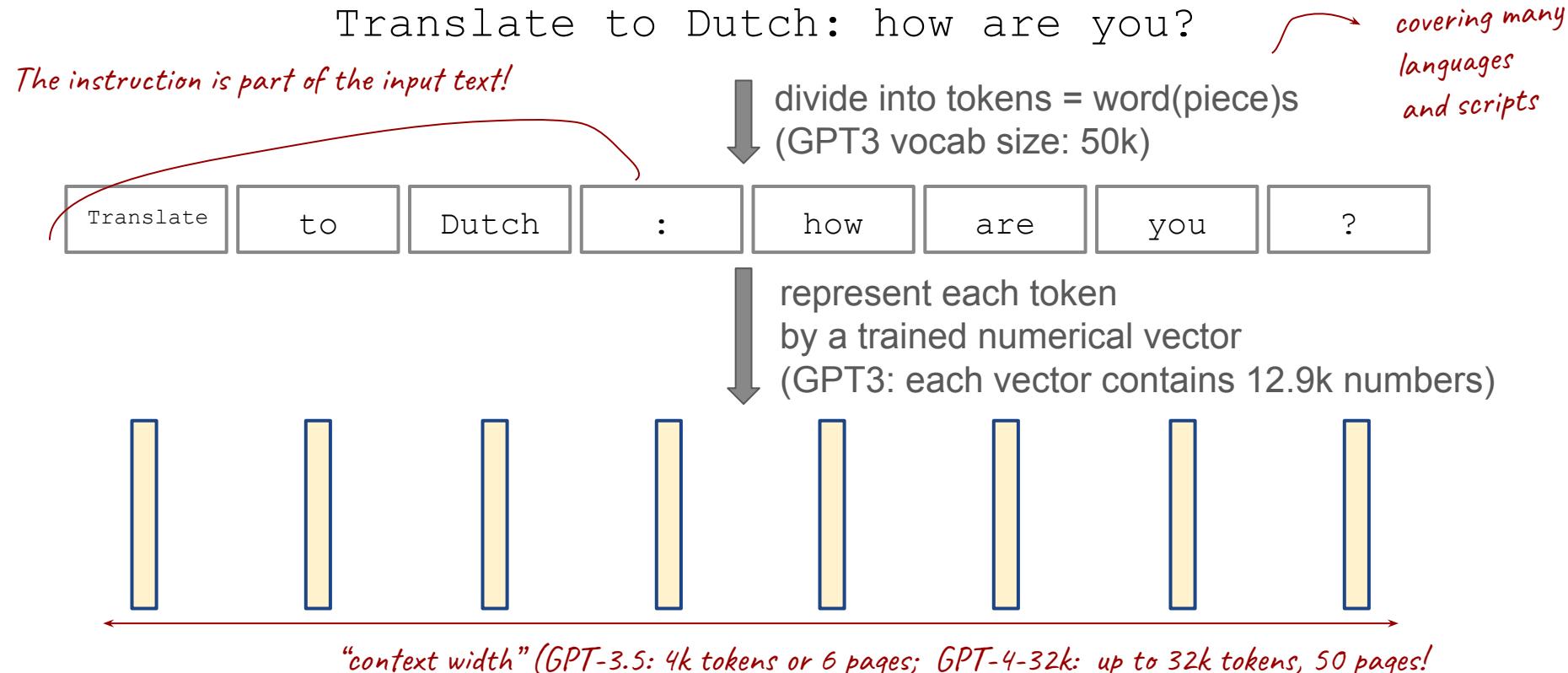
Derived models

Hundreds of adapted architectures in only a few years time

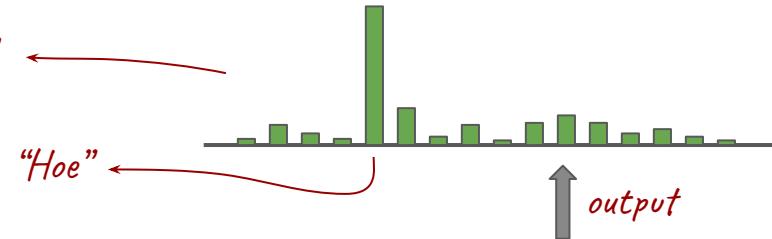
- different fine-tuning *objectives*
(e.g., language-agnostic representations; instruction-finetuning...)
- new fine-tuning *strategies* (e.g., LoRa finetuning)
- ever larger pre-trained models (autoregressive models are taking over)
- multi-modal models are being introduced
- new application domains (programming, protein LMs...)

Transformer-based language models

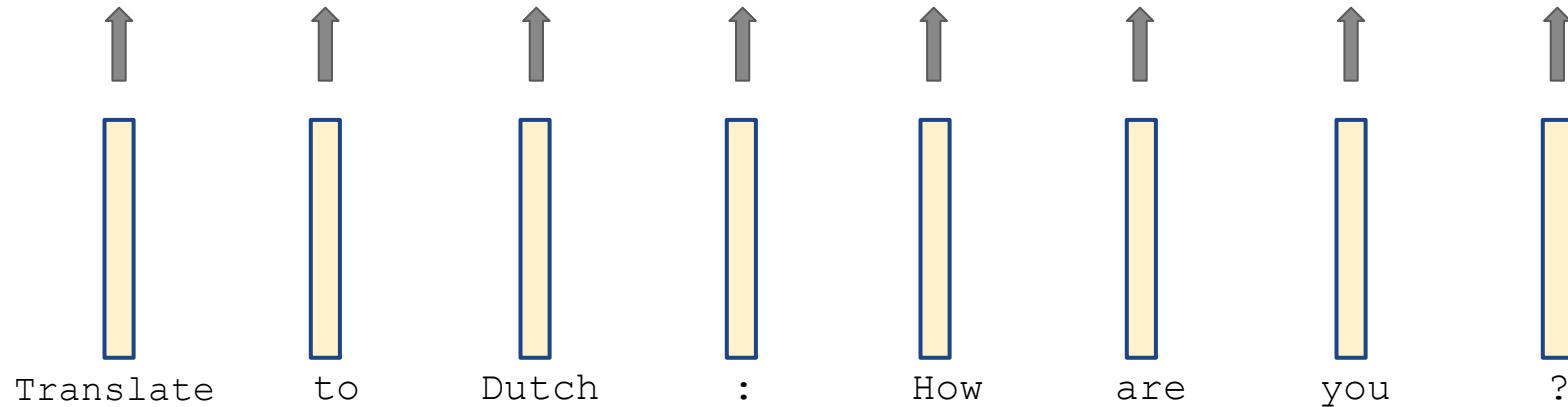
Inference in auto-regressive models



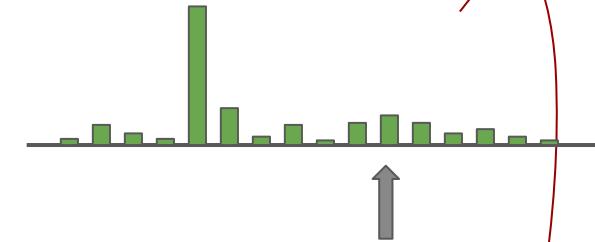
probability distribution over the vocabulary for 'the next word'



