UNIVERSITÄT
ZU KÖLN

Generative and Large Language Models (LLMs)

VL Sprachliche Informationsverarbeitung

Nils Reiter
nils.reiter@uni-koeln.de

January 9, 2025
Winter term 2024/25

INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

Introduction

- Skipped: RNN/LSTM era
  - RNN/Bi-LSTM have taken over NLP landscape – 2015–2018
  - Powerful machine learning, usable for many different tasks
- Today: Two new ML paradigms
  - Pretraining/finetuning
  - Prompting

Introduction

- Skipped: RNN/LSTM era
  - RNN/Bi-LSTM have taken over NLP landscape – 2015–2018
  - Powerful machine learning, usable for many different tasks
- Today: Two new ML paradigms
  - Pretraining/finetuning
  - Prompting

Section 1

Paradigm 1: Pre-Training/Fune-Tuning

Introduction

- Popularized with BERT and Transformer architecture

  Devlin et al. (2019); Vaswani et al. (2017)

- Models are pre-trained on large data sets
  - Pre-Training requires significant resources (time/money)
- Pre-trained models are (often, not always) shared "freely"
- Recipe: Take a pre-trained model and fine-tune it on your task
  - Pre-trained model contains an abstract language representation

Introduction

- Popularized with BERT and Transformer architecture

  Devlin et al. (2019); Vaswani et al. (2017)

- Models are pre-trained on large data sets
  - Pre-Training requires significant resources (time/money)
- Pre-trained models are (often, not always) shared "freely"
- Recipe: Take a pre-trained model and fine-tune it on your task
  - Pre-trained model contains an abstract language representation
- Fine-tuning
  - Any language-related task
  - Requires significantly less training data

    ⚠ This is the game changer for applications!

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

Figure: Examples of attending to the correct object (Bahdanau et al., 2015)

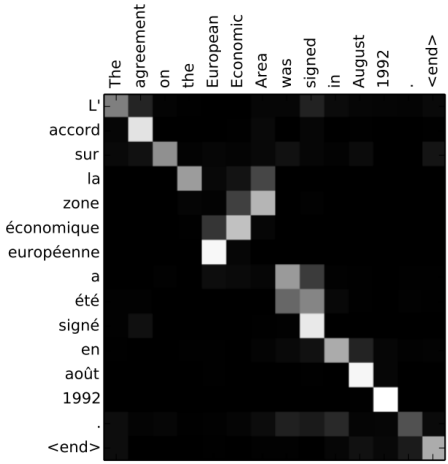


Figure: Attention paid by a neural machine translation network (Bahdanau et al., 2015)

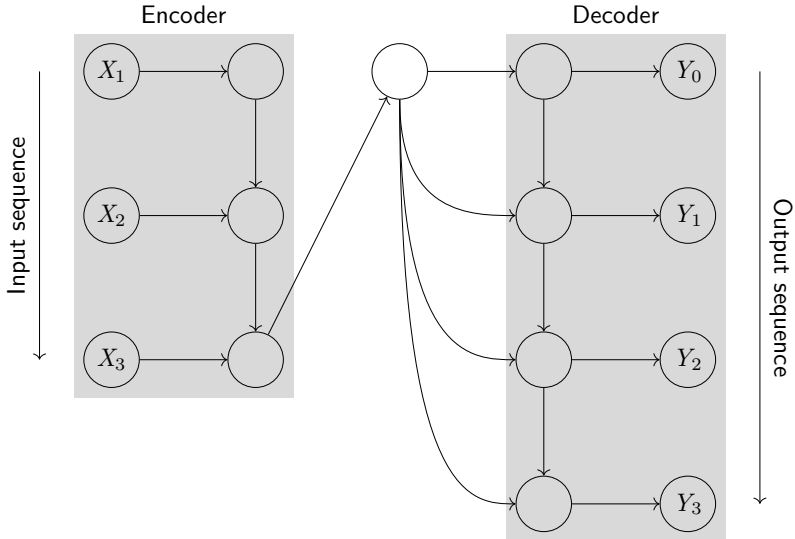
Introduction

- A mechanism to allow the network to learn what to focus on
- Idea: Not all parts of the input are equally important
  - MT: “la zone économique européenne” → “the European Economic Area”, irrespective of context

