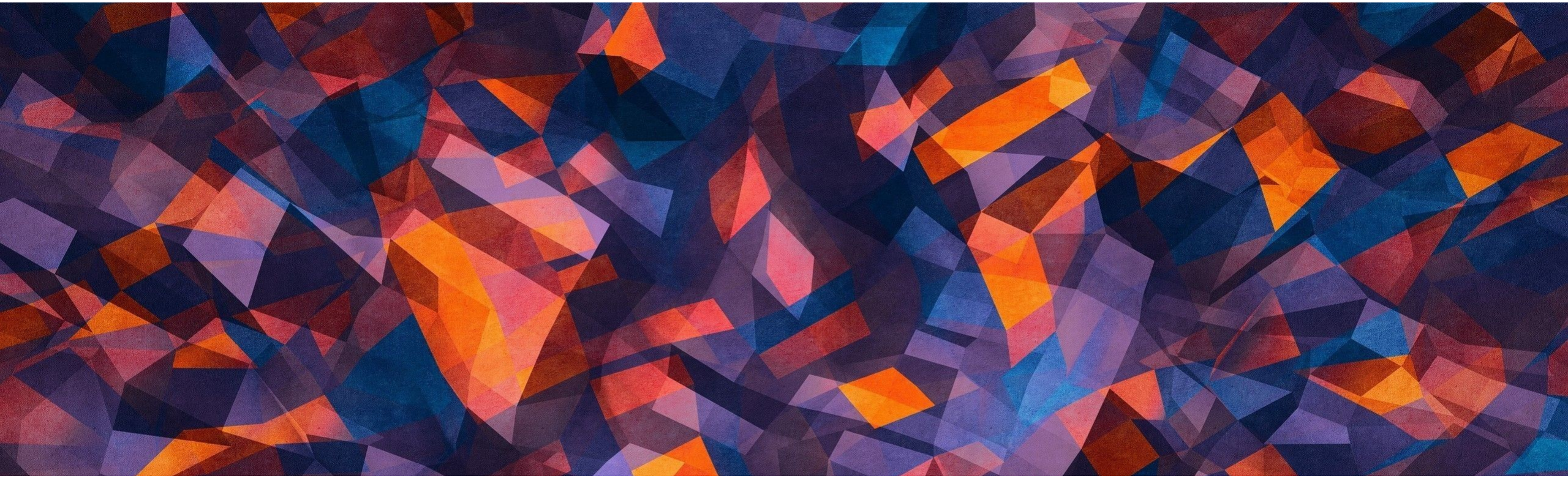
Universität St.Gallen



GSERM - Deep Learning: Fundamentals and Applications

Session 9: GPT, Foundation Models and Transfer Learning

Damian Borth, Korbinian Riedhammer, Marco Schreyer

Universität St.Gallen

Today's Menu

- The Evolution of GPT
  - Byte Pair Encoding
  - Generative Pretraining
  - Multi-task learning
  - Prompt Engineering
  - Reinforcement Learning from Human Feedback (RLHF)
- Transfer Learning and Fine Tuning
  - Natural Language Processing
  - Speech Processing
  - Model Repositories

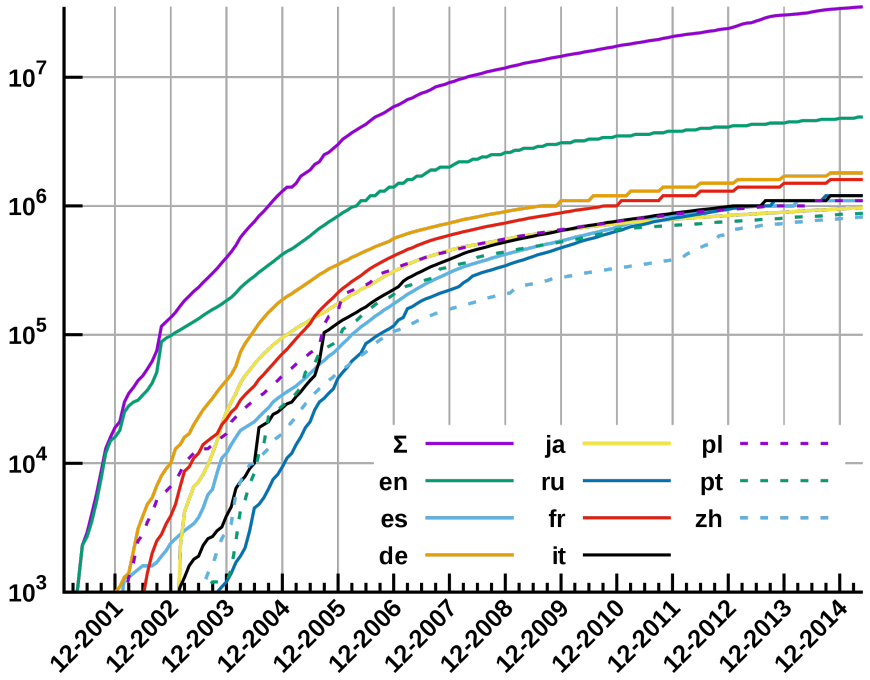
Main Issues in Neural Language Modeling

- Data
- Tokenization
- Compute
- Benchmarking

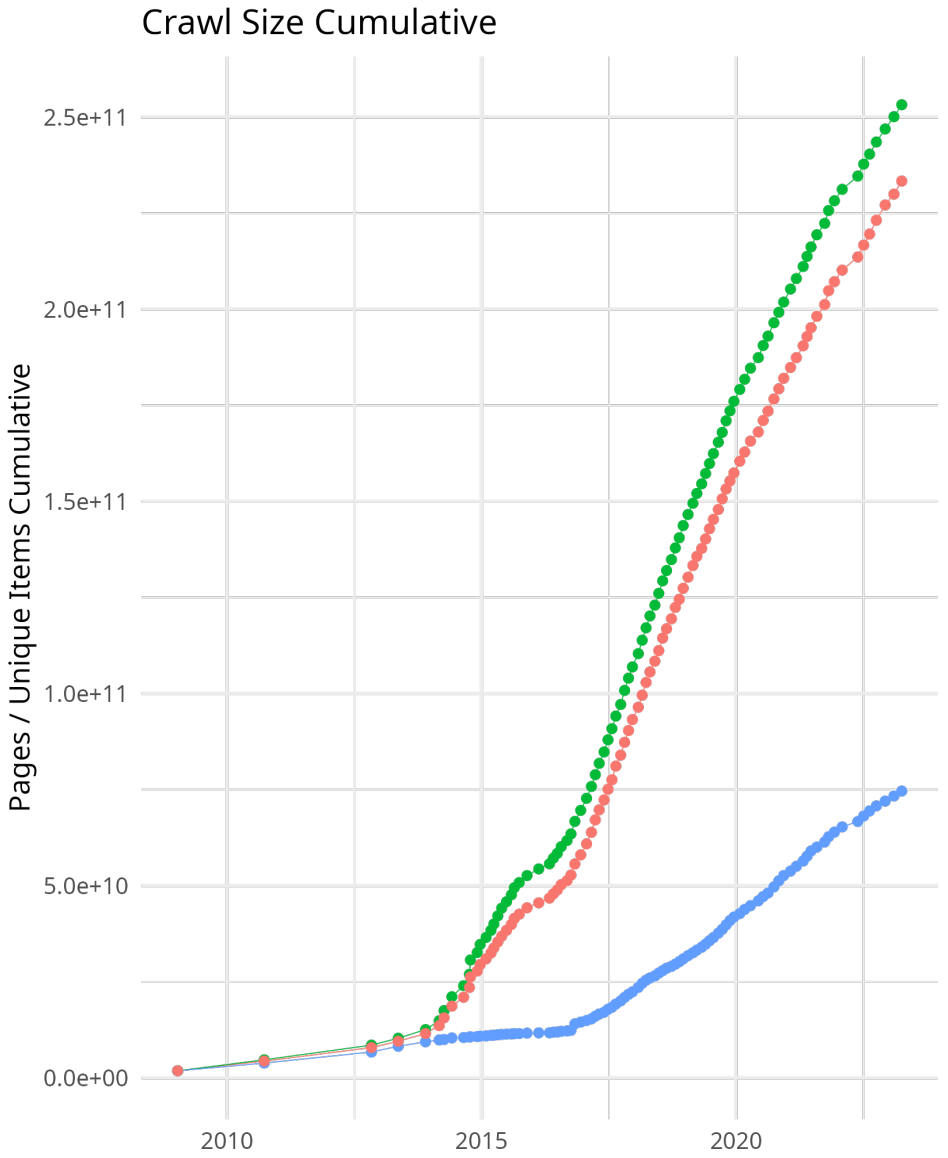
Universität St.Gallen

Data

- Rise of openly accessible data sets



Wikipedia (~21GB)