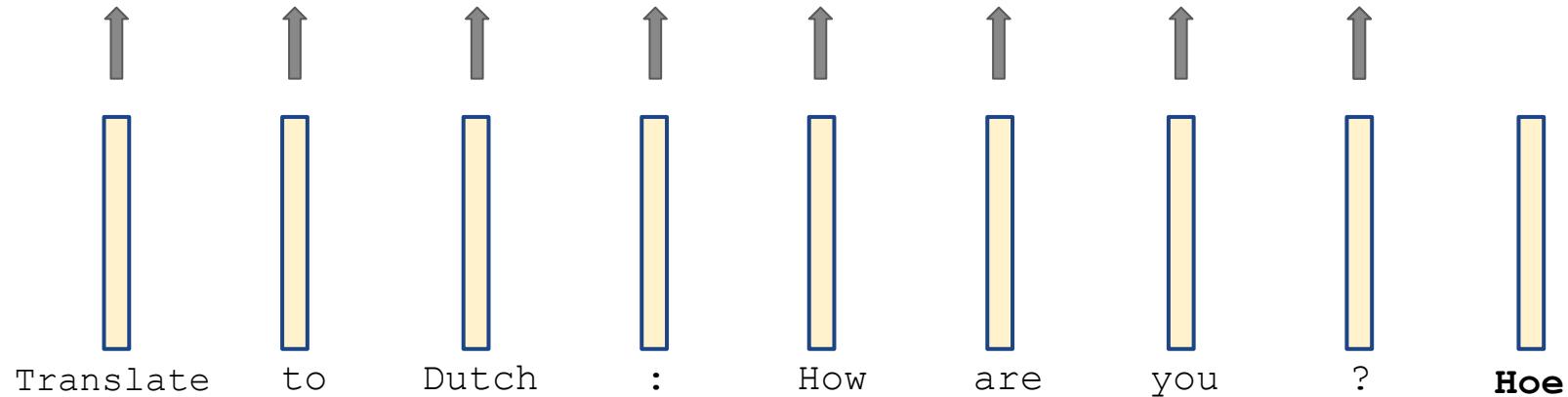
Transformer based autoregressive language model

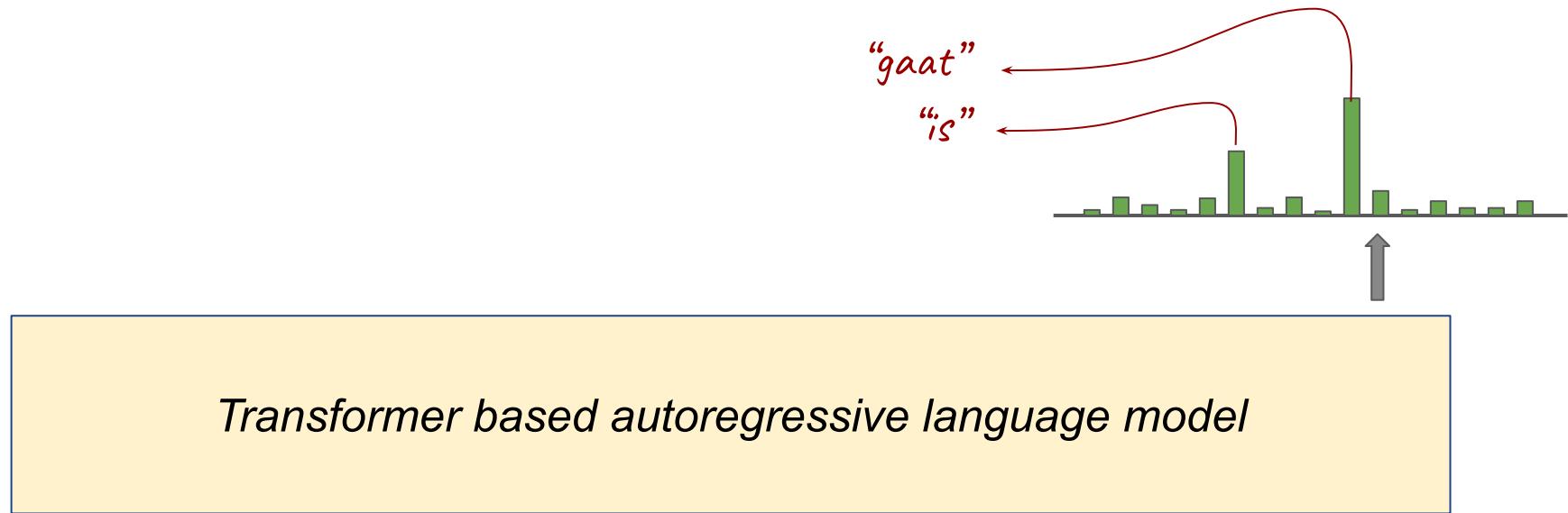


Sample the next word from this distribution



Transformer based autoregressive language model

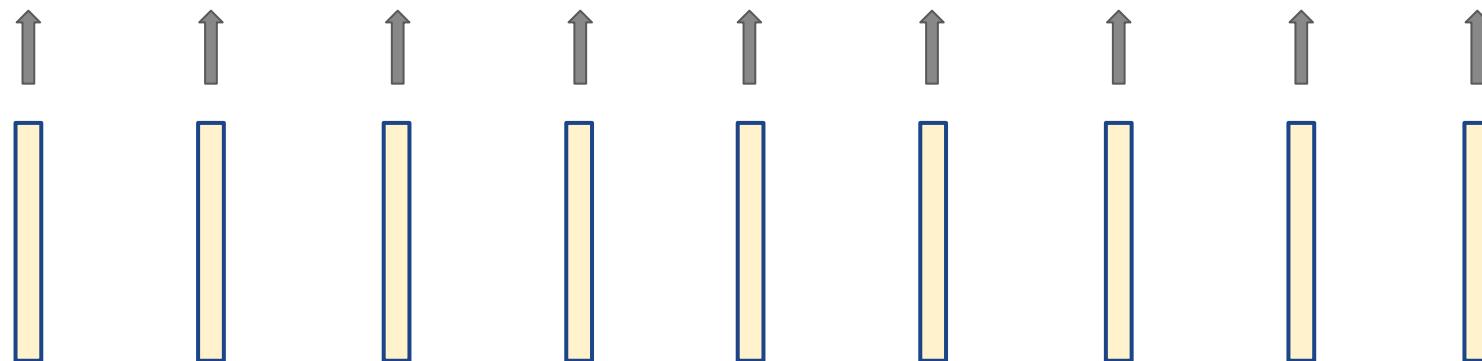




↑
↑
↑
↑
↑
↑
↑
↑
↑
↑

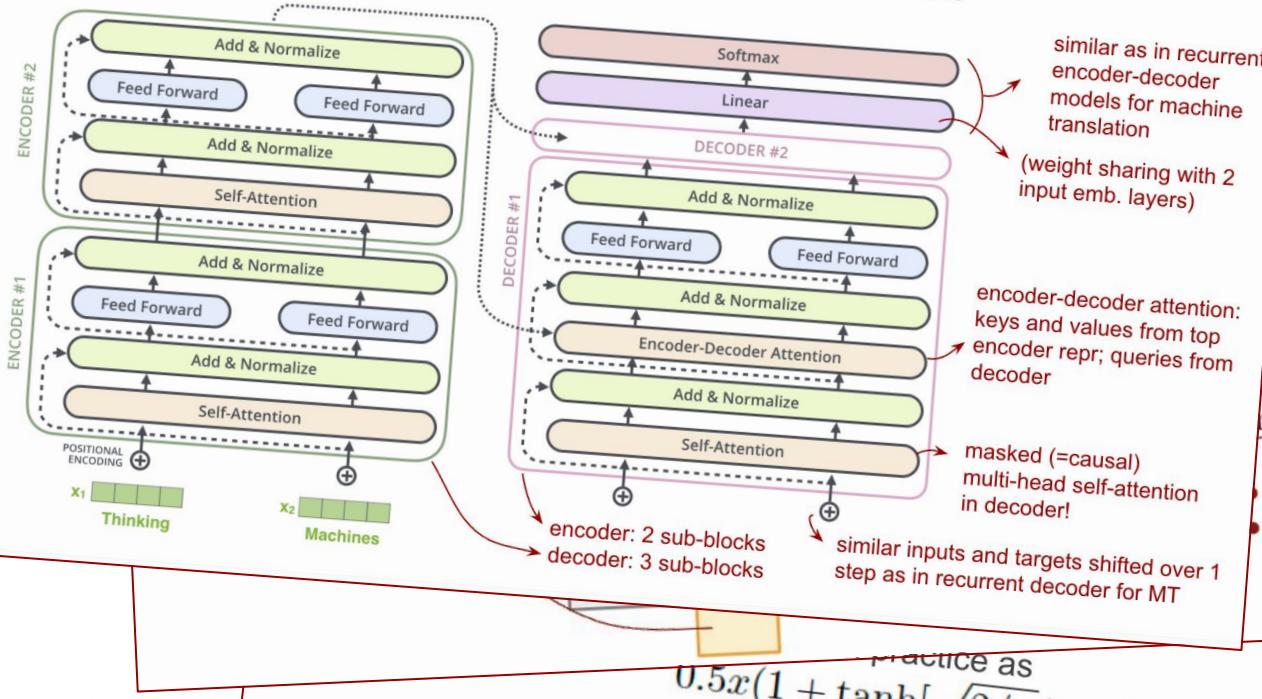
Translate to Dutch : How are you ? **Hoe**