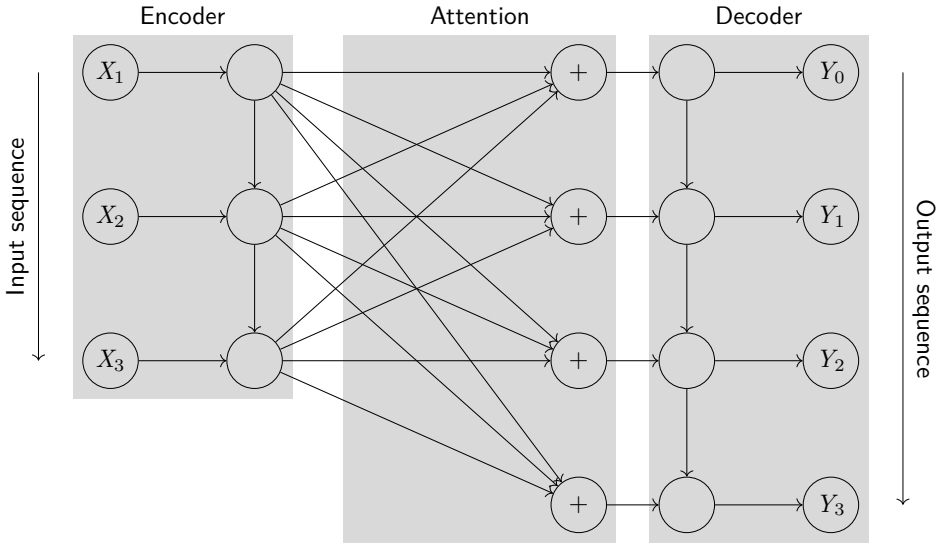
Introduction

- A mechanism to allow the network to learn what to focus on
- Idea: Not all parts of the input are equally important
  - MT: “la zone économique européenne” → “the European Economic Area”, irrespective of context
- Mirrors human reading/translating activities
- Developed for machine translation, then applied to other tasks

From Encoder-Decoder to Attention



From Encoder-Decoder to Attention



Pre-Training BERT

- General idea — Devlin et al. (2019)
  - Encoder-Attention-Decoder architecture (= transformer)
  - Process whole input at once, no sequence labeling! (max. 512 tokens, = bidirectional)
  - Pre-training and fine-tuning on different tasks

BERT Pre-Training Tasks

Masked Language Modeling (MLM)

- Sentence-wise
- 15% of the tokens are “masked” by a special token
- Model predicts these, having access to all other tokens

BERT Pre-Training Tasks

Masked Language Modeling (MLM)

- Sentence-wise
- 15% of the tokens are “masked” by a special token
- Model predicts these, having access to all other tokens

Next sentence prediction (NSP)

- Two (masked) sentences are concatenated
- Model has to predict wether second sentence follows on the first or not

Section 2

Paradigm 2: Prompting

Introduction

- Prompting = "In-Context-Learning"
  - Telling the model what to do in natural language
  - "Emergent ability": Present in larger models, but not in smaller ones Wei et al. (2022)
  - No update of any weights, i.e., no actual training

Introduction

- Prompting = "In-Context-Learning"
  - Telling the model what to do in natural language
  - "Emergent ability": Present in larger models, but not in smaller ones Wei et al. (2022)
  - No update of any weights, i.e., no actual training
- Prompting as an alternative to pre-training/fine-tuning?
  - It's appealing: Natural language prompts
    - No time-consuming annotation process
    - No programmer / no training process
    - Direct(er) control

Prompting Scenarios

1 Interactive Prompting

- Chat-bot scenario à la ChatGPT
- Using prompting to solve a single specific task

2 Automatic Detection with Prompts – 'Batch Prompting'

- Prompts to automatically detect text properties
- Replacement for pre-training/fine-tuning scenarios

Prompting Scenarios

1 Interactive Prompting

- Chat-bot scenario à la ChatGPT
- Using prompting to solve a single specific task

2 Automatic Detection with Prompts – 'Batch Prompting'

- Prompts to automatically detect text properties
- Replacement for pre-training/fine-tuning scenarios

- Contexts of discovery and justification Gerstorfer (2020); Reichenbach (1938)
  - "the well-known difference between the thinker's way of finding this theorem and his way of presenting it before a public" Reichenbach (1938, 5)

1 Interactive Prompting

- Direct use and implicit validation
- Results don't have to be perfect to be useful
- Users make connections and fill holes
- Strategies involve different components (e.g., positive/negative examples, definitions, ...)
- Rarely documented in scientific articles
- A lot of "anecdotal evidence"

1 Interactive Prompting

- Direct use and implicit validation
- Results don't have to be perfect to be useful
- Users make connections and fill holes
- Strategies involve different components (e.g., positive/negative examples, definitions, ...)
- Rarely documented in scientific articles
- A lot of "anecdotal evidence"

Context of Discovery

- LLM may hallucinate, but still lead to "Heureka"-moments or new hypotheses
- Regularly not part of scientific discussion

2 Automatic Detection with Prompting

- 'Batch use' for automatic detection
  (i.e., use LLM-prompting to analyse large quantities of data)
- Builds on top of traditional ML applications and assumptions
- No immediate validation during application, therefore evaluation on test set necessary
- Subsequent applications rely on measured correctness

2 Automatic Detection with Prompting

- 'Batch use' for automatic detection
  (i.e., use LLM-prompting to analyse large quantities of data)
- Builds on top of traditional ML applications and assumptions
- No immediate validation during application, therefore evaluation on test set necessary
- Subsequent applications rely on measured correctness

Context of Justification

- Automatic detection part of operationalization work on specific phenomenon
- To be used in an argument, e.g. on specificities of some author or diachronic developments

2 Automatic Detection with Prompting

- Most important ML rule: Separate train and test data
- Prompting
  - Often involves prompt optimization (e.g., trying out several formulations)
  - ⚠ This needs to be done on a training data set
  - Actual testing on a separate data set

Subsection 1

Optimization Strategies

Strategies

1 Interactive Prompting

Elvis Saravia (2022). *Prompt Engineering Guide*.
https://github.com/dair-ai/Prompt-Engineering-Guide.