CommonCrawl (~380TB, 2022)

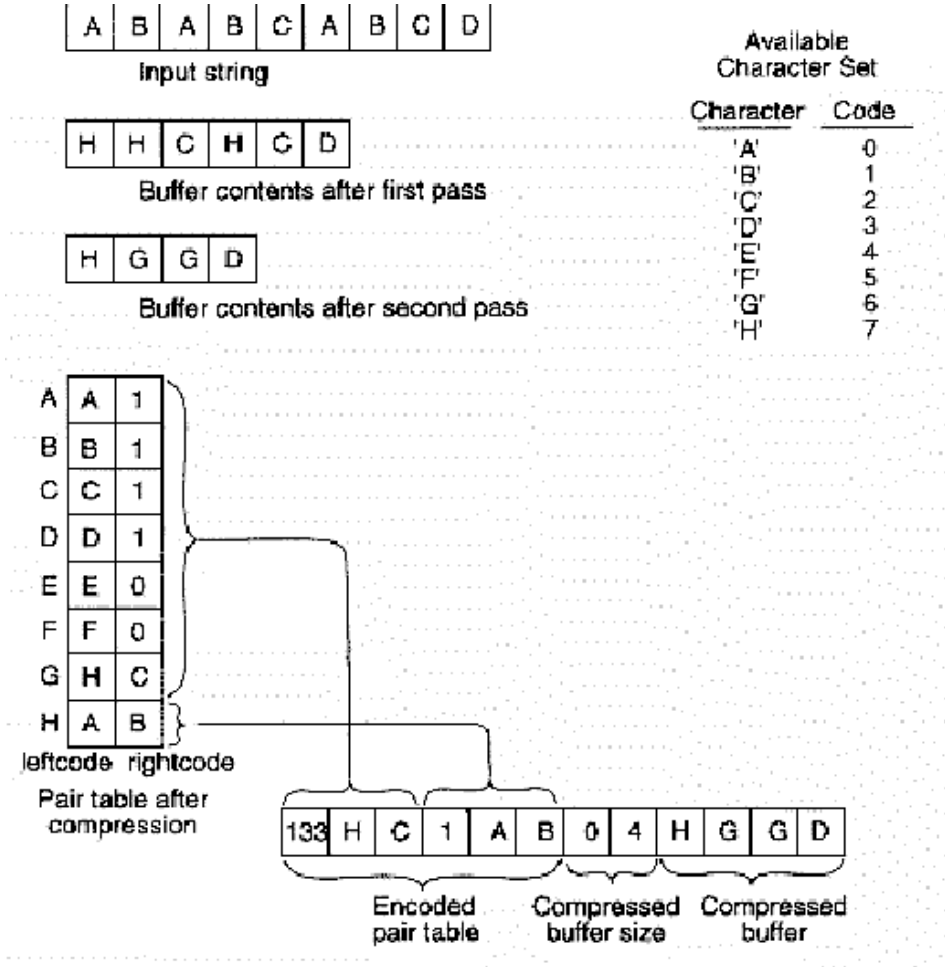
Tokenization

- Traditional NLP
  - word-based using a lexicon
  - stemming
- Big data NLP
  - character n-grams (cf. fasttext)
  - **Byte Pair Encoding (BPE)**

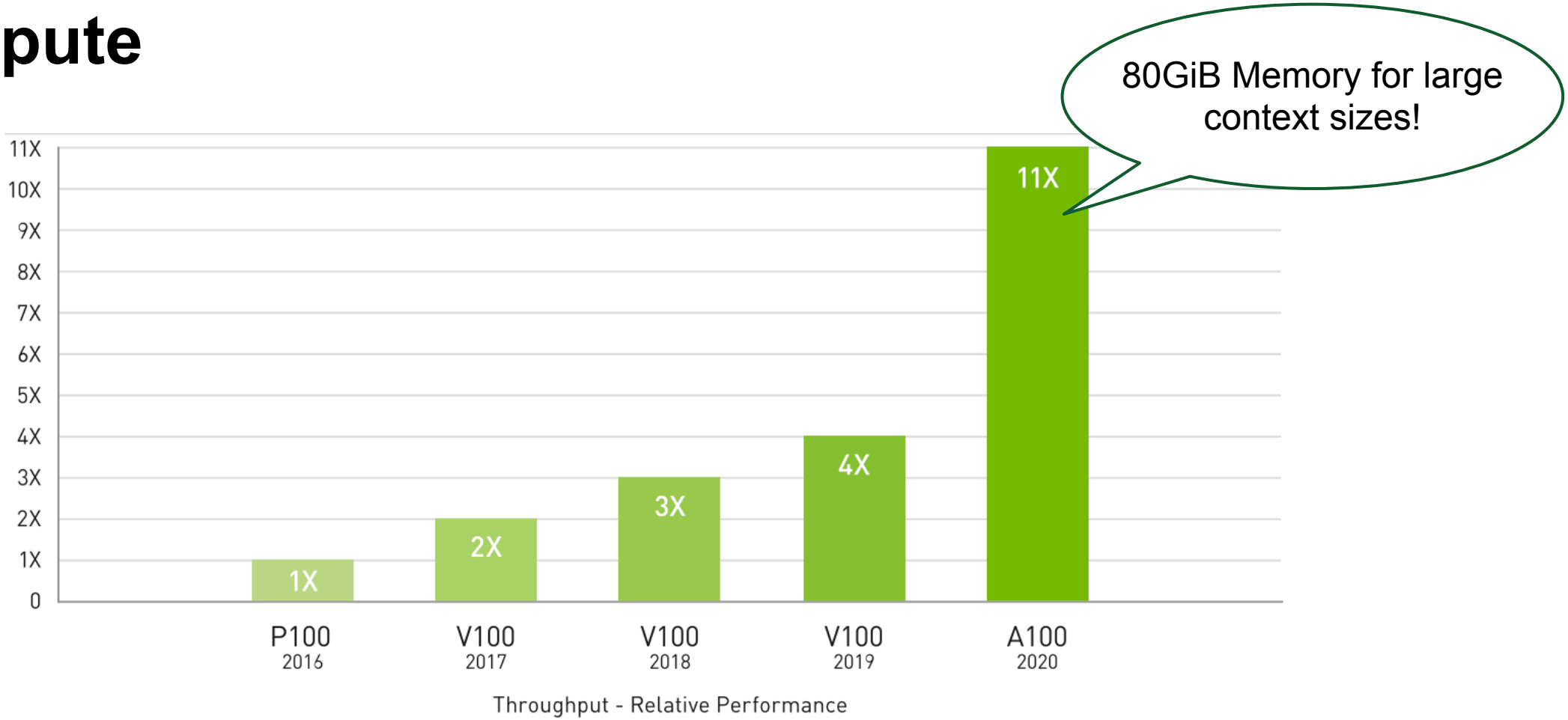
Universität St.Gallen

Byte Pair Encoding

- P. Gage, 1994: *A new algorithm for data compression*. The C Users Journal, Volume 12, Issue 2, 1994
  - Replace common pairs of bytes by single bytes
  - In-memory, multi-pass
- Modern NLP
  - adjust for unicode
  - Sennrich, Haddow and Birch, 2015 (https://arxiv.org/abs/1508.07909)