- *To generate one token, the entire input flows through the model.*
- *Efficient to parallelize (also during training)!*
 - *local computations*
 - *each input compared many times to all others in efficient ‘self-attention’ steps*
- *Very expressive model (very ‘deep’!)
GPT3: 175B trainable parameters!*



Encoder-decoder setup

Qualitative structure overview, with 2 encoder and decoder blocks



tion vectors?

er

ion (ZW^Q, ZW^K, ZW^V)



general:

tion (QW^Q, KW^K, VW^V)

self-att: $Q = K = V = Z$

enc.-dec.-att:

$\begin{cases} Q \text{ from decoder (prev. layer),} \\ K, V \text{ from encoder output} \end{cases}$

$$2 \left[1 + \operatorname{erf}(x/\sqrt{2}) \right]$$

$$0.5x(1 + \tanh[\sqrt{2/\pi}(x + 0.044715x^3)])$$

turns out to
other popul...

The GPT family

“Generative Pretrained Transformers” by OpenAI

Key idea: pretraining of *Transformer* based *autoregressive* language model



highly parallelizable and potentially very deep

learn to predict ‘the next word’

[GPT-1] Radford et al., “Improving Language Understanding by Generative Pre-Training”, 2018,
https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

The GPT family

Key idea: pretraining of *Transformer* based *autoregressive* language model

GPT-1

12 transformer blocks, 40k vocab size, emb. size 768;
context window 512 tokens; 117M parameters.

Key insight:

next-word prediction: effective pre-training objective;
performs well on various tasks with little fine-tuning

[GPT-1] Radford et al., "Improving Language Understanding by Generative Pre-Training", 2018,
https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

The GPT family

Key idea: pretraining of *Transformer* based autoregressive language model

GPT-3

96 transformer blocks, 50k vocab size, emb. size **12,888**; context window **2,048** tokens; **175B** parameters.

Key insight:

generates qualitative natural language.

Clear improvements in many NLP tasks; few-shot learning.

Attention to potential misuse of human-like generation capabilities; potential biases.

[GPT-3] Brown et al, “**Language models are few-shot learners**”, NeurIPS 2020,
<https://papers.nips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf>

```
for _ in range(20):
    response = openai.Completion.create(
        engine="text-davinci-003",
        prompt="He was a doctor, she was",
        max_tokens=2
    )
    print(response.choices[0].text.strip())

a nurse
a nurse
a nurse
his wife
a nurse
a nurse
a lawyer
a nurse
a nurse
a nurse
a writer
a nurse
a nurse
a nurse
a nurse
a nurse
a lawyer
a nurse
a nurse
```

The GPT family

Key idea: pretraining of *Transformer* based *autoregressive language model*

GPT-4

This report focuses on the capabilities, limitations, and safety properties of GPT-4. GPT-4 is a Transformer-style model [39] pre-trained to predict the next token in a document, using both publicly available data (such as internet data) and data licensed from third-party providers. The model was then fine-tuned using Reinforcement Learning from Human Feedback (RLHF) [40]. Given both the competitive landscape and the safety implications of large-scale models like GPT-4, this report contains no further details about the architecture (including model size), hardware, training compute, dataset construction, training method, or similar.

Important:
contexts up to 32k tokens
(=25,000 words!)

Multi-modal:
text + image → text

in the mean time:

InstructGPT, GPT-3.5, ChatGPT... (several variants, also in different sizes)

Trained on (instruction, ideal answer) data,
further trained to mimic human preferences

The GPT family

Key idea: pretraining of *Transformer* based *autoregressive* language model

GPT-4

- in the paying version of ChatGPT (*also in Bing chat, but restricted*)
- *slow, but better*
 - more complex prompts
 - problems that require reasoning
 - complex coding problems
 - advanced text manipulation

The GPT family

InstructGPT

Unwanted behavior of GPT-3, that should be averted:

- hallucination,
- generation of biased / toxic text,
- not following user instructions

How to teach language models to follow instructions and align with human expectations?

Starting point:



Training language models to follow instructions

InstructGPT

- (1) Labelers provide demonstrations of ‘ideal’ (prompt, output) pairs; fine-tune GPT-3.
- (2) Labelers indicate preferences between model outputs;
train a reward model to score (prompt, output) pairs
to predict human-preferred output

reward model: based on fine-tuned GPT-3
- (3) Further fine-tune the model with reinforcement learning,
based on random customer prompts and predicted rewards
can be iterated continuously

*“Reinforcement learning
with human feedback”
(RLHF)*

Conclusions:

- labelers significantly prefer InstructGPT outputs over GPT-3 outputs
- Improvements in truthfulness
- promising generalization to instructions outside RLHF distribution
- But: InstructGPT still makes simple mistakes

“Pretrain & Finetune”

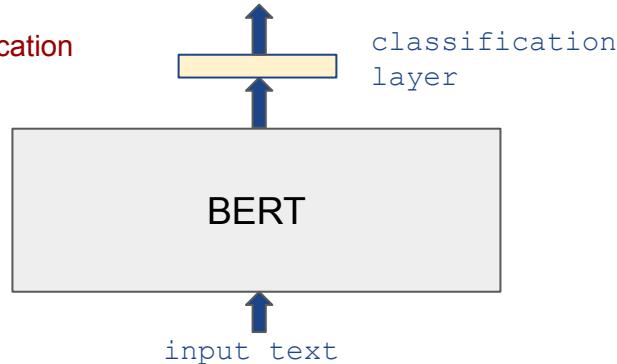
self-supervised pre-training of
transformer encoder and/or decoder

Key paradigm past few years:

- (1) Pretraining: foundation model
- (2) Fine-tune entire model (transformer *and* top layers) for downstream task on (smaller) corpus of annotated items (even few-shot)

Vanilla example:

BERT based text classification



“Pretrain, Prompt, Predict”

Currently the winning paradigm! (Since: November ‘22)

- (1) Start with a (large) foundation model (mostly autoregressive transformer)
- (2) Address it with a ‘prompt’
= *entire input to the model, in which the task description is embedded*

“Pretrain, Prompt, Predict”

Example of prompt engineering with GPT3.5, for the following task:

create a free-text definition for biomedical concepts
(based on database information, to avoid hallucination)

System: Assistant is a large language model, specialized in biomedical and clinical knowledge. It can answer questions about diseases, medications, and more. SnomedCT is a medical ontology, a standardized vocabulary of medical terms. It is reliable, and can be used to classify diseases and medications.

User: Let's talk about medical concepts. What does SnomedCT say about "{medical_concept}"?

Assistant: I found the following facts about "{medical_concept}" in SnomedCT:
{verbalized_snomed_facts}

User: Ok, thanks! Based on this, and your own medical knowledge, write a short definition of {medical_concept} in the style of MEDLINE or UMLS. Do not give a list of alternative names (also called) in the definition, the user already knows about them. Include some details about {required_details}. Leave out unimportant details if they are not useful inside a short definition. Start your reply immediately by the following words: Based on the given information, and my own medical knowledge, "{medical_concept}" refers to

Assistant:

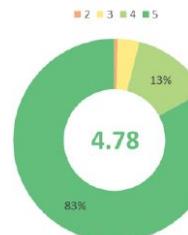
Remy and Demeester, "Automatic Glossary of Clinical Terminology: a Large-Scale Dictionary of Biomedical Definitions Generated from Ontological Knowledge", BioNLP 2023.

