Multilingual NLP

11. Large Language Models & Instruction Tuning (+ Reinforcement Learning from Human Feedback)

Prof. Dr. Goran Glavaš
Center for AI and Data Science (CAIDAS), Uni Würzburg

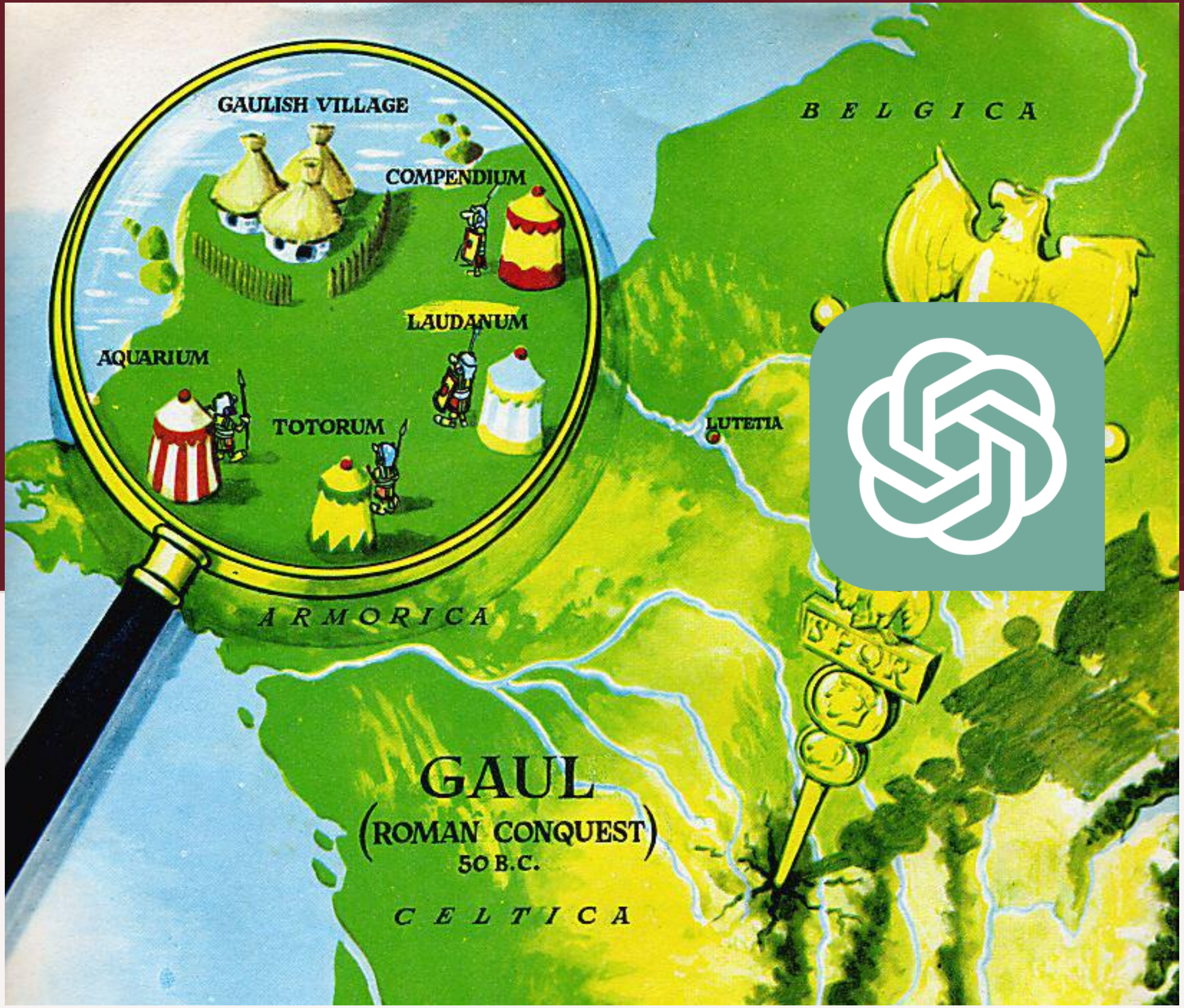
Image: Alexander Mikhalchyk

After this lecture, you'll...

- Understand what Large Language Models (LLMs) are
- Learn to formulate any NLP task as text generation
- Know what in-context learning is
- Understand what instruction tuning is
- Know how multilingual LLMs are

Content

- **Large Language Models**
- Generative NLP
- Learning from Human Preferences
- Final Thoughts



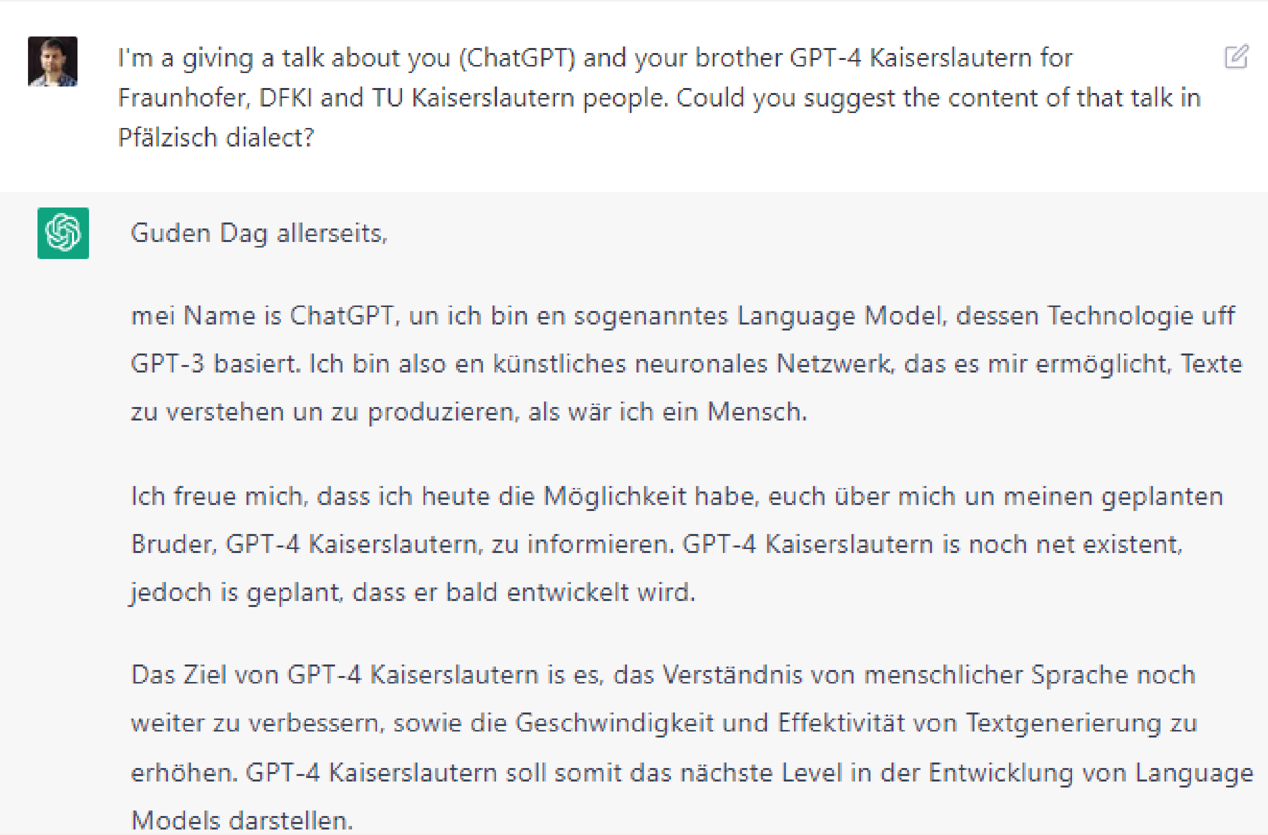
It is November 30th, 2022...