Compute



Source: Nvidia

Universität St.Gallen

Benchmarking

- https://gluebenchmark.com/
  - General Language Understanding Evaluation
  - CoLA: Linguistic Acceptability
  - SST-2: Sentiment
  - MRPC, QQP: semantic equivalence
  - STS-B: test similarity
  - MNLI, RTE: textual entailment
  - QNLI: is-answer?
  - WNLI: entailment after reference substitution

Radford et al., 2019: Improving Language Understanding by Generative Pre-training

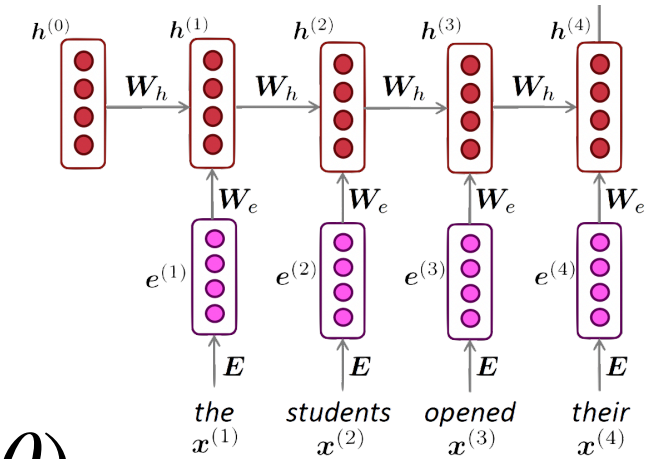
Generative Pre-Trained Transformers

- Prior work focused on learning models for specific tasks (sentiment, entailment, etc.) – they didn't generalize well!
- *Better:* semi-supervised learning (and some tricks)
  1. Unsupervised Language Modelling ("pre-training")
  2. Supervised fine-tuning
  3. Task-specific input transformations

Radford et al., 2018: Improving language understanding by generative pre-training

GPT: LM pre-training

- Train an auto-regressive transformer (decoder) language model
- Using BPE, token & positional embedding
- …on large (!) quantities of text!



$$\mathscr{L}_1(T) = \sum_i \log P(t_i \mid t_{i-k}, \ldots, t_{i-1}; \theta)$$

sequence of tokens

context window size

parameters

GPT: Supervised Fine-Tuning

- Assemble a dataset $\mathscr{C}$ with sequences $\mathbf{x}$ and according labels y
- (Super)GLUE gives us a rich set of tasks and datasets!
- Add linear output layer to final transformer block

$$L_2(\mathscr{C}) = \sum_{(\mathbf{x},y)} \log P(y \mid x_1, \ldots, x_m; \theta)$$

dataset

token sequence

label of this sequence

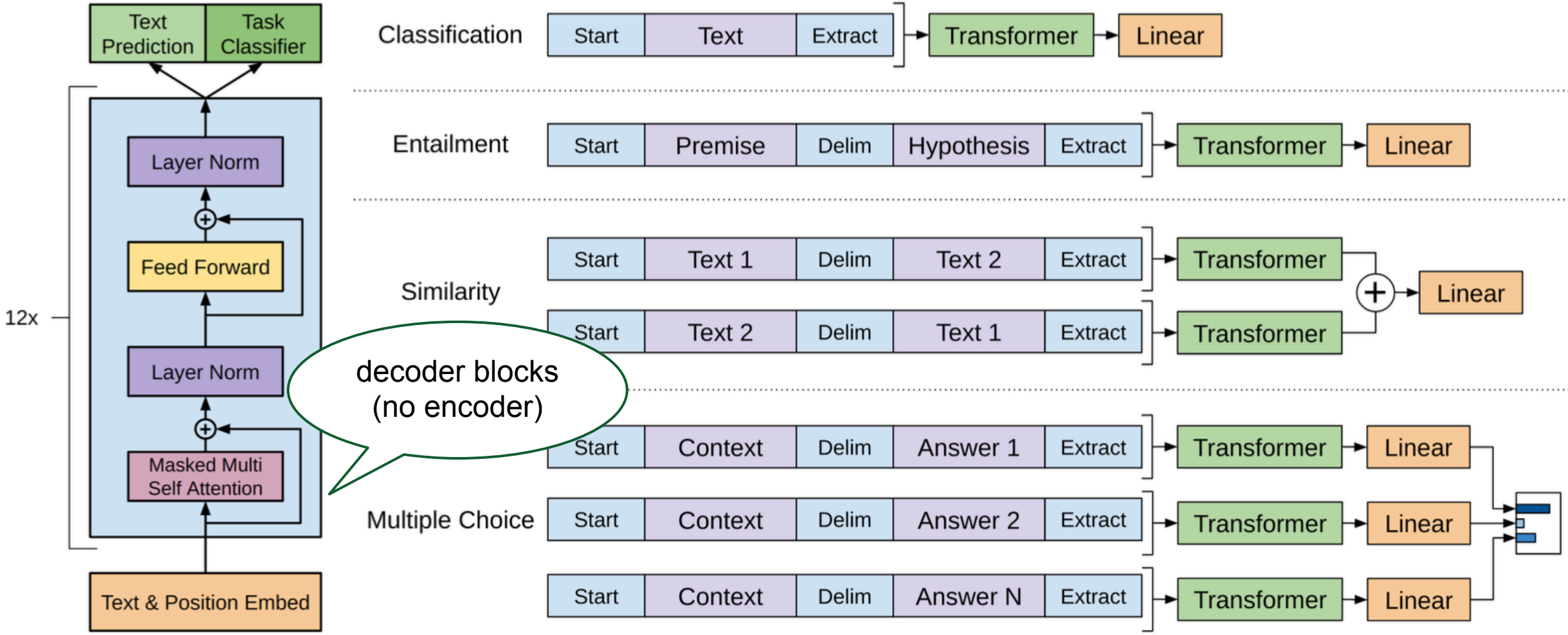
Optionally, for better performance and convergence: $L_3 = L_2(\mathscr{C}) + \lambda L_1(\mathscr{C})$

Universität St.Gallen

GPT: Task-specific input transformations

- For supervised training, we need to rearrange the input so that it works with our architecture
  - Start and end tokens for input sequences
  - Delimiter tokens in between parts of input
- Textual entailment: introduce ‘$’ token in between premise and hypothesis
- Similarity: provide pairs in both orders
- QA/Reasoning: [document; question; $; answer]

GPT: Architecture at a Glance



Universität St.Gallen

GPT: Some More Details

	GPT (2018)
Data	BooksCorpus (~5GB)
BPE	40,000
Parameters	117 Million
Decoder Layers	12
Context Token Size	512
Hidden Layer	768
Batch Size	64

nota bene: not words!