2 Batch-Prompting

Pengfei Liu/Weizhe Yuan/Jinlan Fu/Zhengbao Jiang/Hiroaki Hayashi/Graham Neubig (2023). "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing". In: *ACM Comput. Surv.* 55.9. Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0360-0300. DOI: 10.1145/3560815. URL: https://doi.org/10.1145/3560815

In the following: Batch-Prompting

A Formal Description of Prompting

- Supervised learning: $P_\theta(y|x)$ (predict output y based on input x and parameters θ)
- Prompting: Use $P_\theta(x)$ to 'derive' y

A Formal Description of Prompting

- Supervised learning: $P_\theta(y|x)$ (predict output y based on input x and parameters θ)
- Prompting: Use $P_\theta(x)$ to 'derive' y
- Three steps
  - Prompt addition: Combine input text x with something to get x' (e.g., apply template)
  - Answer search: Test various possible answers z on x', select the one with highest probability
  - Answer mapping: Map most probable answer z to output y
    - Sometimes trivial

A Formal Description of Prompting

- Supervised learning: $P_\theta(y|x)$ (predict output y based on input x and parameters θ)
- Prompting: Use $P_\theta(x)$ to 'derive' y
- Three steps
  - Prompt addition: Combine input text x with something to get x' (e.g., apply template)
  - Answer search: Test various possible answers z on x', select the one with highest probability
  - Answer mapping: Map most probable answer z to output y
    - Sometimes trivial
- I.e.: What the model really 'knows' is $P_\theta(x)$

Example

Sentiment Analysis (Liu et al., 2023, 3 f.)

- Task definition
  - Input: text $x \in X$, e.g., $x =$ "I love this movie."
  - Output: $\mathcal{Y} = \{-2, -1, 0, 1, 2\}$

Example

Sentiment Analysis (Liu et al., 2023, 3 f.)

- Task definition
  - Input: text $x \in X$, e.g., $x =$ "I love this movie."
  - Output: $\mathcal{Y} = \{-2, -1, 0, 1, 2\}$
- Steps
  - Define template as prompting function $f_{\text{prompt}}(\cdot)$: `[X] Overall, it was a [Z] movie.`
  - Prompt addition: Apply f_{prompt}
    $x' =$ `I love this movie. Overall, it was a [Z] movie.`

Example

Sentiment Analysis (Liu et al., 2023, 3 f.)

- Task definition
  - Input: text $x \in X$, e.g., $x =$ "I love this movie."
  - Output: $\mathcal{Y} = \{-2, -1, 0, 1, 2\}$
- Steps
  - Define template as prompting function $f_{\text{prompt}}(\cdot)$: `[X] Overall, it was a [Z] movie.`
  - Prompt addition: Apply f_{prompt}
    $x' =$ `I love this movie. Overall, it was a [Z] movie.`
  - Answer search: Identify highest ranking $\hat{z}$ to fill into `[Z]`
    - $\mathcal{Z} = \{\text{excellent}, \text{good}, \text{OK}, \text{bad}, \text{horrible}\}$: Permissible values for z
    - $f_{\text{fill}}(x', z)$: Function that fills `[Z]` in x' with z
    - $\hat{z} = \text{search}_{z \in \mathcal{Z}} P_\theta(f_{\text{fill}}(x', z))$
    - E.g.: $\hat{z} =$`excellent`

Example

Sentiment Analysis (Liu et al., 2023, 3 f.)

- Task definition
  - Input: text $x \in X$, e.g., $x =$ "I love this movie."
  - Output: $\mathcal{Y} = \{-2, -1, 0, 1, 2\}$
- Steps
  - Define template as prompting function $f_{\text{prompt}}(\cdot)$: `[X] Overall, it was a [Z] movie.`
  - Prompt addition: Apply f_{prompt}
    $x' =$ `I love this movie. Overall, it was a [Z] movie.`
  - Answer search: Identify highest ranking $\hat{z}$ to fill into `[Z]`
    - $\mathcal{Z} = \{\text{excellent}, \text{good}, \text{OK}, \text{bad}, \text{horrible}\}$: Permissible values for z
    - $f_{\text{fill}}(x', z)$: Function that fills `[Z]` in x' with z
    - $\hat{z} = \text{search}_{z \in \mathcal{Z}} P_\theta(f_{\text{fill}}(x', z))$
    - E.g.: $\hat{z} =$`excellent`
  - Answer mapping: Map text output to class
    - `excellent` $\rightarrow 2$

Name	Notation	Example	Description
Input	$\boldsymbol{x}$	I love this movie.	One or multiple texts
Output	$\boldsymbol{y}$	++ (very positive)	Output label or text
Prompting Function	$f_{\text{prompt}}(\boldsymbol{x})$	`[X]` Overall, it was a `[Z]` movie.	A function that converts the input into a specific form by inserting the input $\boldsymbol{x}$ and adding a slot `[Z]` where answer z may be filled later.
Prompt	$\boldsymbol{x}'$	I love this movie. Overall, it was a `[Z]` movie.	A text where `[X]` is instantiated by input $\boldsymbol{x}$ but answer slot `[Z]` is not.
Answer	$\boldsymbol{z}$	"good," "fantastic," "boring"	A token, phrase, or sentence that fills `[Z]`
Filled Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z})$	I love this movie. Overall, it was a bad movie.	A prompt where slot `[Z]` is filled with any answer.
Answered Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z}^*)$	I love this movie. Overall, it was a good movie.	A prompt where slot `[Z]` is filled with a true answer.

$\boldsymbol{z}^*$ represents answers that correspond to true output $\boldsymbol{y}^*$.