This is the prompt

“Pretrain, Prompt, Predict”

Example of prompt engineering with GPT3.5, for the following task:
create a free-text definition for biomedical concepts
(based on database information, to avoid hallucination)

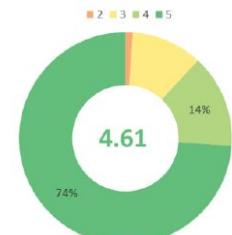
Distribution of Factuality



Distribution of Insight



Distribution of Fluency



X-linked retinitis pigmentosa refers to a genetic disorder that primarily affects the retina, and is associated with a dystrophy of this structure. It is inherited in an X-linked manner, which means that the gene causing the disorder is located on the X chromosome. This type of retinitis pigmentosa typically progresses more rapidly than other types, leading to visual impairment and eventual blindness in affected persons.

House converted to multiple dwellings refers to a type of building that was originally a single-family house but has been converted into multiple units or apartments, with separate entrances and often with shared common areas, such as hallways or yards. This type of dwelling is usually found in urban or suburban areas where housing is in high demand.

Fluorouracil 25mg/mL injection solution 10mL vial refers to a medication in the form of an injection that contains 25mg/mL of the active ingredient, fluorouracil, and is supplied in a vial with a volume of 10 mL. Fluorouracil is a chemotherapy drug used to treat a variety of cancers, including breast, gastrointestinal, and skin cancers. It is administered intravenously and is used either alone or in combination with other chemotherapy drugs.

Foundation models

Foundation model

= “*any model trained on broad data (generally using self-supervision at scale) that can be adapted (e.g., fine-tuned) to a wide range of downstream tasks.*” [*]

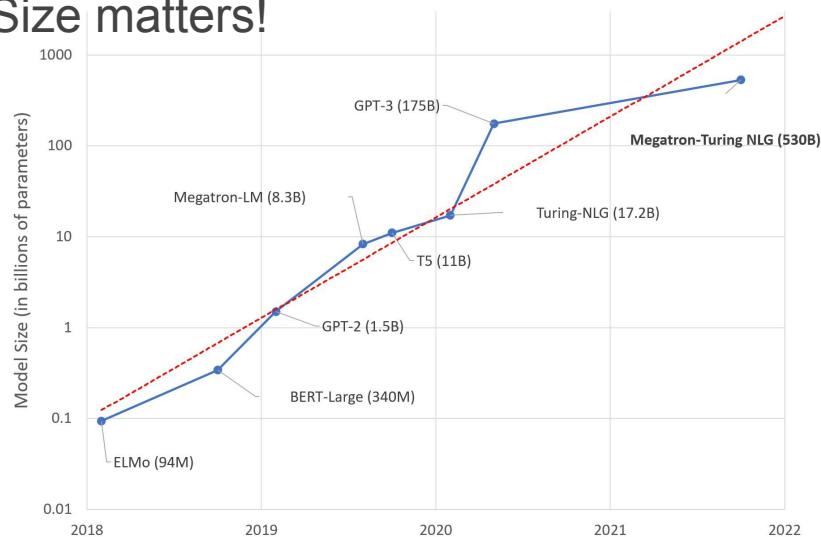
[*] Bommasani et al., “On the Opportunities and Risks of Foundation Models”, 2022,
<https://crfm.stanford.edu/report.html>

Foundation models

Foundation model

= “any model trained on broad data (generally using self-supervision **at scale**) that can be adapted (e.g., fine-tuned) to a wide range of downstream tasks.”

Size matters!



This ‘Moore’ scaling law will likely not continue (cf. **LLaMA**, **Alpaca**...)

June 2021: Wu Dao 2.0:
1.75t parameters (GPT-3 x 10!)
trained on text and images

March 2023: GPT-4:
Size??
trained on text and images

Foundation models

Foundation model

= “any model trained on broad data (generally using **self-supervision** at scale) that can be adapted (e.g., fine-tuned) to a wide range of downstream tasks.” [*]

2013-2014: static word representations ('word2vec')

2015-2018: autoregressive language models, contextual word representations

around 2019: transformer models

last few years: language model = ‘substrate’ of NLP

one model useful for wide range of tasks: ‘foundation models’

now: focus on autoregressive models; instruction-based training

ChatGPT

[*] Bommasani et al., “On the Opportunities and Risks of Foundation Models”, 2022,
<https://crfm.stanford.edu/report.html>

Foundation models

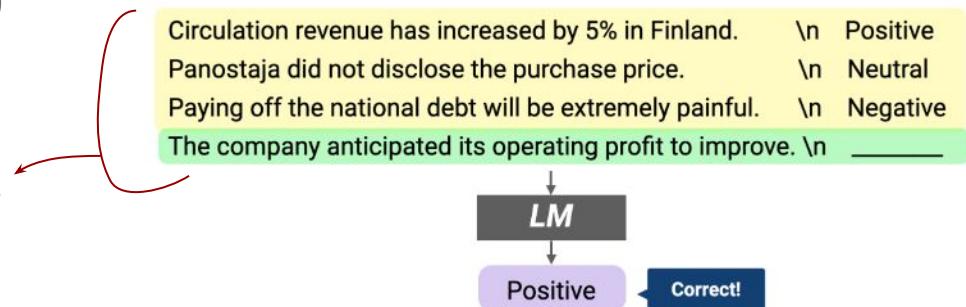
Foundation model

= “any model trained on broad data (generally using self-supervision at scale) that can be adapted (e.g., fine-tuned) to a wide range of downstream tasks.”

Only at scale: “emergent” behavior (not explicitly constructed)

key example: “in-context learning”

input (“prompt”) contains few input-output examples



Introduction to language models

ChatGPT

Prompting strategies

Q&A

What?

ChatGPT:

Chatbot that can write text and answer questions using AI.

Based on a generative language model:

No search engine!

(Does not search for answers, although it may say it does)

Trained on very large amounts of digital text

(books, wikipedia, web pages, online fora, code repositories...)

Key properties

- Autoregressive language model:
trained to predict the most likely next word (on a colossal text corpus)
- Further trained by trial & error on human feedback:
ChatGPT learns to mimic human preferences!
- Important: text generation is **not deterministic!**



*Each word iteratively generated via random sampling
(given the predicted distribution of the next word)*

```
response = openai.Completion.create(  
    engine="text-davinci-003",  
    prompt="On Friday evening, after hours of Q&A following his lecture on ChatGPT, "  
        + "Thomas went back to the iGent tower, took the elevator to the 9th floor, "  
        + "when he heard a sudden unsettling noise.",  
    max_tokens=300  
)  
print(response.choices[0].text.strip())
```

He was startled, but quickly rolled his eyes and closed them again.

```
response = openai.Completion.create(  
    engine="text-davinci-003",  
    prompt="On Friday evening, after hours of Q&A following his lecture on ChatGPT, "  
        + "Thomas went back to the iGent tower, took the elevator to the 9th floor, "  
        + "when he heard a sudden unsettling noise.",  
    max_tokens=300  
)  
print(response.choices[0].text.strip())
```

He tried to ignore the noise and continued on his way, but he soon realized that it was coming from inside the elevator. He looked up and saw that the elevator was moving on its own – without the controls being operated!

Thomas quickly became worried; he hadn't heard of elevator malfunctions like this before. He kept his eye on the elevator display as it slowly began to ascend further and further up. Eventually it came to a stop on the 18th floor. When the door opened, Thomas stepped out, expecting to find himself in the lobby, but what he saw was far from it.

Instead, Thomas found himself in a small, dark room. In the corner was a figure shrouded in a black cloak, and as Thomas cautiously approached, the figure lowered its hood to reveal a pale face belonging to a strange woman.

The woman gave Thomas a sinister smile and said, "Welcome to my realm. I have something I wish to show you..."

By the way, I cheated...