Results from the original tech report

Table 4: Semantic similarity and classification results, comparing our model with current state-of-the-art methods. All task evaluations in this table were done using the GLUE benchmark. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

Method	Classification		Semantic Similarity			GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]		**93.2**	-	-	-	-
TF-KLD [23]		-	**86.0**	-	-	-
ECNU (mixed ensemble)		-	-			
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2			
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8		68.9
Finetuned Transformer LM (ours)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**

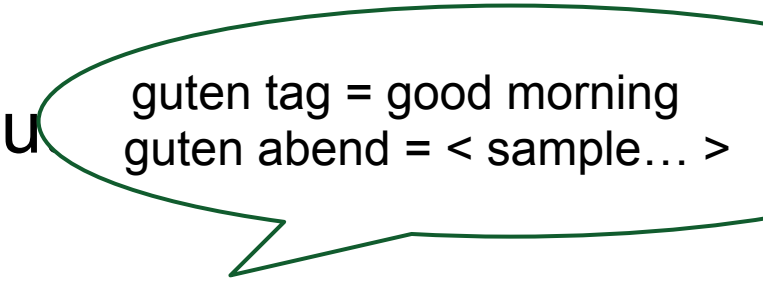
2018: Hey, LSTMs still around! 🙃

6/2023: 91% 🥺

Radford et al., 2019: Language Models are Unsupervised Multitask Learners

GPT-2

- Basically like GPT, just bigger
  - Larger context, more parameters
  - More data: WebText (40GB human curated, by tracing reddit outbound)
  - Better BPE (prevent split across character categories), 50k
- Paving the way to zero shot learning
  - Introduced task conditioning (ie. same input bu depending on task)
  - Instead of separators, use natural language instructions



Universität St.Gallen

GPT-2: Zero Shot Learning

- Technically, no training or fine-tuning allowed
- Model is “primed” with training data, e.g.
  - “guten tag = good morning” …
- At last, sample from model to get answer, e.g.
  - “guten abend = …”

GPT-2: some more details

	GPT (2018)	GPT-2 (2019)
Data	BooksCorpus (5GB)	WebText (40GB)
BPE	40k	50k (tweaked)
Parameters	117 Million	1.5 Billion
Decoder Layers	12	48
Context Token Size	512	1024
Hidden Layer	768	1600
Batch Size	64	512

Universität St.Gallen

Results from the original tech report (1)

GPT

These are all rather simple “fill the gap” tests

Language Models are Unsupervised Multitask Learners

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23						1.08	18.3	**21.8**
117M	**35.13**	45.99						1.17	37.50	75.20
345M	**15.60**	55.48						**1.06**	26.37	55.72
762M	**10.87**	**60.12**						**1.02**	22.05	44.575
1542M	**8.63**	**63.24**						**0.98**	**17.48**	42.16

Ok, we get it: **bigger is better!**

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

GPT-2

Results from the original tech report (2)

TL;DR is often found in the reddit data!

Still, pretty terrible results on a fairly simple dataset

	R-1	R-2	R-L	R-AVG
...om-Up Sum	**41.22**	**18.68**	**38.34**	**32.75**
Lede-3	40.38	17.66	36.62	31.55
Seq2Seq + Attn	31.33	11.81	28.83	23.99
GPT-2 TL;DR:	29.34	8.27	26.58	21.40
Random-3	28.78	8.63	25.52	20.98
GPT-2 no hint	21.58	4.03	19.47	15.03

Table 4. Summarization performance as measured by ROUGE F1 metrics on the CNN and Daily Mail dataset. Bottom-Up Sum is the SOTA model from (Gehrmann et al., 2018)

Brown et al., 2020: Language Models are Few-Shot Learners (NeurIPS 2020)

GPT-3

- **Vision**: Build a general model ("foundation model") that will learn any task with only a few examples ("few shot learner")
- More of everything…

	GPT (2018)	GPT-2	GPT-3
Data	BooksCorpus (5GB)	+WebText (40GB)	+CommonCrawl+Wiki_en
Parameters	117 Million	1.5 Billion	175 Billion
Decoder Layers	12	48	96
Context Token Size	512	1024	2048
Hidden Layer	768	1600	12288
Batch Size	64	512	3.2M

Brown et al., 2020: Language Models are Few-Shot Learners (NeurIPS 2020)

GPT-3

- “**in-context learning**”:
  - prepend examples of the task before your actual example/query
  - k = 0: zero-shot
  - k = 1: one-shot
  - k > 1: few-shot
  - …but still no gradient update!

GPT-3: Few-Shot Learning on SuperGLUE

Few-Shot

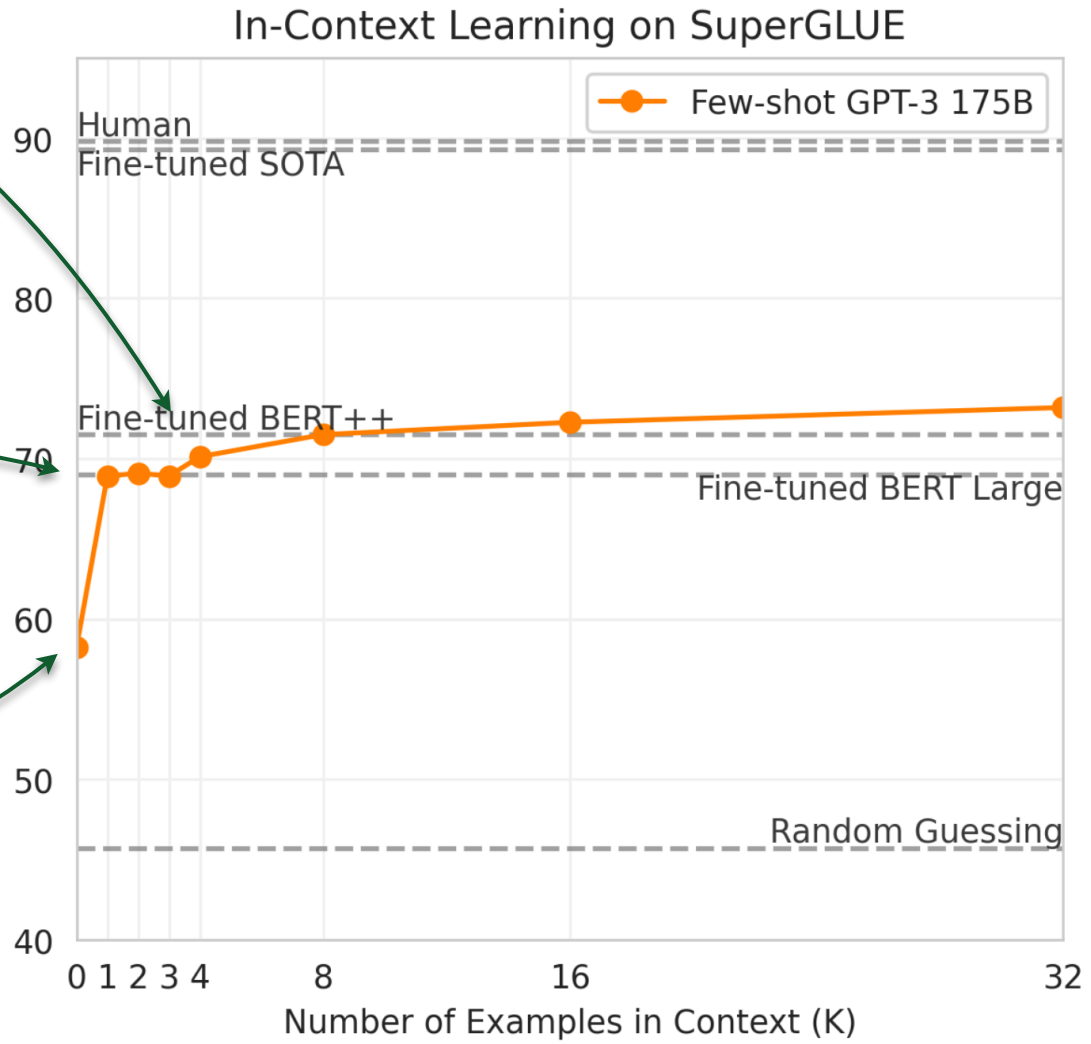
```
Translate English to French:
peppermint => menthe povrée
sea otter => loutre de mer
…
cheese =>
```

One-Shot

```
Translate English to French:
sea otter => loutre de mer
cheese =>
```