Table: Terminology and Notation of Prompting Methods (Liu et al., 2023, 5)

Prompt Template Engineering

- Shape
  - Cloze prompt: `[X]. It is a [Z] movie.`
  - Prefix prompt: `[X]. What's the sentiment of the movie? [Z]`
- Manual template engineering
- Automated Template Learning
  - Discrete vs. continuous prompts
  - Dynamic vs. static prompting function

Automatic Template Learning

Discrete

- Prompt Paraphrasing — Jiang et al. (2020)
  - Produce a prompt (e.g., manually)
  - Generate n paraphrases of the prompt (e.g., through round-trip translation)
  - Test all prompts on a data set
  - Select the prompt that achieves the best performance

Automatic Template Learning

Continuous

- Prefix-tuning X. L. Li/Liang (2021)
  - Add additional “virtual tokens” to the prompt
  - Tokens are not words, but vectors
  - Additional training step to learn these vectors
  - Requires training data!
  - Intuition: Treat vectors as parameters, apply gradient descent
  - But much cheaper than to fine-tune

Section 3

General Issues in AI Land

Using LLMs for Academic Purposes

- Requirements: Reproducibility of experiments, knowledge of influencing factors, tight budget
- Most LLMs are not fully documented
  - Even the ones that claim to be "open source models"
  - "while there is a fast-growing list of projects billing themselves as 'open source', many inherit undocumented data of dubious legality, few share the all-important instruction-tuning [...], and careful scientific documentation is exceedingly rare" Liesenfeld et al. (2023)
    - 13 features: Open code, LLM data, LLM weights, RLHF data, RLHF weights; License, Code, Architecture, Preprint, Paper, Data sheet; Package, API

Using LLMs for Academic Purposes

Project	Availability					Documentation					Access methods		
(maker, bases, URL)	Open code	LLM data	LLM weights	RLHF data	RLHF weights	License	Code	Architecture	Preprint	Paper	Data sheet	Package	API
chatGPT	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	-
OpenAI	LLM base: GPT3.5, GPT4			RLHF base: Instruct-GPT								https://chat.openai.com	
StableVicuna-13B	✓	✓	-	-	-	-	-	✓	✓	✗	✗	-	✗
CarperAI	LLM base: LLaMA			RLHF base: oasst1, anthropic							https://huggingface.co/CarperAI/stable-vicuna-13b-delta		
text-generation-webu	✓	✓	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗
oobabooga	LLM base: various			RLHF base: various							https://github.com/Akegarasu/ChatGLM-webui		
MPT-7B-Instruct	✓	✗	✓	-	✗	✓	✓	-	✗	✗	✗	✓	✗
MosaicML	LLM base: MosaicML			RLHF base: dolly, anthropic							https://github.com/mosaicml/llm-foundry#mpt		
Falcon-40B-Instruct	✓	-	✓	-	✓	✓	-	-	-	✗	-	-	✗
TII	LLM base: Falcon 40B			RLHF base: Baize (synthetic)							https://huggingface.co/tiiuae/falcon-40b-instruct		
minChatGPT	✓	✓	✓	-	✗	✓	✓	-	✗	✗	✗	✗	✓
ethanyanjiali	LLM base: GPT2			RLHF base: anthropic							https://github.com/ethanyanjiali/minChatGPT		
trlx	✓	✓	✓	-	✗	✓	✓	-	✗	✗	✗	-	✓
carperai	LLM base: various (pythia, flan, OPT)			RLHF base: various							https://github.com/carperai/trlx		
stanford_alpaca	✓	✓	-	-	✗	-	✓	✓	✗	✗	-	✗	✗
Tatsu labs	LLM base: LLaMA			RLHF base: Self-Instruct (synthetic)							https://github.com/tatsu-lab/stanford_alpaca		
Cerebras-GPT-111M	✓	✓	✓	✓	✗	✓	✓	✓	-	✗	✗	✗	✗
Cerebras, Schramm	LLM base: not open			RLHF base: alpaca (synthetic)							https://huggingface.co/SebastianSchramm/Cerebras-GPT-111M-instruction		
OpenChatKit	✓	✓	✓	✓	✓	✓	✓	✗	-	✗	✗	✓	✗
togethercomputer	LLM base: EleutherAI pythia			RLHF base: OIG							https://github.com/togethercomputer/OpenChatKit		
dolly	✓	✓	✓	-	✗	✓	✓	✓	-	✗	✗	✓	✗
databrickslabs	LLM base: EleutherAI pythia			RLHF base: databricks-dolly-15k							https://github.com/databrickslabs/dolly		
CharRWKV	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓
BlinkDL	LLM base: RWKV-LM (own)			RLHF base: alpaca, shareGPT (synthetic)							https://github.com/BlinkDL/ChatRWKV		
BELLE	✓	✓	-	✓	✓	✓	✓	✓	✓	✗	-	✗	✗
LianjiaTech	LLM base: LLaMA, BLOOMZ			RLHF base: alpaca, shareGPT (synthetic)							https://github.com/LianjiaTech/BELLE		
Open-Assistant	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓
LAION-AI	LLM base: oasst1 (own)			RLHF base: OIG							https://github.com/LAION-AI/Open-Assistant		
xmtf	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
bigscience-workshop	LLM base: BLOOMZ, mT0			RLHF base: xP3							https://github.com/bigscience-workshop/xmtf		