...and there's a new sheriff in town!

How did we get here?

- The short answer: **scale**
  - By exponentially scaling data and compute up

- But let us learn the whole story…



I'm a giving a talk about you (ChatGPT) and your brother GPT-4 Kaiserslautern for Fraunhofer, DFKI and TU Kaiserslautern people. Could you suggest the content of that talk in Pfälzisch dialect?

Guden Dag allerseits,

mei Name is ChatGPT, un ich bin en sogenanntes Language Model, dessen Technologie uff GPT-3 basiert. Ich bin also en künstliches neuronales Netzwerk, das es mir ermöglicht, Texte zu verstehen un zu produzieren, als wär ich ein Mensch.

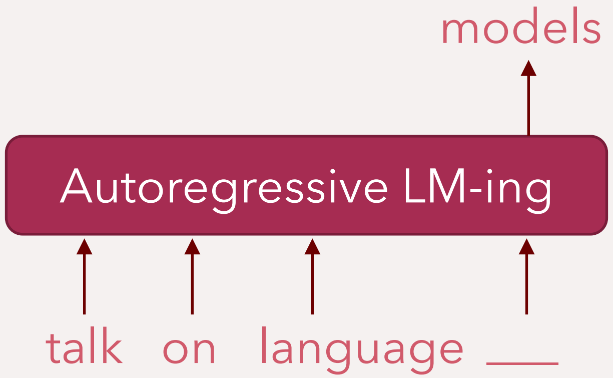
Ich freue mich, dass ich heute die Möglichkeit habe, euch über mich un meinen geplanten Bruder, GPT-4 Kaiserslautern, zu informieren. GPT-4 Kaiserslautern is noch net existent, jedoch is geplant, dass er bald entwickelt wird.

Das Ziel von GPT-4 Kaiserslautern is es, das Verständnis von menschlicher Sprache noch weiter zu verbessern, sowie die Geschwindigkeit und Effektivität von Textgenerierung zu erhöhen. GPT-4 Kaiserslautern soll somit das nächste Level in der Entwicklung von Language Models darstellen.

How did we get here?

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

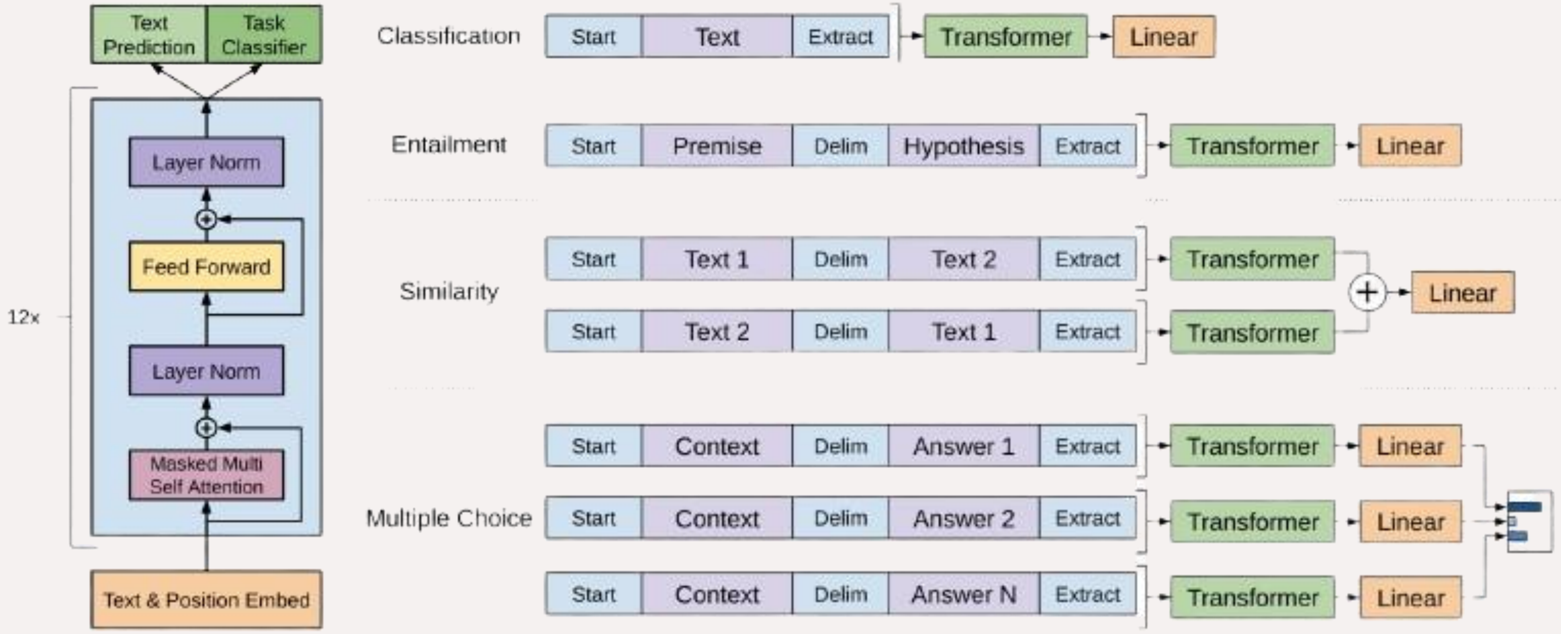
- **GPT** (aka GPT-1): 2018
  - Contemporary to BERT
  - **117 million** parameters
  - Trained on the BookCorpus (7000 unpublished books), ca. **1B tokens**
- Decoder-only model
  - Autoregressive LM pretraining
  - Task-specific fine-tuning
    - Q: How do you fine-tune a decoder LM (for different tasks)?



How did we get here?

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

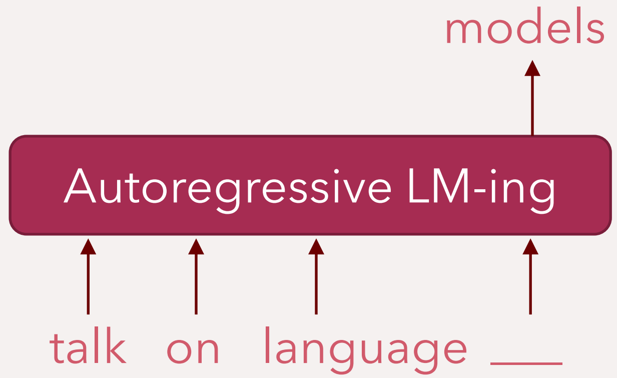
- Q: How do you fine-tune a decoder LM (for different tasks)?