Zero-Shot

```
Translate English to French:
cheese =>
```



GPT-3: Results from the original paper

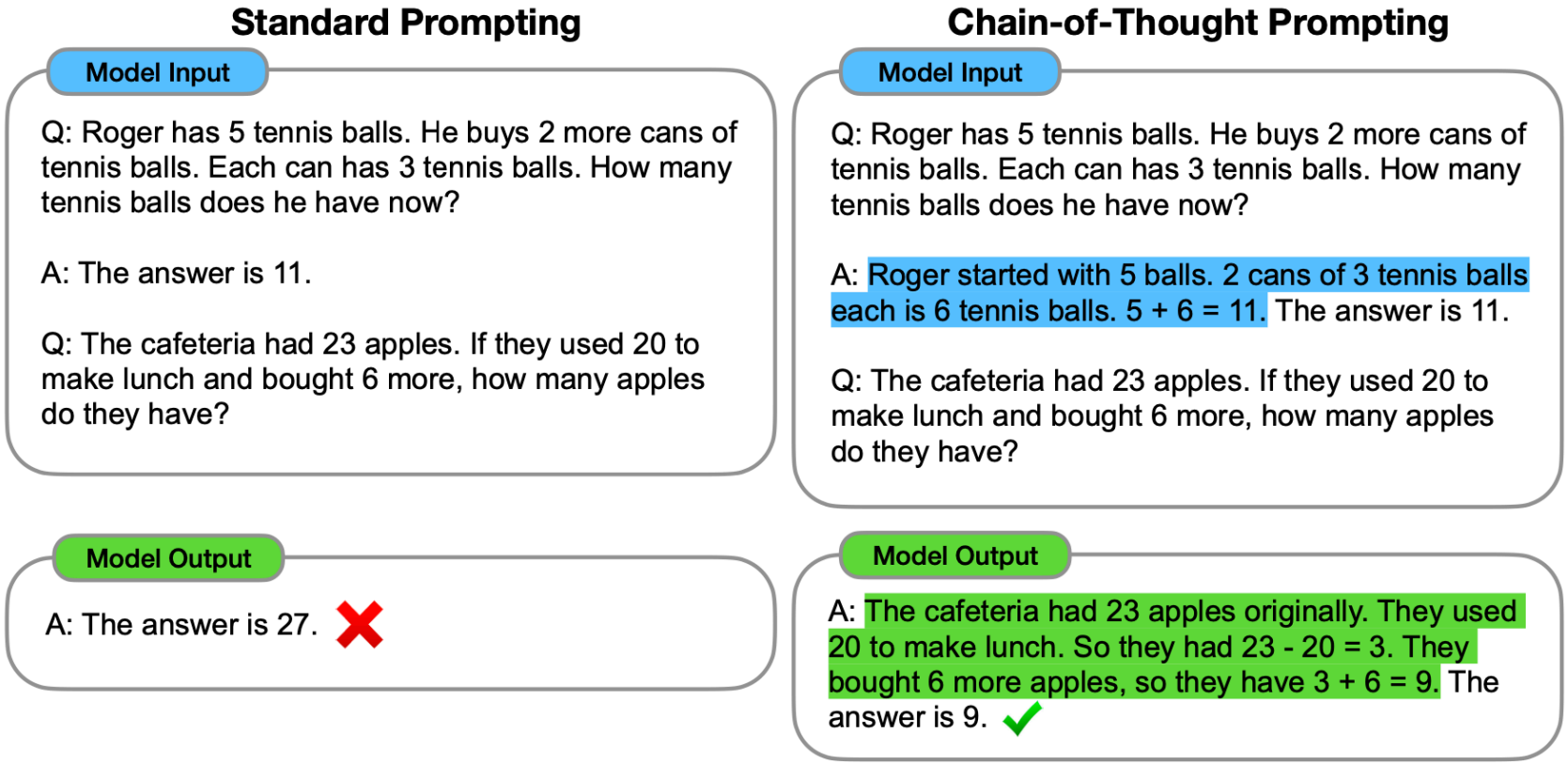
	SuperGLUE Average	BoolQ Accuracy	CB Accuracy	CB F1	COPA Accuracy	RTE Accuracy
Fine-tuned SOTA	**89.0**	**91.0**	**96.9**	**93.9**	**94.8**	**92.5**
Fine-tuned BERT-Large	69.0	77.4	83.6	75.7	70.6	71.7
GPT-3 Few-Shot	71.8	76.4	75.6	52.0	92.0	69.0

	WiC Accuracy	WSC Accuracy	MultiRC Accuracy	MultiRC F1a	ReCoRD Accuracy	ReCoRD F1
Fine-tuned SOTA	**76.1**	**93.8**	**62.3**	**88.2**	**92.5**	**93.3**
Fine-tuned BERT-Large	69.6	64.6	24.1	70.0	71.3	72.0
GPT-3 Few-Shot	49.4	80.1	30.5	75.4	90.2	91.1

Table 3.5: Performance of GPT-3 on SuperGLUE compared to fine-tuned baselines and SOTA. All results are reported on the test set. GPT-3 few-shot is given a total of 32 examples within the context of each task and performs no gradient updates.

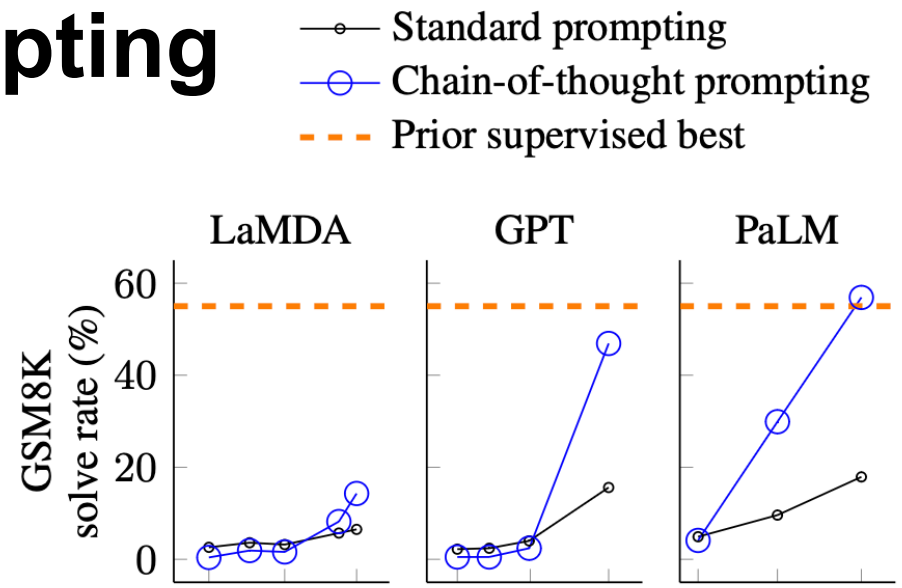
Chain-of-Thought Prompting

- In-context learning seems to have limited performance
- Solution: Change the prompts!