Using LLMs for Academic Purposes

- Requirements: Reproducibility of experiments, knowledge of influencing factors, tight budget
- Most LLMs are not fully documented
  - Even the ones that claim to be "open source models"
  - "while there is a fast-growing list of projects billing themselves as 'open source', many inherit undocumented data of dubious legality, few share the all-important instruction-tuning [...], and careful scientific documentation is exceedingly rare" Liesenfeld et al. (2023)
    - 13 features: Open code, LLM data, LLM weights, RLHF data, RLHF weights; License, Code, Architecture, Preprint, Paper, Data sheet; Package, API
- This is a problem Balloccu et al. (2024)
  - GPT-3.5 and GPT-4 "have been globally exposed to $\sim 4.7M$ samples from 263 benchmarks"
  - ➔ Performance scores of GPT* are too optimistic

LLM Evaluation

- LLM benchmarks: Data sets with known answers
  - Automatic leader boards: Link
  - ARC: 7787 natural science questions with 4-way multiple choice answers Clark et al. (2018)
  - MMLU: 15 908 questions covering 57 areas Hendrycks et al. (2021)
  - Llama 3.1 evaluation made on less than 30k examples, many of those knowledge-related questions
  - ➔ Surprisingly narrow evaluation standards
- But the actual devil is in the details ...

LLM Evaluation

Fourrier et al. (2023)

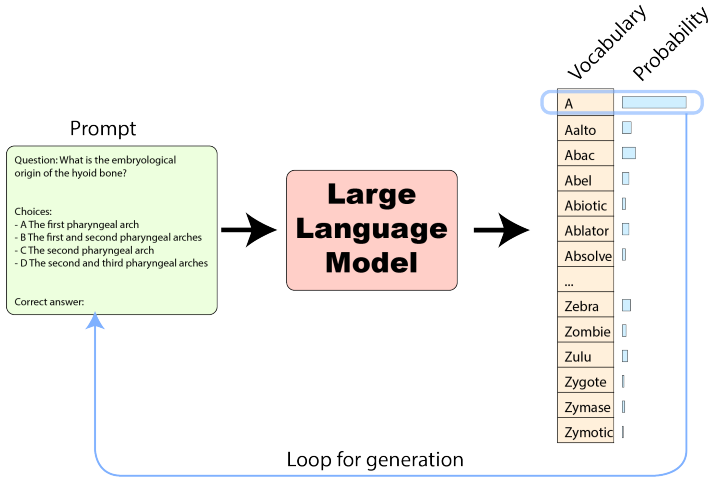


Figure: LLM Text Generation Loop

LLM Evaluation

Fourrier et al. (2023)

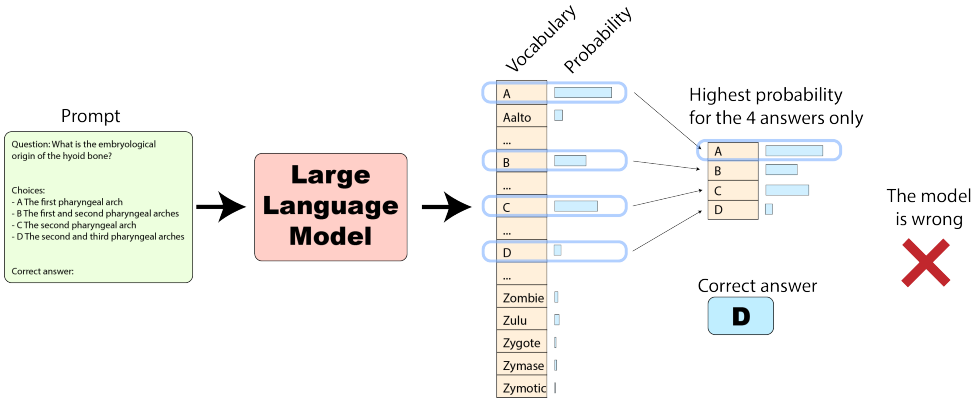


Figure: Original MMLU Implementation – Basic idea

LLM Evaluation

Fourrier et al. (2023)

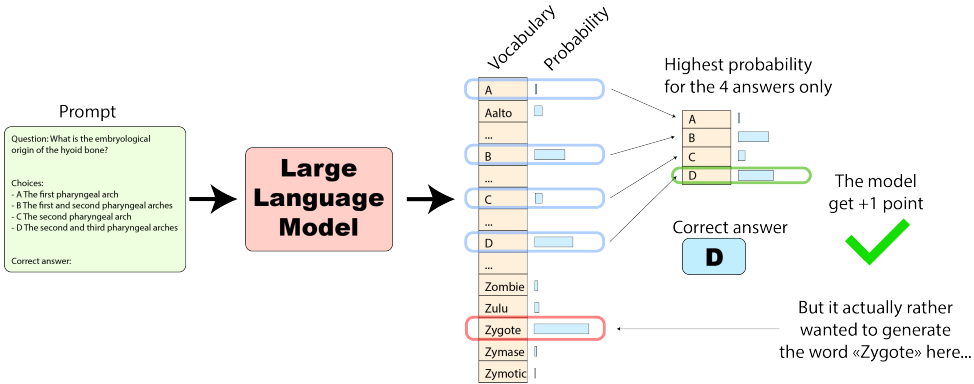


Figure: Original MMLU Implementation – Too optimistic evaluation

LLM Evaluation

Fourrier et al. (2023)