TH

give example python code to query text-davinci-003 with a simple prompt



To make a query to the Text-Davinci-003 model (or any OpenAI's GPT models), you would typically use the OpenAI API. Here is an example of how to do that in Python.

python

Copy code

```
import openai

openai.api_key = 'your-api-key'

response = openai.Completion.create(
    engine="text-davinci-003",
    prompt="Translate the following English text to French:
max_tokens=60
)

print(response.choices[0].text.strip())
```

What can it do?

- ChatGPT = chatbot that responds to input text ('prompt') in well phrased, often convincing, natural language.
- This chatbot can
 - write texts about a given topic, in multiple languages
 - summarize texts, translate, restructure, and correct
 - apply a given style in the generated text
 - generate code, based on instructions in natural language

What else can it do?

- Not only “write a text about...”
- Also:
 - extract given elements from a text
 - detect the occurrence of sentiments in text and synthesize the results
 - analyse linguistic elements in sentences
 - debug computer code and explain the problems with it

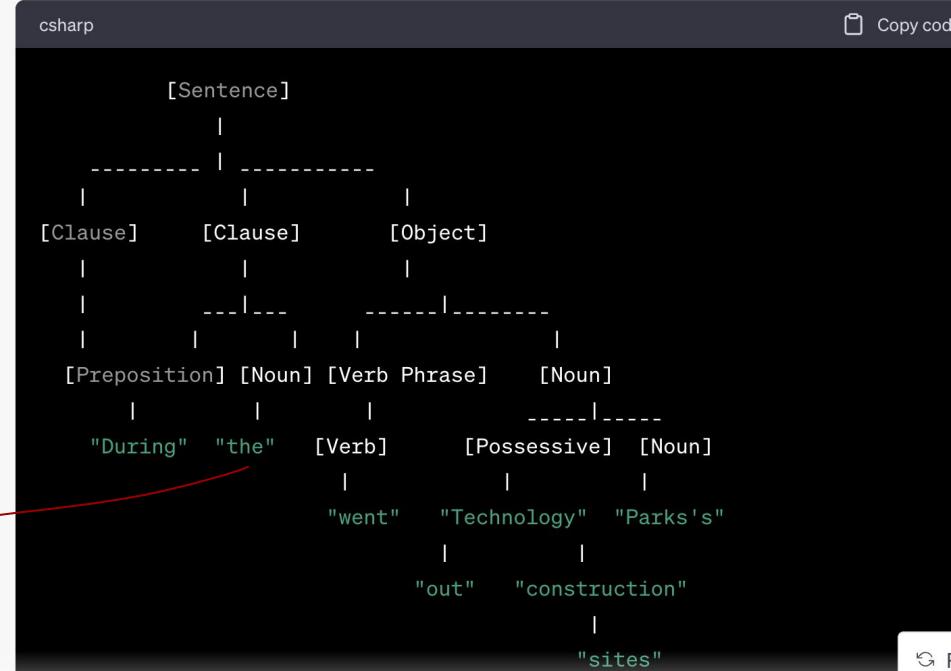
not just a writing assistant
also an analysis tool

TH

Create a parse tree for the sentence: "During the lecture, I went out to explore the Technology Parks's construction sites."



Creating a parse tree for a sentence involves breaking down the sentence into its grammatical components and representing it in a tree structure. Here's a simplified parse tree for the sentence: "During the lecture, I went out to explore the Technology Parks's construction sites."



Regenerate

Limitations

- It hallucinates sometimes (factually wrong answers, non-existing sources)
- No explicit reasoning capacities
- May generate biased text!
Trained on subjective and biased sources. Strong focus on safety mechanisms.
- GPT3.5: trained on data until end of 2021 (no recent data)

TH

Is it possible to state the gender of the first female US president?



No, it is not possible to state the gender of the first female US president with certainty until it actually happens. The gender of the first female US president will depend on future elections, campaigns, and the choices made by voters. It could be any eligible female candidate who runs for and wins the presidency in the future. The outcome will be determined by the democratic process and the choices made by the electorate at the time of the election.



The large ball crashed right through the table because it was made of steel. What was made of steel?



The sentence "The large ball crashed right through the table because it was made of steel" suggests that the large ball was made of steel, not the table.



The large ball crashed right through the table because it was made of styrofoam. What was made of styrofoam?



Based on the sentence "The large ball crashed right through the table because it was made of styrofoam," it is the ball that was made of styrofoam, not the table.



Welke zijn de laatste 5 decimalen van pi?

⚡ Model: Default (GPT-3.5)



De laatste 5 decimalen van pi zijn 92653.

TH

welke zijn de laatste 5 decimalen van pi?

TH

Welke zijn de 5 laatste decimalen van pi?



De laatste vijf decimalen van pi zijn 14159.



De vijf laatste decimalen van π (pi) zijn 92653.

TH

What are the last 5 digits of pi?

(ChatGPT 2023-02-17,
GPT3.5, 2023-05-26,
GPT3.5, 2023-07-11)



The last five digits of pi are 14159.



Kan het dat mijn zus haar schoonzus ook mijn vrouw is?



Nee, het is niet mogelijk dat uw zus haar schoonzus tegelijkertijd uw vrouw is.

Een schoonzus is de echtgenote van de broer of zus van iemand anders. Dus als uw zus getrouwd is, dan is haar schoonzus de vrouw van haar echtgenoot's broer of zus.

Aan de andere kant, als u getrouwd bent, dan is uw vrouw uw echtgenote en niet de schoonzus van uw zus. Het is niet mogelijk dat iemand tegelijkertijd een schoonzus en een echtgenote is van dezelfde persoon.



Oei, het is nochthans zo. Ik woon in België, kan ik hierdoor in wettelijke problemen komen?



Als uw zus getrouwd is met een broer van uw partner, dan is uw zus de schoonzus van uw partner en bent u getrouwd met uw schoonzus. In België is dit echter niet toegestaan en wordt dit beschouwd als een huwelijk tussen naaste verwanten, wat wettelijk verboden is.

Als u momenteel getrouwd bent met uw schoonzus, dan zou u inderdaad in wettelijke problemen kunnen komen. Het is belangrijk om deze situatie zo snel mogelijk aan te pakken en te zoeken naar een wettelijke oplossing, bijvoorbeeld door het huwelijk te ontbinden en eventueel opnieuw te trouwen met iemand anders.

TH

can it be that my sister's sister-in-law is also my wife?



No, it is not possible for your sister's sister-in-law to also be your wife. Let's break down the relationships to clarify:

- Your sister: This is your sibling.
- Sister-in-law: This term typically refers to the sister of your spouse. In other words, your brother's wife.
- Your wife: This is the person you are married to.

Based on these definitions, your sister's sister-in-law cannot be your wife because your sister's sister-in-law would be the spouse of your sister's brother, not you.

TH

Oh, it actually holds true for me. Can I get into trouble in Belgium?



If you are in a situation where your sister's sister-in-law is also your wife, it implies a complex and unconventional family arrangement. While I cannot provide legal advice, I can offer some general information.

Marriage laws and regulations vary from country to country. In some jurisdictions, such as Belgium, marriage between certain close relatives is prohibited due to concerns about genetic disorders and societal norms. Typically, marriage between siblings, including half-siblings, is not permitted.

TH

Can it be that my sister's sister-in-law is also my wife?



Yes, it is possible. If your sister's sister-in-law is her spouse's sister, and you are married to that woman, then your sister's sister-in-law would indeed be your wife. In this scenario, your sister is married to your wife's brother. This would make you and your sister's husband siblings-in-law, and your wife and your sister would also be siblings-in-law.

TH

tel het aantal i's in 'mississippi'



Het aantal 'i's in 'mississippi' is 4.

TH

tel het aantal i's in 'mississippi-river'



Het aantal 'i's in 'mississippi-river' is 4.

TH

There are five birds on a branch. Shoot one bird off the branch. How many are left on the branch?



If one bird is shot off the branch, the answer is that there will be zero birds left on the branch. Shooting a bird off the branch implies that it is no longer on the branch, resulting in none remaining.