Chain-of-Thought Prompting

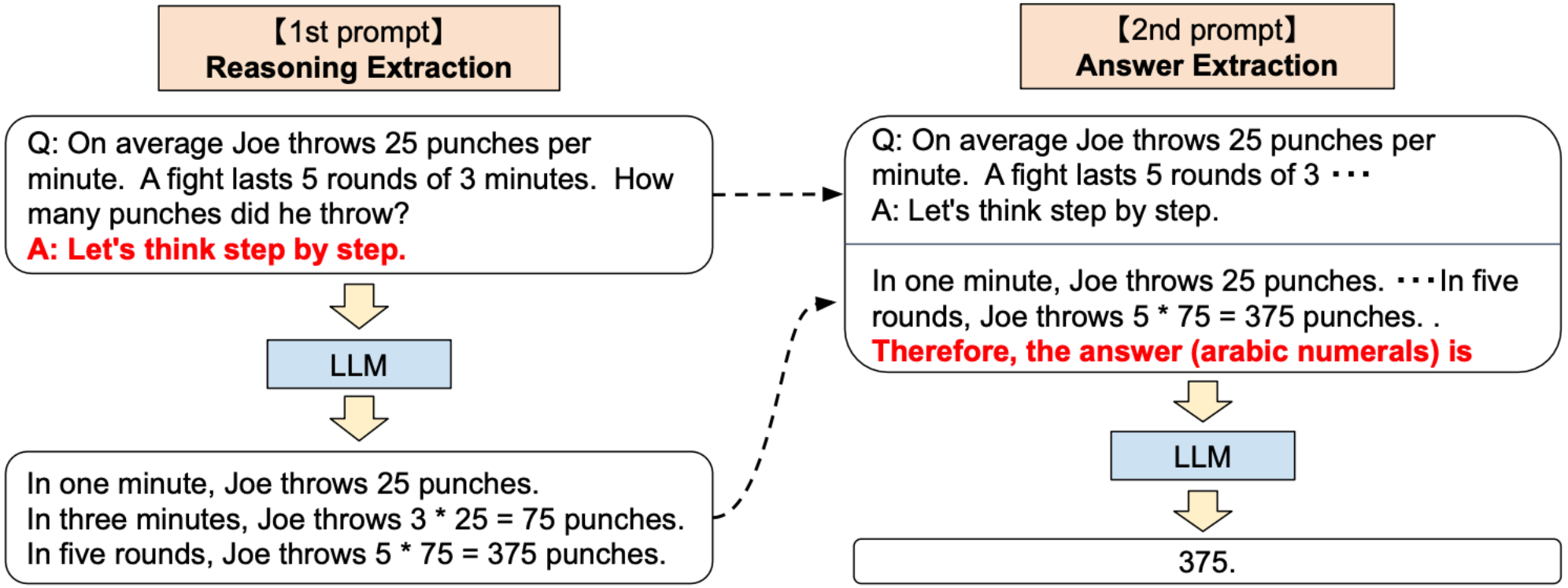
- Yes, it works!
- But…



As for limitations, we first qualify that although chain of thought emulates the thought processes of human reasoners, this does not answer whether the neural network is actually "reasoning," which we leave as an open question. Second, although the cost of manually augmenting exemplars with chains of thought is minimal in the few-shot setting, such annotation costs could be prohibitive for finetuning (though this could potentially be surmounted with synthetic data generation, or zero-shot generalization). Third, there is no guarantee of correct reasoning paths, which can lead to both correct and incorrect answers; improving factual generations of language models is an open direction for

Kojima et al., 2022. LLMs are Zero-Shot Reasoners (NeurIPS 2022)

Zero-Shot Chain-of-Thought



Kojima et al., 2022. LLMs are Zero-Shot Reasoners (NeurIPS 2022)

Zero-Shot Chain-of-Thought *(on MultiArith dataset)*

No.	Category	Template	Accuracy
1	instructive	Let's think step by step.	**78.7**
2		First, (*1)	77.3
3			74.5
4		ng it into steps. (*2)	72.2
5		Let step by step.	70.8
6		Let's think like a detective step by step.	70.3
7		Let's think	57.5
8		Before we dive into the answer,	55.7
9		The answer is after the proof.	45.7
10	misleading	Don't think. Just feel.	18.8
11		Let's think step by step but reach an incorrect answer.	18.7
12		Let's count the number of "a" in the question.	16.7
13		By using the fact that the earth is round,	9.3
14	irrelevant	By the way, I found a good restaurant nearby.	17.5
15		Abrakadabra!	15.5
16		It's a beautiful day.	13.1
-		(Zero-shot)	17.7

"Prompt Engineering"

Ouyang et al., 2022: Training LMs to follow instructions with human feedback

Language Modeling ≠ Assisting Users

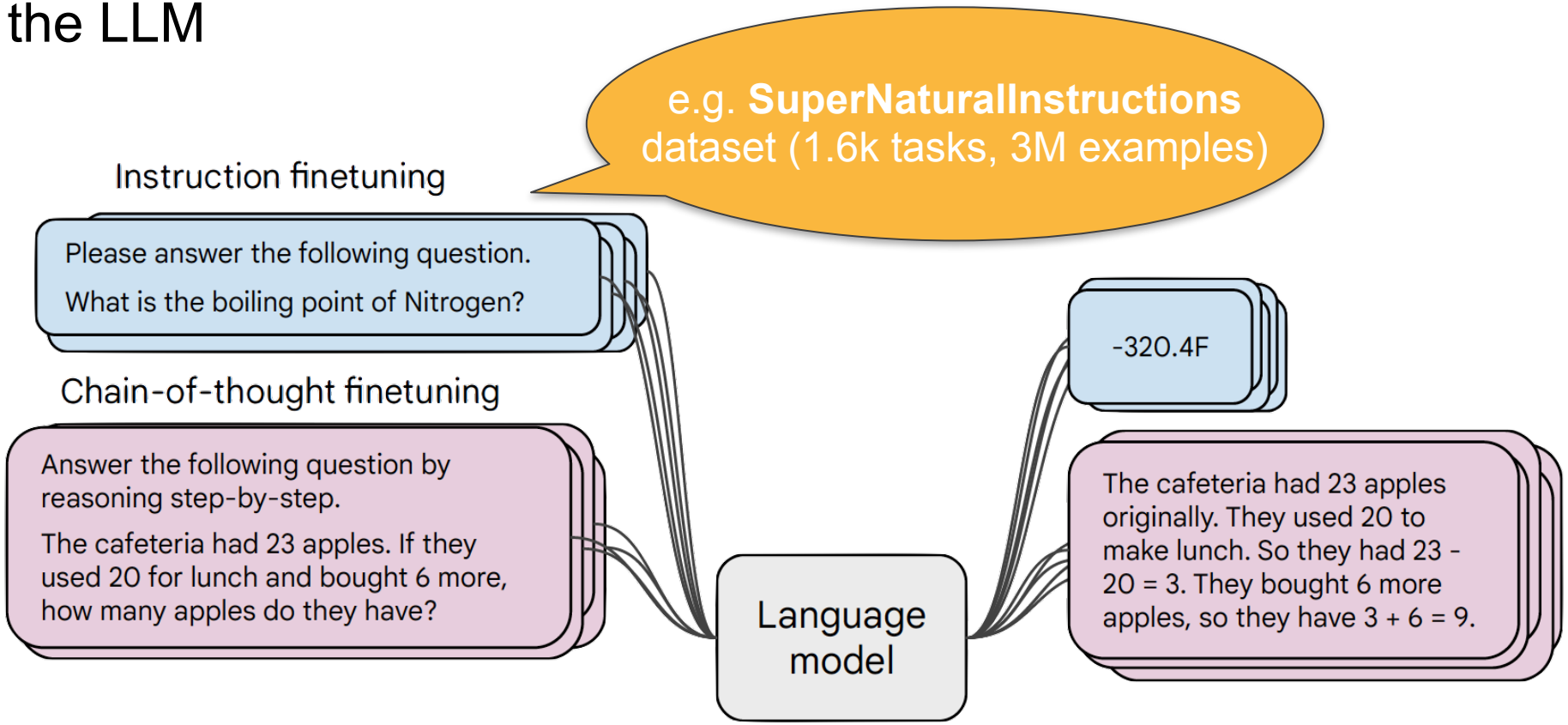
Prompt	*Explain the moon landing to a 6 year old in a few sentences.*
Completion	GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few
sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

LMs are not aligned with user intent!