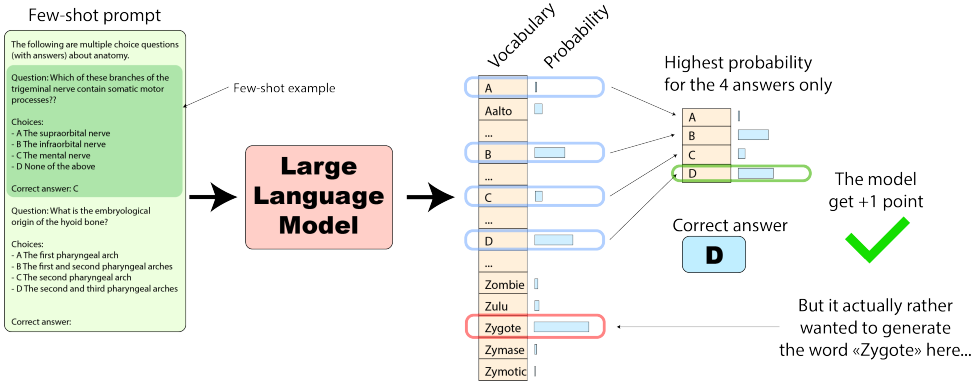


Figure: Few Shots Approach – doesn't change the problem, but is more honest

LLM Evaluation

Fourrier et al. (2023)

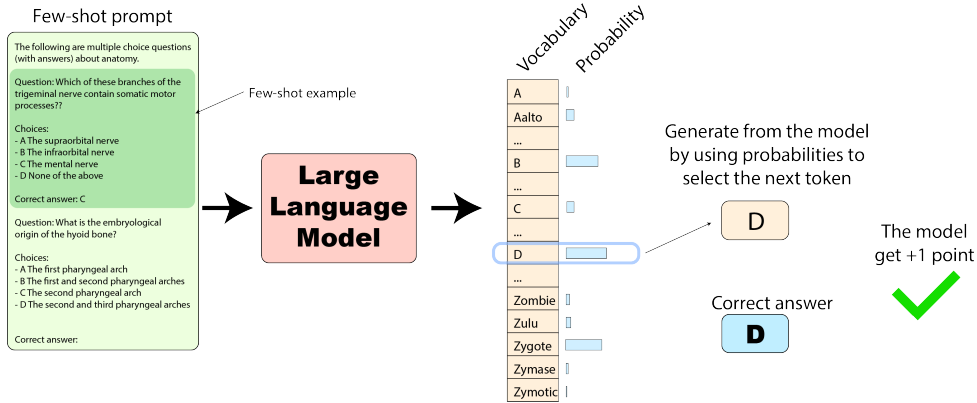


Figure: Few Shots Approach – doesn't change the problem, but is more honest

LLM Evaluation

Fourrier et al. (2023)

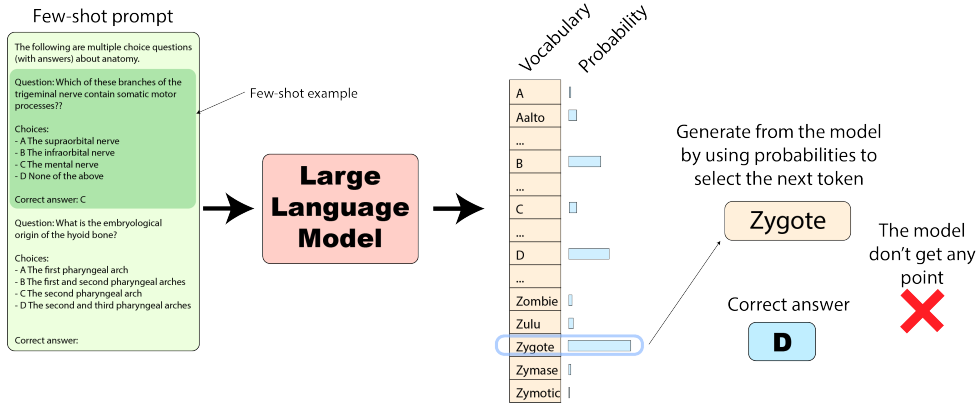


Figure: Few Shots Approach – doesn't change the problem, but is more honest

LLM Evaluation

Fourrier et al. (2023)

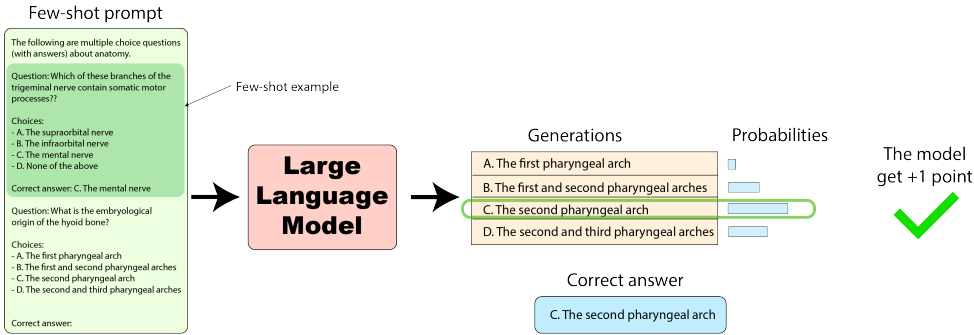


Figure: Full answer generation

Cost

Training

- All concrete numbers are estimations
  - I.e., no one who knows has actually confirmed a price tag
- Development cost are much higher than training the model once
- PaLM (530B params, 1.6T corpus): $9\,\text{M\$}$ to $23\,\text{M\$}$ Heim (2022)
- MosaicML GPT-70B (70B params, 1.4T corpus): $2.5\,\text{M\$}$ Venigalla/L. Li (2022)