How did we get here?

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

- **GPT-2**: 2019
  - **1.5 B** ($1.5*10^9$) parameters
  - Trained on the „WebText"
    - Ca. 40GB of text
    - Collected from outbound links from Reddit posts
  - Pretraining: still only autoreg. LM-ing
- **Zero-shot task performance!**
  - Can successfully perform a task without being fine-tuned for it



How did we get here?

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

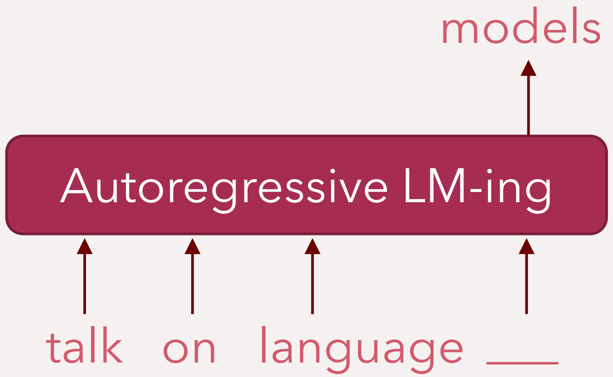
Zero-shot task performance!

- Can successfully perform a task without being fine-tuned for it
- But only if…
  - Examples of the task naturally occured in the pretraining corpus (e.g., translation or question answering)
  - We describe the task with a good prompt
- What is a „prompt"
  - Prompt = natural language text provided to the LM, describing the task that the LM needs to solve, i.e., what it needs to generate
    - Example: „Translate from English to French: [English text]"
  - Brittle: Performance of GPT-2 very prompt-dependent

How did we get here?

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

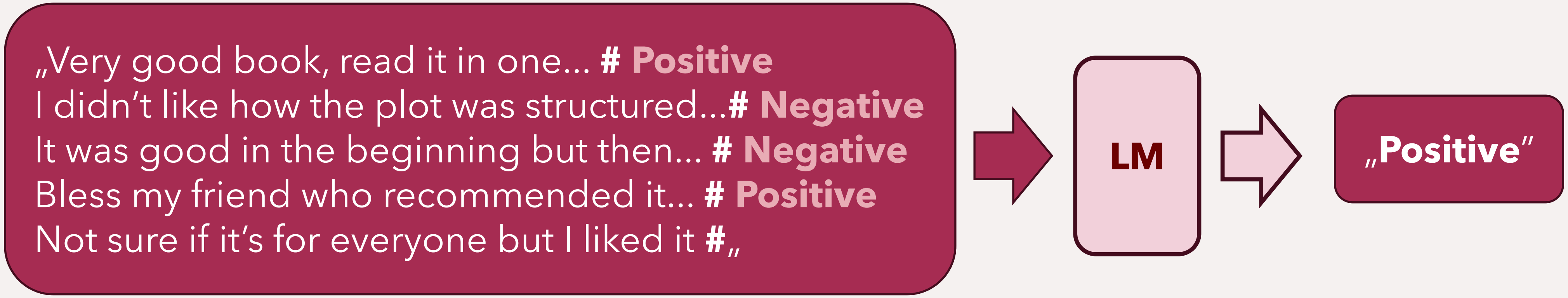
- **GPT-3**: 2020
  - **175 B** (1.75*10^{11}) parameters
  - Trained on 45TB of web text
  - Pretraining: still only autoreg. LM-ing
- **Few-Shot In-Context „Learning"**
  - Can successfully perform a task without being fine-tuned for it
  - But labeled examples provided „in the context", i.e, as part of the prompt



How did we get here?

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

- **Few-Shot In-Context „Learning"**
  - There is **no** learning in the machine learning sense (no parameter updates), LM is just doing inference starting with the prompt
  - Labeled task examples provided „in the context", as part of the prompt



„Very good book, read it in one... # **Positive**
I didn't like how the plot was structured...# **Negative**
It was good in the beginning but then... # **Negative**
Bless my friend who recommended it... # **Positive**
Not sure if it's for everyone but I liked it #„

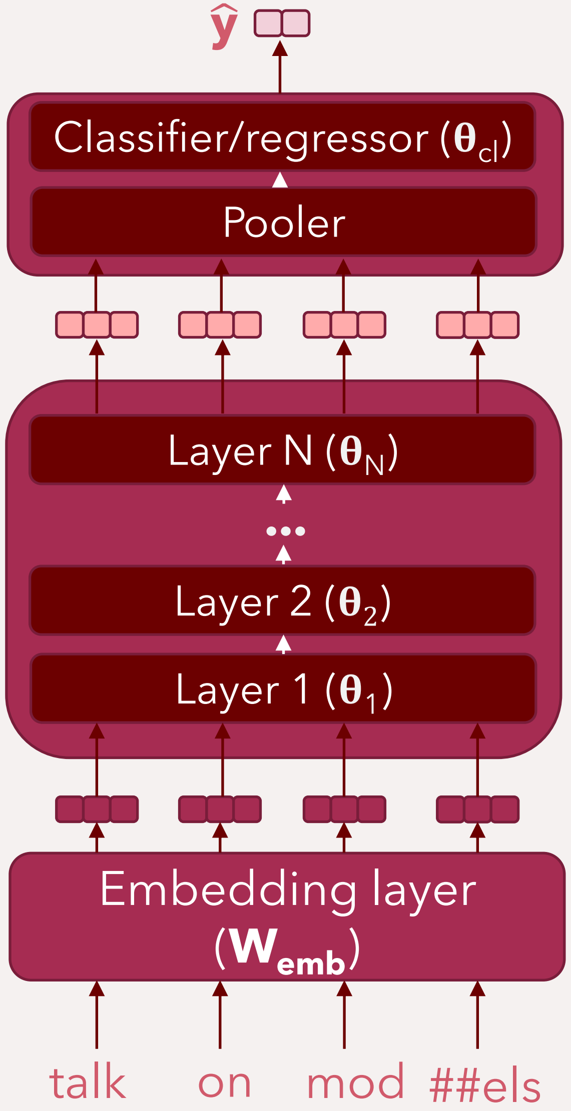
Content

- Large Language Models
- **Generative NLP**
- Learning from Human Preferences
- Final Thoughts

Recap: Uniform NLP with Encoders

- The vast majority of NLP tasks fall into one of three categories
  - Sequence classification
  - Token classsification
  - Text generation

- Q: What is still not uniform across tasks?
  - Task-specific classifiers/heads
    - Different from the pretraining classifier (LM Head)!
    - Impedes transfer learning


$\hat{\mathbf{y}}$
Classifier/regressor ($\boldsymbol{\theta}_{cl}$)
Pooler
Layer N ($\boldsymbol{\theta}_N$)
...
Layer 2 ($\boldsymbol{\theta}_2$)
Layer 1 ($\boldsymbol{\theta}_1$)
Embedding layer ($\mathbf{W}_{\mathbf{emb}}$)
talk
on
mod
##els

Generative NLP

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research, 21(140), 1-67.