TH

There are five birds on a branch. Shoot one bird off the branch. How many are left on the branch? Give intermediate steps.



If there are five birds on a branch and you shoot one bird off the branch, the intermediate steps are as follows:

1. Initially, there are five birds on the branch.
2. You shoot one bird off the branch.
3. After shooting one bird, there are four birds left on the branch.

So, there are four birds left on the branch after shooting one bird off.

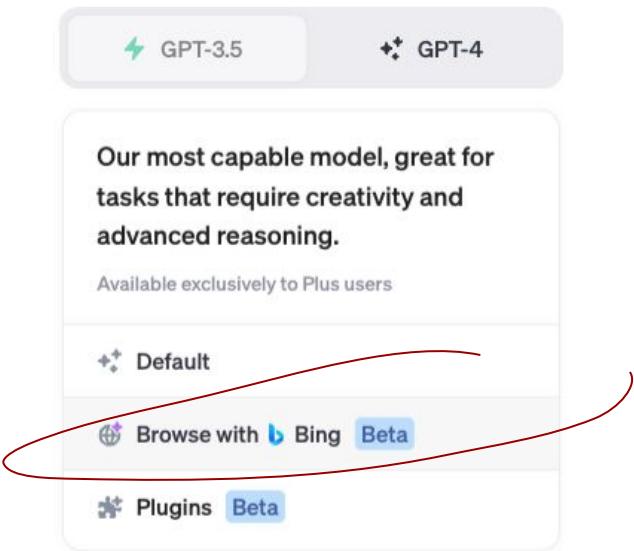
Interesting possibilities

Language model by itself:

- lack of recent data
- warnings / hallucinations

In combination with web search:

- correct references (and links)
- access to recent data



TH

Who's UGent's Bo Kang?



I'm sorry, but as of my last knowledge update in September 2021, I do not have information about an individual named Bo Kang associated with UGent (Ghent University) or any other notable information about them. It's possible that this person became known or relevant after my last update, or they may not be widely recognized in public sources of information.



If Bo Kang is a recent addition to UGent or has gained prominence in their field after my last update, I would recommend checking UGent's official website or conducting an internet search for the most up-to-date and relevant information about this individual.

TH

Who's UGent's Bo Kang?



Finished browsing



Bo Kang is a postdoctoral researcher at Ghent University (UGent) in Belgium, affiliated with the Department of Electronics and Information Systems under the Faculty of Engineering and Architecture¹. He has held the position of postdoctoral researcher since January 2020, and is involved in various projects with the Flanders AI Research (FLAIR)². Bo Kang obtained his Ph.D. degree in Computer Science Engineering from Ghent University in 2019, and is associated with the IDLab at the same university³. Prior to his postdoctoral position, he was a PhD student at Ghent University from 2015 to 2019 and had a brief stint as a PhD student at the University of Bristol in 2015⁴.

His research interests and activities, while not explicitly outlined, appear to be in the realm of data mining, machine learning, and artificial intelligence, as reflected by his affiliation with FLAIR and the projects he's been part of. His academic and professional profiles can be found on various platforms including Google Scholar, DBLP, ORCID, and LinkedIn⁴.

⟳ Regenerate

Privacy concerns?

TRENDING Best standing desk deals Best cloud storage 2023 What is AI?

When you purchase through links on our site, we may earn an affiliate commission. [Here's how it works.](#)

Pro > Software & Services

Samsung workers made a major error by using ChatGPT

News By Lewis Maddison published April 04, 2023

Samsung meeting notes and new source code are now after being leaked in ChatGPT

[Facebook](#) [Twitter](#) [Pinterest](#) [Flipboard](#) [Email](#)



<https://www.techradar.com/news/samsung-workers-leak-chatgpt-data>

<https://openai.com/blog/march-20-chatgpt-outage/#e>

<https://www.dw.com/en/ai-italy-lifts-ban-on-chatgp>

IN FOCUS Wagner Group Russia's war in Ukraine H → Latest videos Live TV Latest audio

TECHNOLOGY | ITALY

Italy lifts ban on ChatGPT after data privacy improvements

04/29/2023

The hotly debated AI chatbot is back online in Italy after installing new warnings for users and the option to opt-out of having chats be used to train ChatGPT's algorithms.

f t v



OpenAI March 24, 2023 Authors OpenAI ↓

We took ChatGPT offline earlier this week due to a bug in an open-source library which allowed some users to see titles from another active user's chat history. It's also possible that the first message of a newly-created conversation was visible in someone else's chat history if both users were active around the same time.

The bug is now patched. We were able to restore both the ChatGPT service and, later, its chat history feature, with the exception of a few hours of history. As promised, we're publishing more [technical details](#) of this problem below.

Upon deeper investigation, we also discovered that the same bug may have caused the unintentional visibility of payment-related information of 1.2% of the ChatGPT Plus subscribers who were **active** during a specific nine-hour window. In the hours before we took ChatGPT offline on Monday, it was possible for some users to see another **active** user's first and last name, email address, payment address, credit card number, and the last four digits (only) of a credit card number, and credit card expiration date. These details were not exposed at any time.

Privacy concerns?

Risks associated with the use of ChatGPT concerning privacy and data protection?

Ter info: <https://help.openai.com/en/articles/5722486-how-your-data-is-used-to-improve-model-performance>



The screenshot shows the top navigation bar of the OpenAI website. It features the OpenAI logo on the left, a search bar with the placeholder "Search for articles...", and a "Go to OpenAI" button on the right.

All Collections > OpenAI API > Policy > How your data is used to improve model performance

How your data is used to improve model performance

Learn more about OpenAI's data usage policies for our API, ChatGPT and DALL-E

 Written by Yaniv Markovski
Updated this week

One of the most useful and promising features of AI models is that they can improve over time. We continuously improve our models through research breakthroughs as well as exposure to real-world problems and data. When you share your data with us, it helps our models become more accurate and better at solving your specific problems and it also helps improve their general capabilities and safety. We don't use data for selling our services, advertising, or building profiles of people—we use data to make our models more helpful for people. ChatGPT, for instance, improves by further training on the conversations people have with it, unless you choose to disable training.

ChatGPT

When you use our non-API consumer services ChatGPT or DALL-E, we may use the data you provide us to improve our models. You can switch off training in ChatGPT settings (under Data Controls) to turn off training for any conversations created while training is disabled or you can submit [this form](#). Once you opt out, new conversations will not be used to train our models.

API

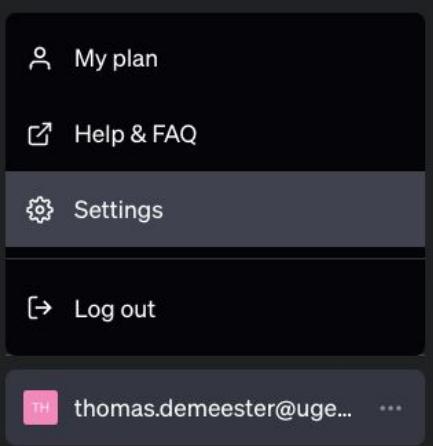
OpenAI does not use data submitted by customers via our API to train OpenAI models or improve OpenAI's service offering. In order to support the continuous improvement of our models, you can fill out [this form](#) to opt-in to share your data with us.

What the process looks like

We retain certain data from your interactions with us, but we take steps to reduce the amount of personal information in our training datasets before they are used to improve our models. This data helps us better understand user needs and preferences, allowing our model to become more efficient over time.

For details on our data policy, please see our [Data Usage Policies](#), [Privacy Policy](#) and [Terms of Use](#).

Privacy concerns?



Settings

General

Beta features

Data controls

Chat history & training

Save new chats on this browser to your history and allow them to be used to improve our models. Unsaved chats will be deleted from our systems within 30 days. This setting does not sync across browsers or devices. [Learn more](#)

Shared links Manage

Export data Export

Delete account Delete

Disable two factor authentication Disable

Introduction to language models

ChatGPT

Prompting strategies

Q&A

Examples in next slides:
GPT3.5, July 12, 2023

Interesting prompting strategies

- Describe your question in detail.