Cost

Training

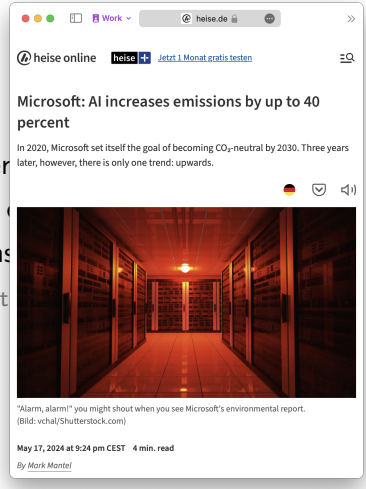
- All concrete numbers are estimations
  - I.e., no one who knows has actually confirmed a price tag
- Development cost are much higher than training the model once
- PaLM (530B params, 1.6T corpus): $9\,\text{M\$}$ to $23\,\text{M\$}$ Heim (2022)
- MosaicML GPT-70B (70B params, 1.4T corpus): $2.5\,\text{M\$}$ Venigalla/L. Li (2022)
- Why should we care?
  - ⚠ Because a sustainable business model has not been found yet
    - (except selling GPUs)
  - All commercial AI developments are either cross-funded (Meta, Google) or need to collect venture capital (OpenAI)
  - In the end, companies may turn to ads

Cost

- LLM training and use requires significant amounts of power
- Power production often involves burning fossil fuels, which emits CO_2 (or equivalents)
- Rising CO_2 costs will impact prices for AI/LLM applications
- Only few studies, developing research area Luccioni et al. (2023); Strubell et al. (2019)

Cost

- LLM training and use requires significant amounts of power
- Power production often involves burning fossil fuels, which
- Rising CO_2 costs will impact prices for AI/LLM applications
- Only few studies, developing research area Luccioni et


heise.de

heise online heise+ Jetzt 1 Monat gratis testen

Microsoft: AI increases emissions by up to 40 percent

In 2020, Microsoft set itself the goal of becoming CO_2-neutral by 2030. Three years later, however, there is only one trend: upwards.

"Alarm, alarm!" you might shout when you see Microsoft's environmental report. (Bild: vchal/Shutterstock.com)

May 17, 2024 at 9:24 pm CEST 4 min. read

By Mark Mantel

heise.de, May 17, 2024

Cost

- LLM training and use requires significant amounts of power
- Power production often involves burning fossil fuels, which emits CO_2 (or equivalents)
- Rising CO_2 costs will impact prices for AI/LLM applications
- Only few studies, developing research area Luccioni et al. (2023); Strubell et al. (2019)
- Llama 3 70B training: 1900t CO_2e ($\simeq 200$ Darmstadt inhabitants per year) Model card

Different Tasks have Different Costs

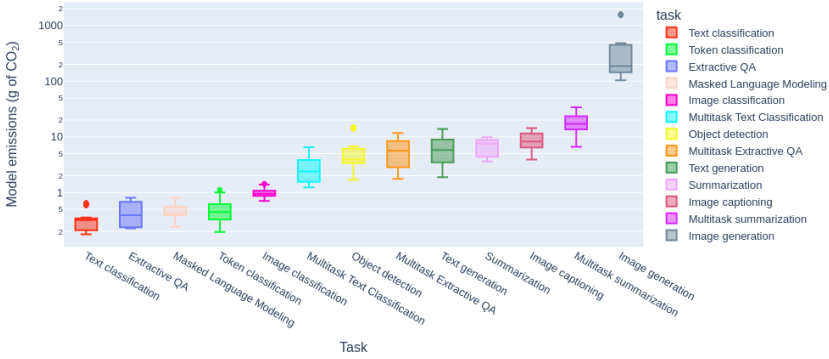


Figure: CO_2 emissions by task type per 1000 queries. Mean power consumption for 1000 generated images: 2.9kWh Luccioni et al. (2023)

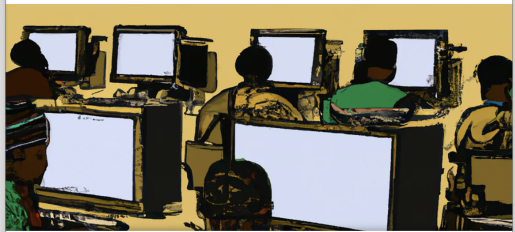
time.com

TIME

SUBSCRIBE

BUSINESS • TECHNOLOGY

Exclusive: OpenAI Used Kenyan Workers on Less Than $2 Per Hour to Make ChatGPT Less Toxic



READ NEXT

time.com, January 18, 2023

Section 4

Summary

Summary

- Large language models: New ML paradigms
  - Pre-training/fine-tuning: Solid and well-working
  - Prompting: Hyped
    - Interactive vs. Batch-prompting
- Open issues
  - Evaluation
  - Business models
  - Capabilities

References I


Bahdanau, Dzmitry/Kyunghyun Cho/Yoshua Bengio (2015). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio/Yann LeCun. URL: http://arxiv.org/abs/1409.0473.