- **Key Idea**: cast every NLP task as a text generation task!
  - I.e., we cast sequence and token classification/regression as text generation
    - Class/score labels in classification/regression tasks are tokens!

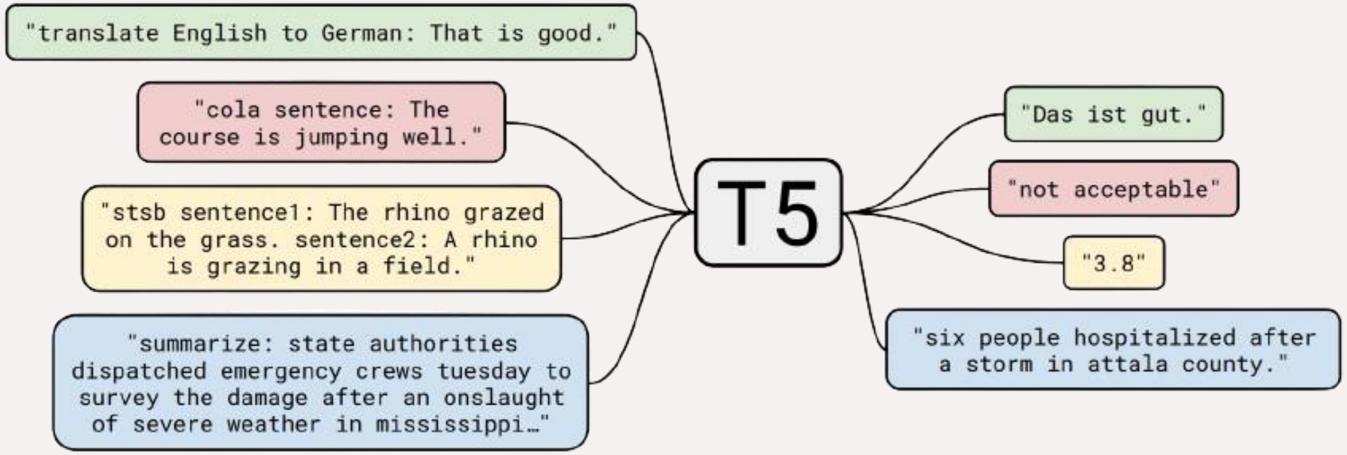
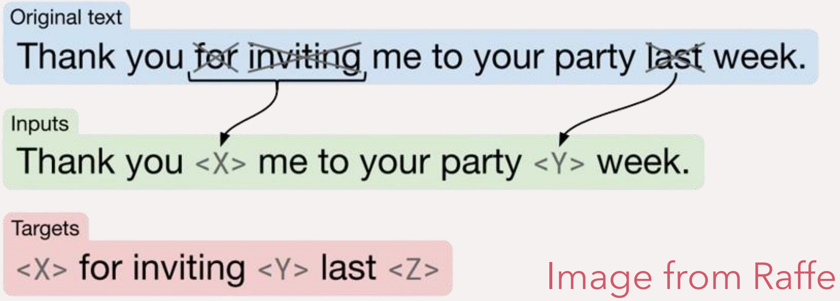

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Image from Raffel et al.

Generative NLP

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research, 21(140), 1-67.

- **T5** is an encoder-decoder model
  - Main pretraining objective: generating/predicting masked out spans
  - On a novel corpus:
    - Colossal Cleaned Common Crawl (C4)
  - Model sizes: from 60M (small) to 11B (XXL) parameters



Original text

Thank you ~~for inviting~~ me to your party ~~last~~ week.

Inputs

Thank you <X> me to your party <Y> week.

Targets

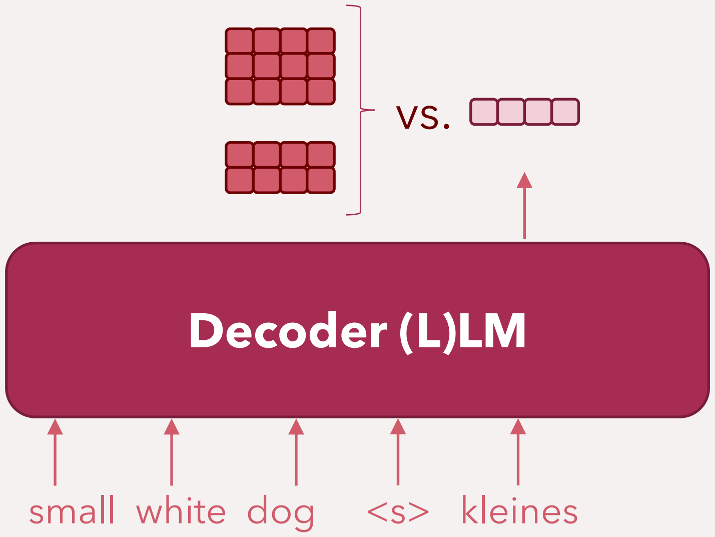
<X> for inviting <Y> last <Z>

Image from Raffel et al.

- Task-specific fine-tuning
  - Same for all tasks (LM objective), but carried out independently for each task
  - As such, no task descriptions!

Constrained Decoding in Generative NLP

- In open-ended text generation, LLM is generally allowed to generate any token from the vocabulary at every time step
  - Contextualized representation of the last input token $\mathbf{x}$ compared with output embeddings of all vocabulary tokens $\mathbf{v}_1$, $\mathbf{v}_2$, … $\mathbf{v}_{|V|}$
    - Typically just a dot-product, $\mathbf{x}^T\mathbf{v}_\mathbf{i}$
- When casting some classification task generatively
  - „Valid" tokens to be generated are just the class-specific tokens
    - E.g., „positive", „negative" for sentiment
  - Compare $\mathbf{x}$ only with output embeddings of tokens corresponding to task classes $\mathbf{v}_{C1}$, …, $\mathbf{v}_{Cn}$
  - Softmax over much shorter vectors of logits (of length Cn), much faster decoding



Generative NLP

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., ... & Raffel, C. (2021, June). mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In Proceedings of the NAACL 2021 (pp. 483-498).