Instruction Fine-Tuning (1)

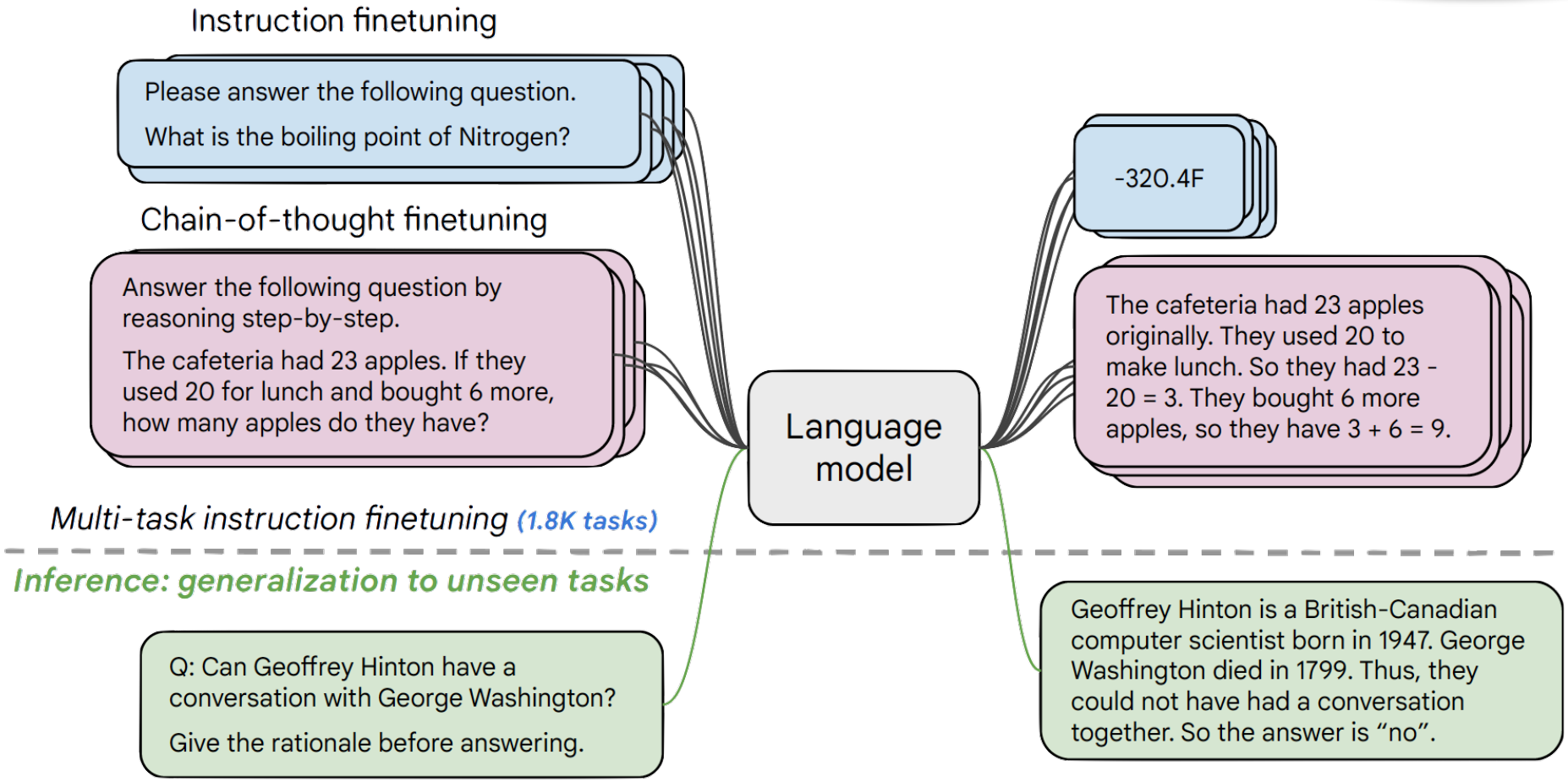
- Collect examples of (instruction, output) pairs across **many tasks** and fine-tune the LLM



Instruction Fine-Tuning (2)

FLAN-T5
SNI + 1.8k tasks

- Evaluate on **unseen task**



Limits of Instruction Fine-Tuning (FLAN)

- Ground-truth data is expensive to collect
- Open-ended (creative) tasks do not have a right answer
  - “Write a poem about deep learning”
- Language Modeling penalizes all token-level mistakes equally – but some errors are worse than others!
  - “You’re fired” vs. “You’re hired”!

From LMs to Assistants: Recap

- Zero-shot and few-shot in-context learning
  - No fine-tuning needed
  - prompt engineering can improve performance
  - Limits to what you can fit in context
  - Complex tasks will probably need parameter update
- Instruction fine-tuning
  - Simple and straight-forward, generalizes to unseen tasks
  - Collecting ground-truth for many tasks is expensive (and exhaustive…)

Mismatch between LM object and human preference

Optimizing for Human Preference

- Let's say, we're training for summarization
- Remember: We can sample multiple outputs!

```
SAN FRANCISCO,
California (CNN) --
A magnitude 4.2
earthquake shook the
San Francisco
...
overturn unstable
objects.
```

```
An earthquake hit San
Francisco. There was
minor property damage,
but no injuries.
```

👍

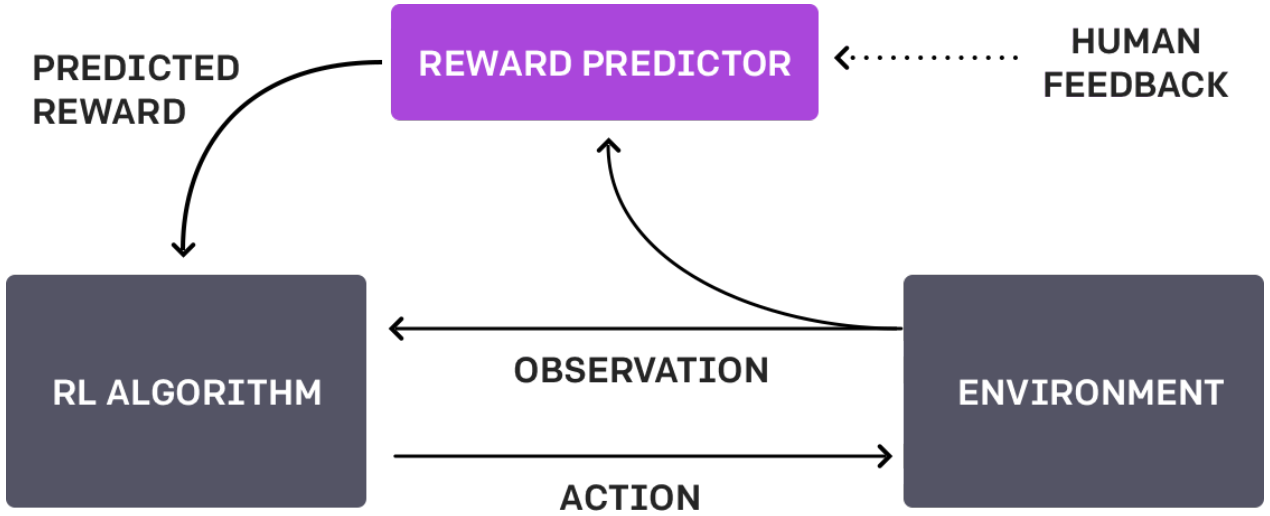
```
The Bay Area has good
weather but is prone
to earthquakes and
wildfires.
```

👎

Ouyang et al., 2022: Training LMs to follow instructions with human feedback

Optimizing for Human Preference

- Which one is better, or: which one has a higher reward?
- Mathematical framework: policy gradient for reinforcement learning



Reinforcement Learning from Human Feedback (RLHF)

A Model of Human Preference (1)

- **Problem**: human-in-the-loop is costly (and slow)
- **Solution**: Model human preference as a separate (NLP) task 🤓

```
A 4.2 magnitude
earthquake hit San
Francisco, resulting
in massive damage.
```

👍? 👎?