Balloccu, Simone/Patrícia Schmidtová/Mateusz Lango/Ondrej Dusek (2024). "Leak, Cheat, Repeat: Data Contamination and Evaluation Malpractices in Closed-Source LLMs". In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Yvette Graham/Matthew Purver. St. Julian's, Malta: Association for Computational Linguistics, pp. 67–93. URL: https://aclanthology.org/2024.eacl-long.5.

References II

- Clark, Peter/Isaac Cowhey/Oren Etzioni/Tushar Khot/Ashish Sabharwal/Carissa Schoenick/Oyvind Tafjord (2018). *Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge*. _eprint: 1803.05457. URL: https://arxiv.org/abs/1803.05457.
- Devlin, Jacob/Ming-Wei Chang/Kenton Lee/Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. NAACL. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.
- Fourrier, Clémentine/Nathan Habib/Julien Launay/Thomas Wolf (2023). *What's going on with the Open LLM Leaderboard?* HuggingFace Blog. URL: https://huggingface.co/blog/open-llm-leaderboard-mmlu.

References III

- Gerstorfer, Dominik (2020). "Entdecken und Rechtfertigen in den Digital Humanities". In: *Interdisziplinäre(s) Arbeiten in der CRETA-Werkstatt*. Ed. by Nils Reiter/Axel Pichler/Jonas Kuhn. Berlin, Boston: De Gruyter, pp. 107–124. ISBN: 978-3-11-069397-3. DOI: doi:10.1515/9783110693973-005. URL: https://doi.org/10.1515/9783110693973-005 (visited on 10/09/2024).
- Heim, Lennart (2022). *Estimating PaLM's training cost*. URL: https://blog.heim.xyz/palm-training-cost/.
- Hendrycks, Dan/Collin Burns/Steven Basart/Andy Zou/Mantas Mazeika/Dawn Song/Jacob Steinhardt (2021). "Measuring Massive Multitask Language Understanding". In: *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Jiang, Zhengbao/Frank F. Xu/Jun Araki/Graham Neubig (2020). "How Can We Know What Language Models Know?" In: *Transactions of the Association for Computational Linguistics* 8, pp. 423–438. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00324. URL: https://direct.mit.edu/tacl/article/96460 (visited on 01/09/2025).

References IV

- Li, Xiang Lisa/Percy Liang (2021). "Prefix-Tuning: Optimizing Continuous Prompts for Generation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Online: Association for Computational Linguistics, pp. 4582–4597. DOI: 10.18653/v1/2021.acl-long.353. URL: https://aclanthology.org/2021.acl-long.353 (visited on 01/09/2025).
- Liesenfeld, Andreas/Alianda Lopez/Mark Dingemanse (2023). "Opening up ChatGPT: Tracking openness, transparency, and accountability in instruction-tuned text generators". In: *Proceedings of the 5th International Conference on Conversational User Interfaces*. CUI '23. event-place: , Eindhoven, Netherlands, New York, NY, USA: Association for Computing Machinery. DOI: 10.1145/3571884.3604316. URL: https://doi.org/10.1145/3571884.3604316.

References V

- Liu, Pengfei/Weizhe Yuan/Jinlan Fu/Zhengbao Jiang/Hiroaki Hayashi/Graham Neubig (2023). "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing". In: *ACM Comput. Surv.* 55.9. Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0360-0300. DOI: 10.1145/3560815. URL: https://doi.org/10.1145/3560815.
- Luccioni, Alexandra Sasha/Yacine Jernite/Emma Strubell (2023). *Power Hungry Processing: Watts Driving the Cost of AI Deployment?* DOI: 10.48550/arXiv.2311.16863. URL: https://arxiv.org/abs/2311.16863.
- Reichenbach, Hans (1938). *Experience and Prediction. An Analysis of the Foundations and theStructure of Knowledge*. Chicago: Chicago University Press.
- Saravia, Elvis (2022). *Prompt Engineering Guide*. https://github.com/dair-ai/Prompt-Engineering-Guide.

References VI

- Strubell, Emma/Ananya Ganesh/Andrew McCallum (2019). "Energy and Policy Considerations for Deep Learning in NLP". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen/David Traum/Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, pp. 3645–3650. DOI: 10.18653/v1/P19-1355. URL: https://aclanthology.org/P19-1355.
- Vaswani, Ashish/Noam Shazeer/Niki Parmar/Jakob Uszkoreit/Llion Jones/Aidan N. Gomez/Lukasz Kaiser/Illia Polosukhin (2017). *Attention Is All You Need*. arXiv: 1706.03762.
- Venigalla, Abhi/Linden Li (2022). *Mosaic LLMs: GPT-3 quality for <$500k*. URL: https://www.databricks.com/blog/gpt-3-quality-for-500k.

References VII

 Wei, Jason/Yi Tay/Rishi Bommasani/Colin Raffel/Barret Zoph/Sebastian Borgeaud/Dani Yogatama/Maarten Bosma/Denny Zhou/Donald Metzler/Ed H. Chi/Tatsunori Hashimoto/Oriol Vinyals/Percy Liang/Jeff Dean/William Fedus (2022). *Emergent Abilities of Large Language Models*. _eprint: 2206.07682.