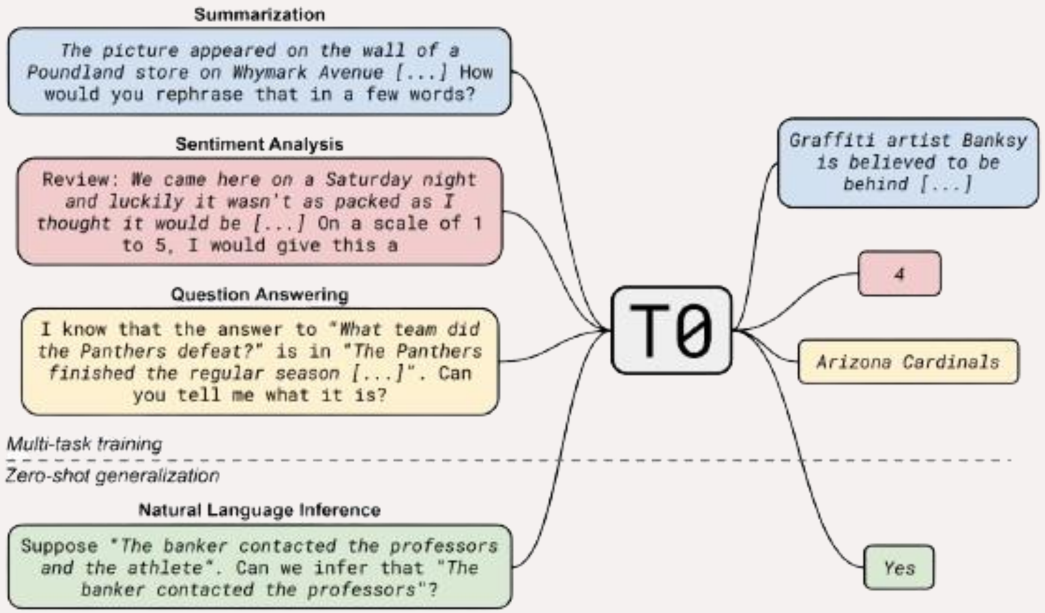
- **mT5: just a multilingual T5**
  - Trained on a large multilingual corpus mC4, encompassing 107 languages
  - Standard temperature over/under-sampling for low/high-resource languages

- Self-supervised multilingually pretrained encoder-decoder model
  - Q: Haven't we seen some already?
  - Yes, mBART!
  - mT5 pretrained with different objectives and on orders of magnitude more data
  - But mBART did not carry out task-specific fine-tuning generatively (except for inherently generative tasks like MT)

Generative NLP

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., ... & Rush, A. M. (2022, April). Multitask Prompted Training Enables Zero-Shot Task Generalization. ICLR 2022, Tenth International Conference on Learning Representations.

- Q: What happens if we fine-tune T5 simultaneously for many tasks?
- **T0** = Multi-task instruction-based generative fine-tuning of T5



Summarization

The picture appeared on the wall of a Poundland store on Whymark Avenue [...] How would you rephrase that in a few words?

Sentiment Analysis

Review: We came here on a Saturday night and luckily it wasn't as packed as I thought it would be [...] On a scale of 1 to 5, I would give this a

Question Answering

I know that the answer to "What team did the Panthers defeat?" is in "The Panthers finished the regular season [...]". Can you tell me what it is?

Multi-task training

Zero-shot generalization

Natural Language Inference

Suppose "The banker contacted the professors and the athlete". Can we infer that "The banker contacted the professors"?

T0

Graffiti artist Banksy is believed to be behind [...]

4

Arizona Cardinals

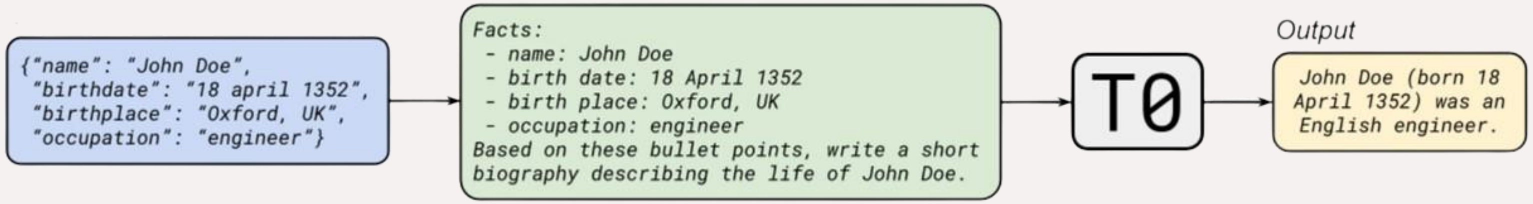
Yes

Image from Sanh et al.

Generative NLP: Instruction-Tuning

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., ... & Rush, A. M. (2022, April). Multitask Prompted Training Enables Zero-Shot Task Generalization. ICLR 2022, Tenth International Conference on Learning Representations.

- Q: What happens if we fine-tune T5 simultaneously for many tasks?
- **T0** = Multi-task and instruction-based generative fine-tuning of T5
- **Instruction-tuning**: any type of generative fine-tuning of LLMs that provides the description (explanation) of the task as part of the input prompt
- In T0, they convert 170 English NLP datasets into ca. 2000 different instruction-based prompts (example below for a data-to-text task)



```
{"name": "John Doe",
 "birthdate": "18 april 1352",
 "birthplace": "Oxford, UK",
 "occupation": "engineer"}
```

```
Facts:
 - name: John Doe
 - birth date: 18 April 1352
 - birth place: Oxford, UK
 - occupation: engineer
Based on these bullet points, write a short
biography describing the life of John Doe.
```

```
John Doe (born 18
April 1352) was an
English engineer.
```

Image from Sanh et al.

Generative NLP: Instruction-Tuning

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., ... & Rush, A. M. (2022, April). Multitask Prompted Training Enables Zero-Shot Task Generalization. ICLR 2022, Tenth International Conference on Learning Representations.

- **Multi-task instruction-tuning**: pushes the model to learn to generalize over tasks from their respective instructions
  - Instruction = NL description of the task

- **Zero-shot generalization** thanks to instructions:
  - Model should be able to „figure out" a new task from the task's instruction (i.e., natural language description)
  - T0 generalizes well to tasks similar to those in training
  - Limitation: many NLP benchmark tasks do not correspond to tasks that humans would use LLMs for

Generative NLP: Instruction-Tuning

Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Le Scao, T., ... & Raffel, C. (2023, July). Crosslingual Generalization through Multitask Finetuning. In Proceedings of Annual Meeting of Association for Computational Linguistic (pp. 15991-16111).

- **mT0** = massively multilingual variants of T0
  - Instruction-tuned mT5
  - Q: On which data (in which languages) to instruction-tune mT5?
    - Most labeled datasets are in English
    - T0 prompts also in English

- Possibilities for training data
  - English only (prompts and data) – zero-shot XLT possible due to mT5 (analogous to zero-shot XLT with encoder models like mBERT or XLM-R)
  - English prompts, multilingual data
    - Gold multilingual training data or obtained with MT (i.e., „translate-train")
  - Multilingual prompts with multilingual data
    - MT-translated from English prompts

Content

- Large Language Models
- Generative NLP
- **Learning from Human Preferences**
- Final Thoughts

Fine-Tuning from Human Preferences

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35, 27730-27744.