Many useful online resources!

<https://platform.openai.com/docs/guides/chat>

<https://the-decoder.com/chatgpt-guide-prompt-strategies/> (inspiration for the next few slides)

<https://prompts.chat/>

...

how to make emotion detection data



To create emotion detection data, you typically need to follow these steps:



1. Define the Emotions: Decide on the specific emotions you want to detect. Emotions can include happiness, sadness, anger, fear, surprise, disgust, and others. Be clear about the emotions you want to detect, as it will guide the labeling process.
2. Gather a Diverse Dataset: Collect a diverse dataset that includes examples of different emotions. You can obtain data from various sources, such as social media platforms, online forums, or existing emotion-labeled datasets. Ensure that the dataset covers a range of individuals, cultures, languages, and contexts to capture the variability in emotional expressions.
3. Preprocess the Data: Clean and preprocess the collected data to remove any noise or irrelevant information. This step may involve removing duplicates, handling missing values, standardizing text formats, and anonymizing personally identifiable information if required.
4. Label the Data: Assign emotion labels to each data sample. This step can be done manually by human annotators who are trained to identify and classify emotions accurately. Annotators may use guidelines or predefined criteria to ensure consistency across the labeled dataset. Alternatively, you can use automated techniques, such as pre-trained emotion detection models, to provide initial labels, which can then be reviewed and corrected by human annotators.

5. Establish Annotation Guidelines: Develop clear guidelines for annotators to follow. These guidelines should provide instructions on how to identify and label emotions in the dataset. Include examples and edge cases to help annotators understand the nuances of emotional expressions and improve consistency in the labeling process.
6. Quality Control: Implement a quality control process to ensure the accuracy and reliability of the labeled data. This can involve multiple rounds of review, inter-annotator agreement checks, and resolving disagreements through discussions or additional training for annotators.
7. Split the Data: Divide the labeled dataset into training, validation, and testing subsets. The training set is used to train the emotion detection model, the validation set helps fine-tune the model's parameters, and the testing set evaluates the model's performance on unseen data.
8. Optional: Augment the Data: If you have limited labeled data, you can apply data augmentation techniques to artificially increase the size of your dataset. This can involve techniques such as adding noise, altering sentence structure, or replacing words with their synonyms while maintaining the original emotion.
9. Train and Evaluate the Model: Use the labeled dataset to train an emotion detection model. You can choose from various machine learning or deep learning approaches, such as Support Vector Machines (SVMs), Recurrent Neural Networks (RNNs), or Transformer-based models. Evaluate the model's performance using appropriate metrics, such as accuracy, precision, recall, or F1 score.

Give a **brief overview of required steps** to create a training set for the detection of emotion trajectories in customer service dialogues. **Give 2 examples of short conversations** (each with at least 3 turns), with indication of the emotion for user utterances, as well as valence-arousal-dominance annotations (on a five-point scale).



Creating a training set for the detection of emotion trajectories in customer service dialogues involves the following steps:

1. **Task Definition:** Define the goal of detecting emotion trajectories in customer service dialogues.
2. **Data Collection:** Gather customer service dialogues, either from existing datasets or by collecting new dialogues through recordings or transcripts.
3. **Annotation Guidelines:** Develop guidelines for annotators to label user utterances with emotion categories and provide valence-arousal-dominance (VAD) annotations on a five-point scale.
4. **Annotator Training:** Train annotators on the guidelines, ensuring consistency in understanding and applying emotion labels and VAD annotations.
5. **Annotation Process:** Annotate user utterances in the customer service dialogues with appropriate emotion labels and VAD annotations based on the expressed or implied emotions.
6. **Quality Control:** Implement measures for inter-annotator agreement, regular checks, and feedback to ensure annotation quality and consistency.
7. **Data Preprocessing:** Clean and preprocess the annotated dataset, including removing irrelevant information, normalizing text, and tokenizing the dialogues.
8. **Dataset Splitting:** Divide the dataset into training, validation, and test sets, maintaining a balance of emotions and trajectories across the splits.



9. **Baseline Evaluation:** Train a baseline model using the annotated dataset and evaluate its performance on the validation and test sets to establish a baseline performance.
10. **Iteration and Improvement:** Analyze the model's performance, refine annotation guidelines if necessary, collect additional data if required, and iterate to improve the training set and model performance.

Examples of short conversations with emotion labels and VAD annotations (on a five-point scale):

1. User: *I've been waiting for my order for a month! This is ridiculous!* (Emotion: Frustration, VAD: 1, 4, 4)
Agent: *I apologize for the delay. Let me check the status for you.*
User: *Please do it quickly. I'm extremely annoyed.* (Emotion: Anger, VAD: 1, 5, 5)
2. User: *Thank you for your quick response! I appreciate it.* (Emotion: Gratitude, VAD: 4, 2, 3)
Agent: *You're welcome! Is there anything else I can assist you with?*
User: *No, that was all. Thanks again!* (Emotion: Gratitude, VAD: 5, 2, 2)

Note: These examples demonstrate short conversations in customer service dialogues, with at least three turns. The user utterances are labeled with corresponding emotions, and VAD annotations are provided on a five-point scale (valence, arousal, dominance). These annotations capture the emotional trajectories within the dialogues, enabling the training of models for emotion detection and analysis in customer service interactions.

Interesting prompting strategies

- Describe your question in detail.
- Define ChatGPT's role and the target audience

TH

List 5 key uses of ChatGPT.



Sure! Here are five key uses of ChatGPT:



1. Customer Support: ChatGPT can be used to provide automated customer support by answering frequently asked questions, assisting with common issues, and guiding users through troubleshooting steps. It can handle a wide range of inquiries and help resolve customer concerns efficiently.
2. Personal Assistant: ChatGPT can act as a virtual personal assistant, helping users manage their schedules, set reminders, provide information, and perform tasks like sending emails or making online reservations. It can understand natural language queries and assist users in various day-to-day activities.
3. Language Translation: ChatGPT can be utilized for language translation purposes, allowing users to communicate across language barriers. It can translate text or assist in generating translations by providing suggestions and context-based understanding.
4. Content Generation: ChatGPT can generate human-like text in response to prompts, making it useful for content creation. It can help with writing articles, blog posts, social media updates, product descriptions, and more. Writers and marketers can leverage ChatGPT's creative capabilities to streamline their content generation process.
5. Education and Learning: ChatGPT can serve as an interactive learning tool, providing explanations, answering questions, and engaging in discussions on various topics. It can help students with homework, provide explanations for complex concepts, and offer personalized learning experiences by adapting to the user's level of understanding.

TH

You are a writing coach, talking to journalists. List 5 key uses of ChatGPT.



As a writing coach for journalists, here are five key uses of ChatGPT specifically tailored to their needs:

1. Research Assistance: ChatGPT can assist journalists in conducting research by providing relevant information, statistics, historical context, or references on a given topic. It can help journalists gather data, verify facts, and explore different angles for their stories.
2. Idea Generation: ChatGPT can be a valuable tool for brainstorming and generating story ideas. Journalists can provide prompts or keywords, and ChatGPT can offer suggestions, angles, or even potential sources to explore. It can help spark creativity and uncover unique perspectives for engaging articles.
3. Interview Preparation: ChatGPT can simulate interview scenarios, enabling journalists to practice and refine their interview skills. It can provide example questions, offer insights into potential responses, and help journalists anticipate and prepare for different interview situations.
4. Writing and Editing Support: ChatGPT can act as a virtual writing assistant, providing suggestions for sentence structure, grammar, vocabulary, and style. Journalists can receive real-time feedback on their writing, helping them improve clarity, conciseness, and overall quality of their articles.
5. Breaking News Updates: ChatGPT can keep journalists informed about breaking news, providing real-time updates on current events, trends, or specific topics of interest. It can aggregate information from various sources and present concise summaries, allowing journalists to stay up-to-date and react swiftly to developing stories.



TH

You are an expert translator, talking to students aiming to be professional translators. List 5 key uses of ChatGPT.