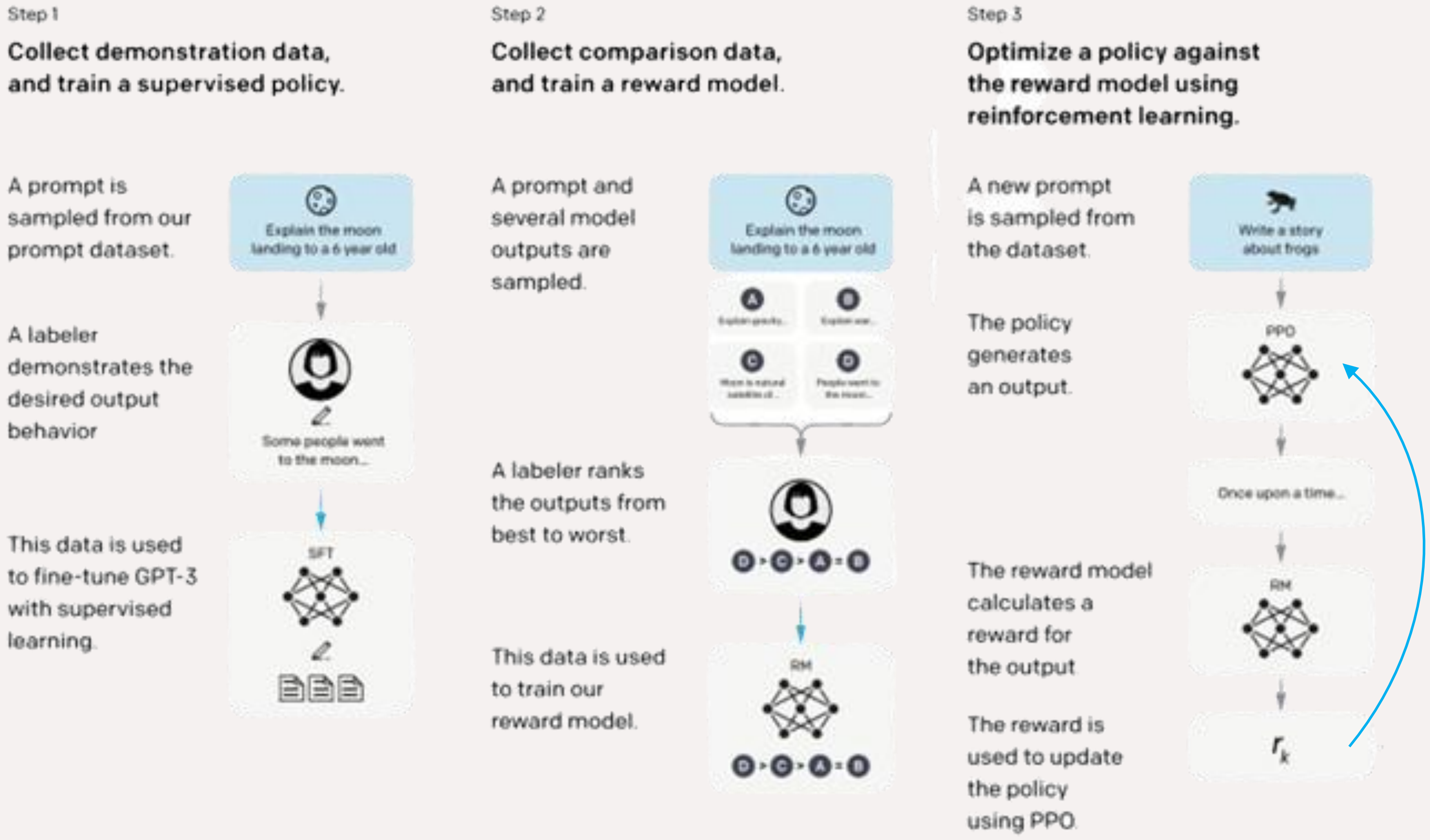
- **Instruct-GPT**: training LLMs to follow (arbitrary) human instructions

- Two training steps, starting fom GPT-3:

  1. Supervised instruction-tuning
     - Direct LM-training on human-labeled prompt/answer pairs
  2. Reinforcement learning from human feedback (RLHF)
     - Collect a dataset of human rankings of model outputs
     - Fine-tune on this preference data using RL
     - This requires a „reward model", which is trained in advance

Fine-Tuning from Human Preferences

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35, 27730-27744.



Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior

This data is used to fine-tune GPT-3 with supervised learning.

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output

The reward is used to update the policy using PPO

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Explain the moon landing to a 6 year old
A
B
Explain gravity...
Explain war...
C
D
Moon is natural satellite of...
People went to the moon...
D > C > A = B
RM
D > C > A = B
Write a story about frogs
PPO
Once upon a time...
RM
r_k

Image from Ouyang et al.

Fine-Tuning from Human Preferences

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35, 27730-27744.

- Step 1: Collect demonstration data and train a supervised „policy"
  - „Policy" is an RL term, basically denotes the „main model", i.e., our LLM

  - Start from human prompts submitted to GPT-3 API **+** some newly written human prompts
    - Include tasks like creative generation, QA, summarization, extraction, ...
    - 13K human prompts in total
    - Hired 40 human annotators to write answers to those prompts

  - Fine-tune GPT-3 on these 13K prompt-answer pairs
    - Q: Training objective?
    - Plain simple autoregressive LM-ing (loss on the answer tokens, given prompt)

Fine-Tuning from Human Preferences

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35, 27730-27744.

- Step 2: Collect comparison/preference data and train a reward model (RM)
  - RM is a regression model: takes the prompt-answer pair as input and outputs a scalar
  - RM is used as the „value function" in the RL algorithm (Step 3)
  - Creation of comparison dataset:
    - Ask the instruction-tuned GPT-3 (i.e., the "policy") to generate multiple answers
      - They collect between 4 and 9 responses for the same prompt
      - Q: How to generate different answers with the same LLM?
    - For each pair of answers to the same prompt, ask humans which one they prefer
      - Let y_w be the winning generation, and y_l the losing one
  - Fine-tune a 6B parameter GPT-3 on a type of contrastive loss
    - Let $r_\theta(y, x)$ be the score that RM (with params θ) produces for answer y given prompt x
    - $loss(x, y_w, y_l) = \log(\sigma(r_\theta(y_w, x) - r_\theta(y_l, x)))$
    - Q: Why sigmoid?

RL based on Policy Optimization

- **General RL framework**
  - An agent makes actions in an environment based on the state of the environment
  - Environment states have value, as captured by the value function – operationalized through reward
  - Agent's action changes the environment → new state
  - **Goal**: agent that maximizes the (sum of) reward(s) in the interaction with the environment

- **Policy-optimization-based RL**
  - Agent is called a policy and is commonly optimized with gradient-based methods; denoted with $\pi_\theta(a|s)$
  - We need to compute the estimate of the gradient w.r.t. to policy parameters
    - Q: Gradient of what? Which function?

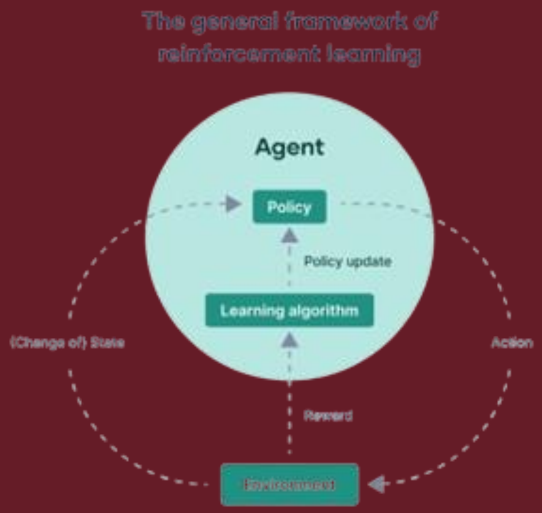


Image from https://www.scribbr.com/ai-tools/reinforcement-learning/

RL based on Policy Optimization

- **Policy-optimization-based RL**
  - Agent is called a policy and is commonly optimized with gradient-based methods
  - We need to estimate the gradient w.r.t. to policy parameters
  - Basic policy gradient estimation, for a batch of instances D – called trajectories $\tau = \{s_0, a_1, s_1, a_2, \ldots, a_T, s_T\}$ – is as follows:

$$\nabla'_\theta J(\pi_\theta) = \frac{1}{|D|} \sum_{\tau \in D} \left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)\, R(\tau) \right]$$

    - Where $R(\tau)$ is the „return", i.e., in the simplest form just sum of rewards from the individual time-steps

$$R(\tau) = \sum_{t=0}^{T} r(s_t)$$

      where $r(s)$ is the reward score for the state s_t
    - Policy's params finally updated with gradient ascent:

$$\theta^{(k+1)} = \theta^{(k+1)} + \eta \nabla'_\theta J(\pi_\theta)$$

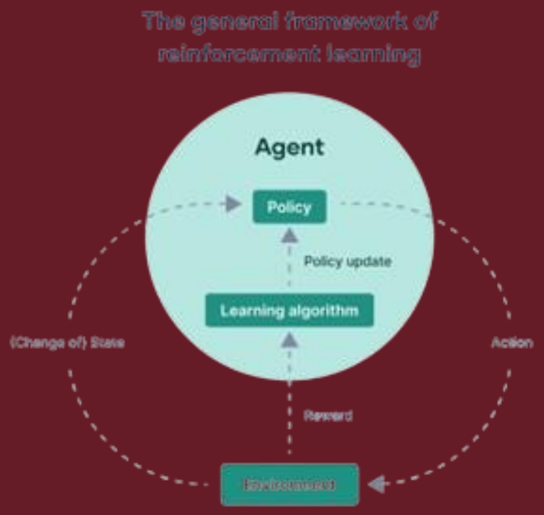


Image from https://www.scribbr.com/ai-tools/reinforcement-learning/

Fine-Tuning from Human Preferences

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35, 27730-27744.

- Step 3: Reinforcement learning, using the RM as the value function
  - Concretely, a gradient-optimization based policy-oriented RL algorithm called proximal policy optimization (PPO)
    - More advanced than the basic policy gradient estimation
    - Makes sure that the policy does not change too much with the updates
  - Q: But how exactly is a neural LM an RL „policy”? Q: Where are the states and actions?
    - Autoregressive LMs generate text one token at a time (i.e., time step)
    - Action: next word (i.e., which word to generate?)
    - State: preceding text; next state: preceding text + the generated token
  - Training:

    (1) RM from Step 2 provides the reward for the generated text, using which the policy gradient estimation on the „maximize reward objective” is computed
    (2) PPO computes updates to policy parameters so that it doesn’t change too much

ChatGPT & GPT-4

- The holy **scale**…
    - ChatGPT is effectively a **larger-scale** Instruct-GPT

    - Starts from a larger vanilla LLM: „GPT-3.5"

    - Performs RLHF on a much much larger scale
    - Most of effort and money went into human labeling
        - Many many more human prompts, covering a wider variety of tasks
        - Many more preference annotations, leading to larger-scale RLHF

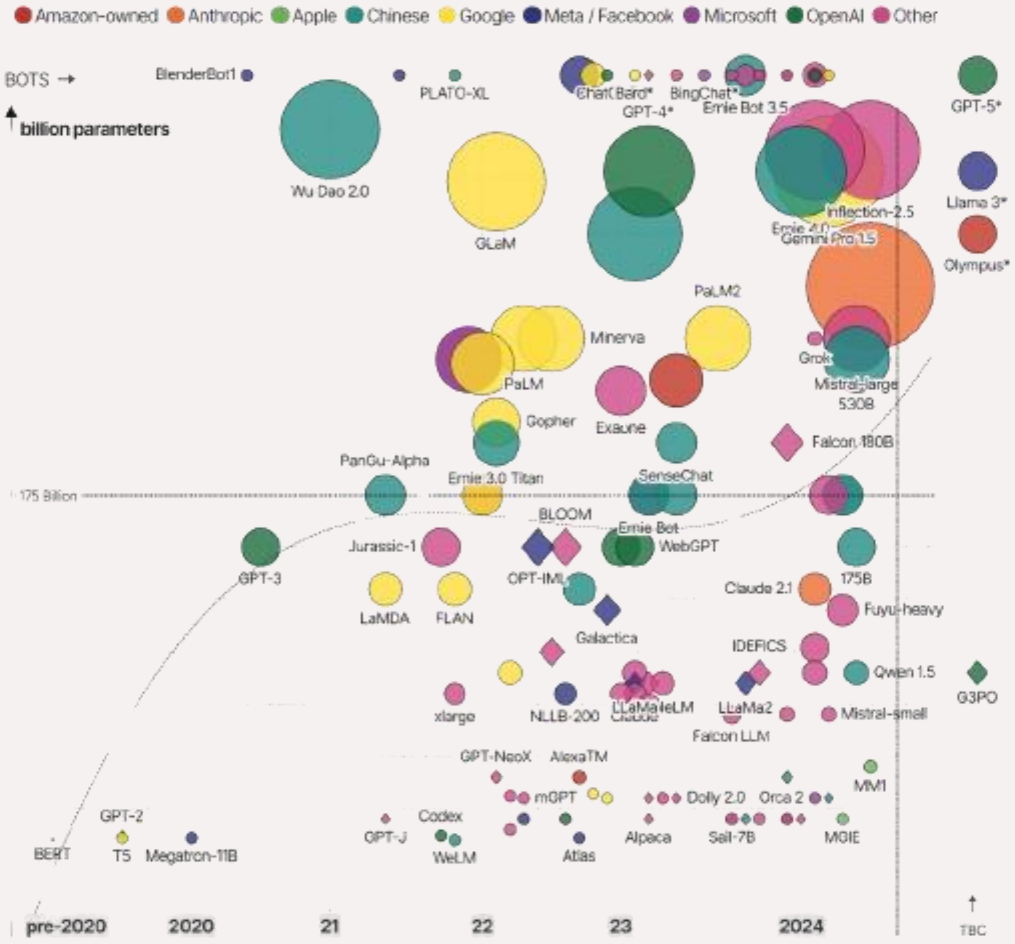
    - Q: How much larger? We don't know for sure ☺.

Content

- Large Language Models
- Generative NLP
- Learning from Human Preferences
- **Final Thoughts**

LLM Zoo

- ChatGPT represented a paradigmatic shift
  - From encoders to decoders *for everything*
- Open LLMs obtained with more or less the same recipe
  - Come in pairs – vanilla LLM + instruction-tuned variant
    - Llama
    - Mistral / Mixtral
    - Command R+
    - Nemotron
    - …



Source: https://lnkd.in/dbG3JkRZ

Evaluating LLMs

Chiang, W. L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., ... & Stoica, I. Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. In Forty-first International Conference on Machine Learning.