As an expert translator advising students aiming to become professional translators, here are five key uses of ChatGPT in the field:

1. Language Assistance: ChatGPT can assist translators by providing language support and addressing questions related to grammar, vocabulary, idiomatic expressions, and word usage. It can serve as a valuable resource for language learners, helping them deepen their understanding of the target and source languages.
2. Translation Aid: ChatGPT can act as a translation aid, offering suggestions and alternative translations for specific words, phrases, or sentences. It can help translators overcome challenges and provide them with additional options to consider, enhancing the quality and accuracy of their translations.
3. Terminology Research: ChatGPT can assist in terminology research, helping translators find accurate and contextually appropriate translations for specialized terms in various fields such as legal, medical, technical, or scientific. It can provide suggestions, definitions, and examples to ensure precise usage.
4. Style and Tone Guidance: ChatGPT can provide guidance on the appropriate style, tone, and register for different types of texts or target audiences. Translators can seek advice on adapting the tone of the source text to match the desired effect in the target language, ensuring effective communication in the translated content.
5. Cultural Context: ChatGPT can help translators navigate cultural nuances and considerations in their translations. It can provide insights into cultural references, idioms, customs, and other specific elements that require adaptation in the target language. This ensures that translations are culturally sensitive and resonate with the intended audience.



Interesting prompting strategies

- Describe your question in detail.
- Define ChatGPT's role and the target audience
- Specify style and format

TH

You are a writing coach, talking to journalists. List 5 key uses of ChatGPT. Subtitles in boldface; keyphrases only.



Certainly! Here are five key uses of ChatGPT for journalists, with subtitles in boldface and keyphrases only:



1. Research Assistance:

- Background information, statistics, relevant sources.

2. Interview Preparation:

- Simulate interviews, practice questioning, anticipate answers.

3. Fact-Checking Support:

- Cross-referencing, verify statements, highlight inaccuracies.

4. Writing and Editing Assistance:

- Improve sentence structure, clarity, style suggestions, grammar, word choice.

5. Story Generation:

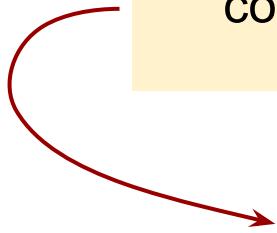
- Prompts and topics, unique angles, perspectives, potential sources.

By leveraging ChatGPT for these key uses, journalists can enhance their research, interviewing, fact-checking, writing, and story development processes. However, it's crucial to independently verify information from reliable sources to maintain journalistic integrity.

Interesting prompting strategies

- Describe your question in detail.
- Define ChatGPT's role and the target audience
- Specify style and format
- 'Chained prompting' (first global structure, than zoom in)

Goal: write a master thesis on using computational linguistics in healthcare



Chained prompting:

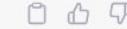
- first develop ideas
- then create title, high-level structure
- next, zoom in on specific sections
- ...

TH

Propose a specific research idea where computational linguistics techniques are applied to the healthcare domain.



Research Idea: Applying Computational Linguistics Techniques for Automated Analysis of Patient Medical Records



Objective:

The objective of this research is to develop and apply computational linguistics techniques to automatically analyze and extract valuable information from patient medical records. By leveraging natural language processing (NLP) and machine learning approaches, the research aims to improve healthcare processes, patient care, and clinical decision-making.

Methodology:

1. Data Collection: Gather a large dataset of anonymized patient medical records from different healthcare providers. These records should include a variety of text formats, such as discharge summaries, progress notes, diagnostic reports, and physician orders.
2. Text Preprocessing: Clean and preprocess the textual data by removing irrelevant information (e.g., patient identifiers) and standardizing the formatting (e.g., removing line breaks, special characters, etc.). Apply techniques like tokenization, sentence

Propose a title and high-level structure for a master thesis on this topic.



 Title: Automated Analysis of Patient Medical Records using Computational Linguistics Techniques: Improving Healthcare Processes and Clinical Decision-making

High-Level Structure:

1. Introduction

- Background and motivation
- Research objectives and scope
- Significance and potential impact of the research

2. Literature Review

- Overview of computational linguistics techniques in healthcare
- Previous studies on automated analysis of medical records
- NLP and machine learning approaches for healthcare data analysis
- Applications of sentiment analysis and relationship extraction in healthcare

3. Data Collection and Preprocessing

- Description of the dataset and its characteristics
- Data cleaning and preprocessing techniques applied
- Tokenization, sentence splitting, and part-of-speech tagging

4. Named Entity Recognition (NER) for Healthcare

- Introduction to NER and its relevance in medical records
- Design and implementation of a healthcare-specific NER model
- Training and evaluation of the NER model on the medical records dataset

5. Relationship Extraction in Medical Records

- Overview of relationship extraction techniques
- Development of relationship extraction models for medical records
- Extraction of key relationships between medical entities

6. Sentiment Analysis in Patient Narratives

- Introduction to sentiment analysis and its importance in healthcare
- Design and implementation of a sentiment analysis model
- Analyzing patient narratives for sentiment and emotional tone

7. Integration with Clinical Decision Support System

- Incorporating the extracted information into a clinical decision support system
- Development of algorithms for decision support based on the analysis results
- Evaluation of the system's impact on clinical decision-making

8. Evaluation and Validation

- Evaluation metrics for assessing the performance of the developed techniques
- Comparison with manual annotations or expert judgments
- Statistical analysis of results and discussion of findings

9. Ethical Considerations

- Discussion of ethical concerns related to patient privacy and data security
- Compliance with regulatory guidelines (e.g., HIPAA)
- Informed consent and anonymization of patient data

10. Conclusion and Future Directions

- Summary of the research findings
- Contributions and limitations
- Potential future research directions in automated medical record analysis

 Regenerate response

Send a message 

TH

Write out the chapter on relationship extraction from medical records. List section and subsection titles.



Chapter 5: Relationship Extraction in Medical Records

- 5.1 Introduction to Relationship Extraction
- 5.2 Preprocessing Medical Text for Relationship Extraction
- 5.3 Rule-based Relationship Extraction Approaches
- 5.4 Machine Learning-Based Relationship Extraction Models
 - 5.4.1 Feature Engineering for Relationship Extraction
 - 5.4.2 Training Supervised Machine Learning Models
- 5.5 Neural Network-Based Relationship Extraction Models
- 5.6 Evaluation of Relationship Extraction Models
- 5.7 Case Studies and Application Scenarios
- 5.8 Limitations and Challenges
- 5.9 Future Directions and Research Opportunities

TH

Write a summary for section 5.4 (ML-based relationship extraction models).



Section 5.4: Machine Learning-Based Relationship Extraction Models

Machine learning-based models offer a powerful approach to relationship extraction in medical records. This section explores the utilization of supervised machine learning techniques for this task. It covers two main aspects: feature engineering for relationship extraction and training supervised machine learning models.

In subsection 5.4.1: Feature Engineering for Relationship Extraction, the focus is on identifying relevant features that can capture the relationship information within medical records. Various types of features are discussed, including syntactic, lexical, and contextual features. The section highlights the importance of selecting features that can effectively represent the relationships between entities in the medical text. It also touches upon the use of embedding-based representations, such as word embeddings or contextualized word embeddings, for enhancing relationship extraction performance.

Subsection 5.4.2: Training Supervised Machine Learning Models delves into the process of training machine learning models for relationship extraction. It begins with an overview of different machine learning algorithms commonly used in this context, such as Support Vector Machines (SVM) and Conditional Random Fields (CRF). The section then discusses the preparation of training data and the

Regenerate response



Send a message



Interesting prompting strategies

- Describe your question in detail.
- Define ChatGPT's role and the target audience
- Specify style and format
- 'Chained prompting' (first global structure, than zoom in)

- chain-of-thought prompting
- add examples to the prompt (cf. 'in-context learning')
- ask to rewrite previous output (possible based on self-criticism)

Thanks for your attention!

time for questions

thomas.demeester@ugent.be