- Normally, we have benchmark NLP datasets on which we measure the performance of LLMs
- Problem with evaluating LLMs: they've seen „the entire web"
  - In LM pretraining
  - In instruction-tuning, likely to have seen most of benchmarks
  - Data Leakage!!!
- Evaluation datasets cannot be „static" anymore
  - LLM Chatbot Arena – crowdsourcing comparison on user-specified tasks, producing Elo-rankings for models (like in chess)

How Multilingual are LLMs?

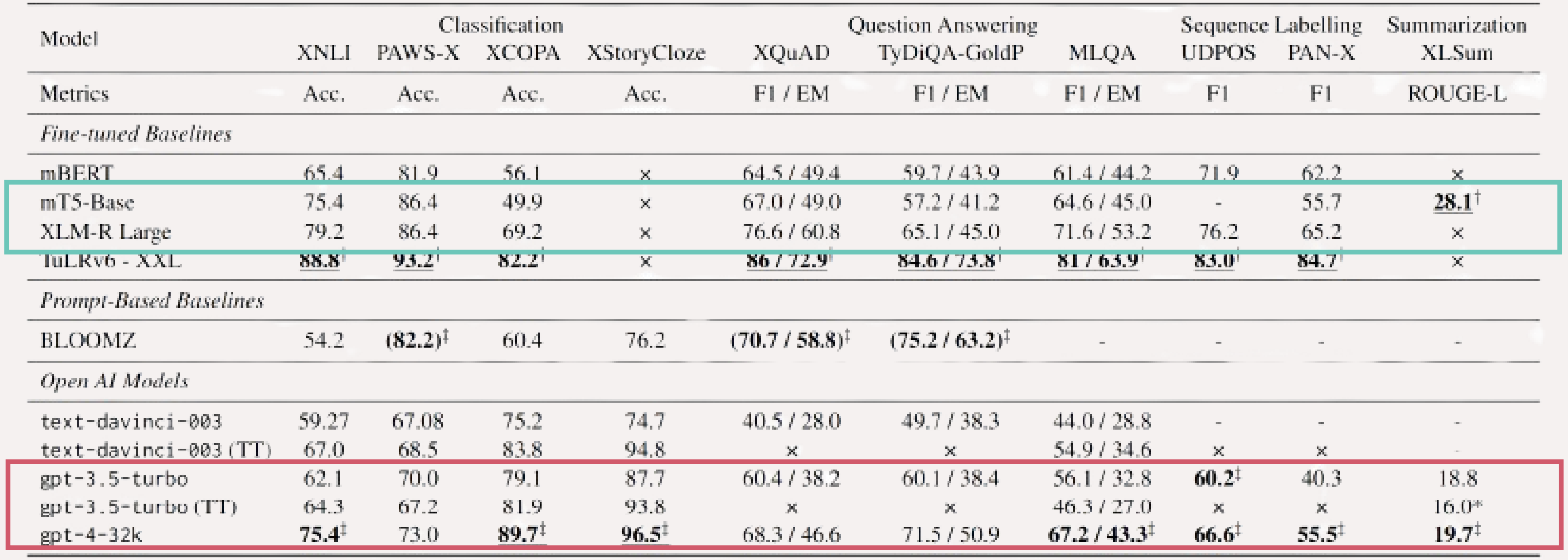
Ahuja, K., Diddee, H., Hada, R., Ochieng, M., Ramesh, K., Jain, P., ... & Sitaram, S. MEGA: Multilingual Evaluation of Generative AI. In The 2023 Conference on Empirical Methods in Natural Language Processing.

- Less multilingual than much smaller „massively multilingual encoders" (e.g., XLM-R)
- Q: Why?
  - 70+B param. models have to be trained on „all available text"
  - We cannot afford to „undersample" data for major languages → would result in much less capable LLMs
  - Relative underrepresentation of small languages much more pronounced
    - Result: LLMs are very Anglo-centric

How Multilingual are LLMs?

Ahuja, K., Diddee, H., Hada, R., Ochieng, M., Ramesh, K., Jain, P., ... & Sitaram, S. MEGA: Multilingual Evaluation of Generative AI. In The 2023 Conference on Empirical Methods in Natural Language Processing.

- E.g., Compare LLMs against XLT with „small" MMTs (e.g., XLM-R)



Model	Classification				Question Answering			Sequence Labelling		Summarization
	XNLI	PAWS-X	XCOPA	XStoryCloze	XQuAD	TyDiQA-GoldP	MLQA	UDPOS	PAN-X	XLSum
Metrics	Acc.	Acc.	Acc.	Acc.	F1 / EM	F1 / EM	F1 / EM	F1	F1	ROUGE-L
Fine-tuned Baselines										
mBERT	65.4	81.9	56.1	×	64.5 / 49.4	59.7 / 43.9	61.4 / 44.2	71.9	62.2	×
mT5-Base	75.4	86.4	49.9	×	67.0 / 49.0	57.2 / 41.2	64.6 / 45.0	-	55.7	**28.1**†
XLM-R Large	79.2	86.4	69.2	×	76.6 / 60.8	65.1 / 45.0	71.6 / 53.2	76.2	65.2	×
TuLRv6 - XXL	**88.8**†	**93.2**†	**82.2**†	×	**86 / 72.9**†	**84.6 / 73.8**†	**81 / 63.9**†	**83.0**†	**84.7**†	×
Prompt-Based Baselines										
BLOOMZ	54.2	**(82.2)**‡	60.4	76.2	**(70.7 / 58.8)**‡	**(75.2 / 63.2)**‡	-	-	-	-
Open AI Models										
`text-davinci-003`	59.27	67.08	75.2	74.7	40.5 / 28.0	49.7 / 38.3	44.0 / 28.8	-	-	-
`text-davinci-003` (TT)	67.0	68.5	83.8	94.8	×	×	54.9 / 34.6	×	×	-
`gpt-3.5-turbo`	62.1	70.0	79.1	87.7	60.4 / 38.2	60.1 / 38.4	56.1 / 32.8	**60.2**‡	40.3	18.8
`gpt-3.5-turbo` (TT)	64.3	67.2	81.9	93.8	×	×	46.3 / 27.0	×	×	16.0*
`gpt-4-32k`	**75.4**‡	73.0	**89.7**‡	**96.5**‡	68.3 / 46.6	71.5 / 50.9	**67.2 / 43.3**‡	**66.6**‡	**55.5**‡	**19.7**‡

Final Thoughts

- Scale matters more than we'd like to admit…
  - The Bitter Lesson [of AI progress] (by Rich Sutton)

> „The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. "

- Progress in NLP needs to scale both data and computation
  - Scaling data is much more difficult
  - Scaling data for thousands of low-resources languages is largely infeasible

  - **Q$_1$**: How do we get to „ChatGPT" in Quechuan?
  - **Q$_2$**: How do we get to good Named Entity Recognition (NER) in Quechuan?

The End

Image: Alexander Mikhalchyk