A Model of Human Preference (2)

- **Problem**: humans don't agree and are often miscalibrated
- **Solution**: ask for pair-wise comparison (binary preference)

`An earthquake hit San Francisco. There was minor property damage, but no injuries.` > `A 4.2 magnitude earthquake hit San Francisco, resulting in massive damage.` > `The Bay Area has good weather but is prone to earthquakes and wildfires.`

Universität St.Gallen

InstructGPT

Step 1

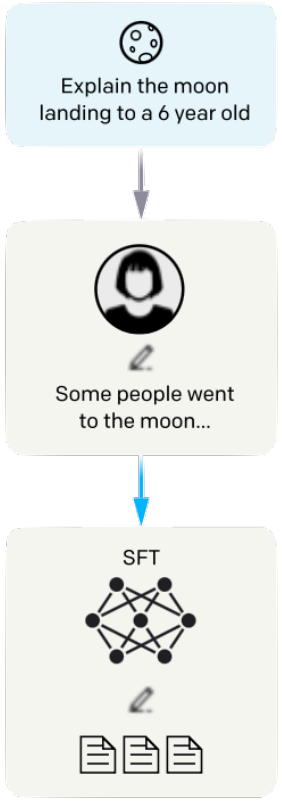
Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

30k tasks!

A labeler demonstrates the desired output behavior.

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



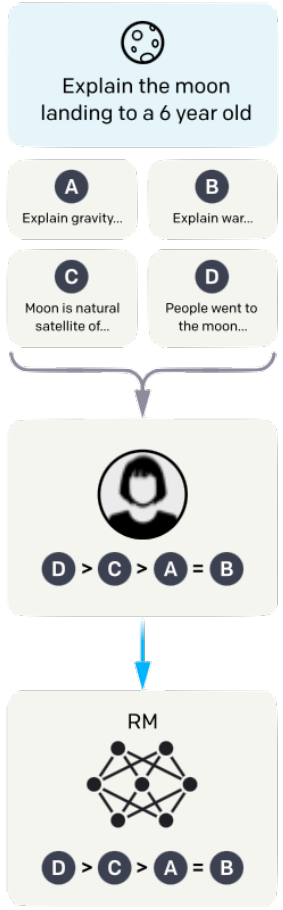
Step 2

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

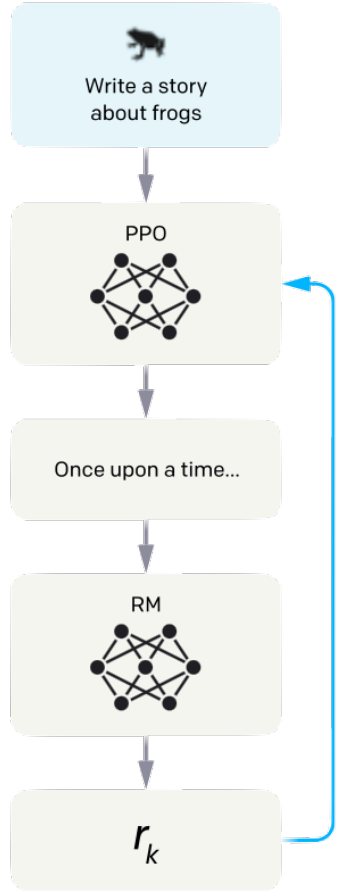
Optimize a policy against the reward model using reinforcement learning.

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.



Ouyang et al., 2022: Training LMs to follow instructions with human feedback

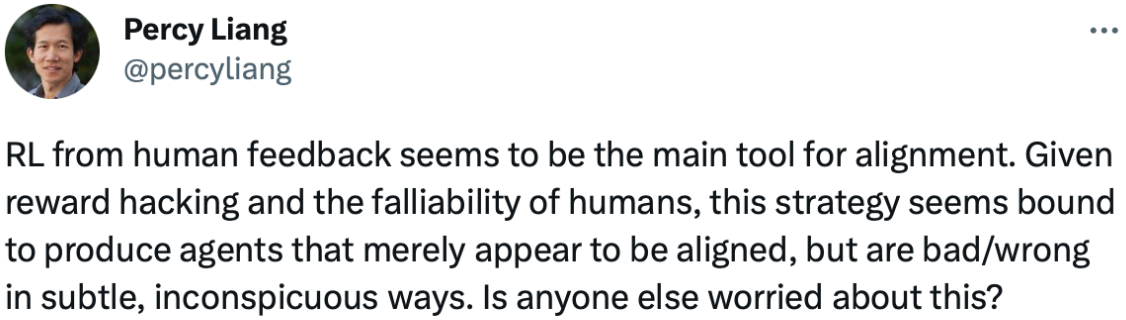
https://openai.com/blog/chatgpt

ChatGPT

- OpenAI not so open anymore… details not really revealed
- Blog post suggests:
  - GPT-3.5 (bigger, you guessed it…)
  - Instruction fine-tuning
  - RLHF
  - …on dialog data
- **Unclear**: How much “plain” (handcrafted, rule-based) engineering is
  - in the dialog state?
  - in the “last mile”? (e.g. for SQL queries, API calls)

A Note on RL with Reward Modeling

- **Human preference** is unreliable (and subject to change…)
  - "Reward hacking" is an issue in RL
  - Chatbots are rewarded to produce responses that seem authoritative and helpful, *regardless of truth*
  - Possible root cause of "alternative facts" and hallucinations
- **Models** of human preference will by design be inferior 🥺

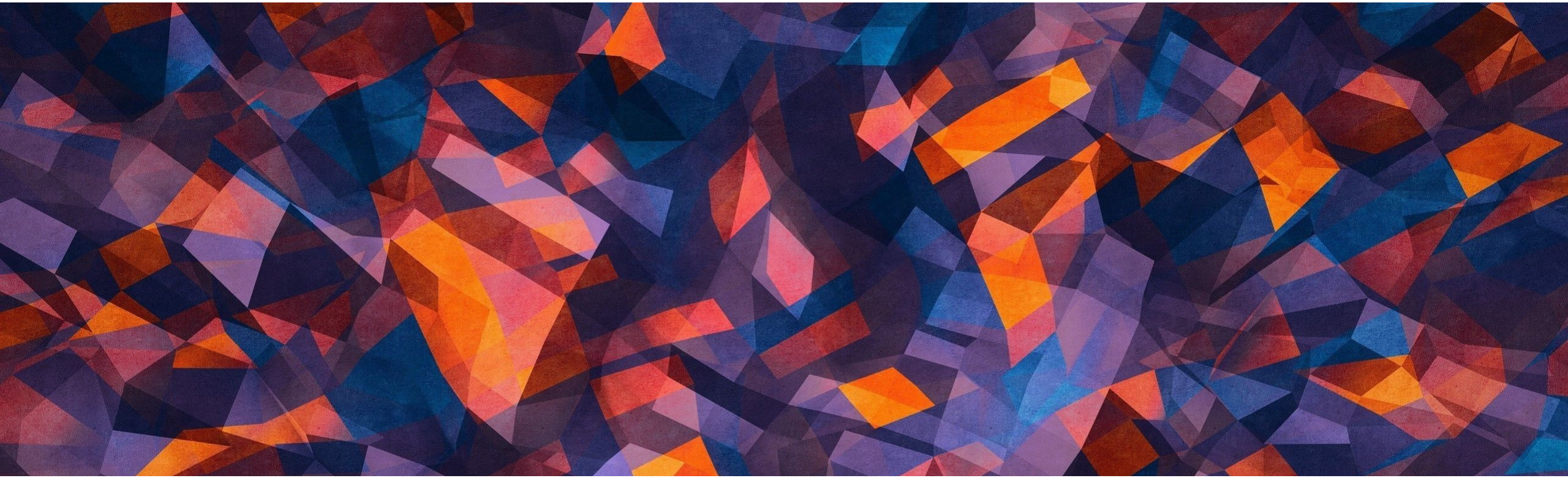


https://twitter.com/percyliang/status/1600383429463355392

Recap

- GPT
  - Generative pre-trained transformers for language modelling
  - Fine-tuned to (multitude of) tasks
  - “bigger is better”
- Prompt engineering (zero-/one-/few-shot)
- Instruction fine-tuning
- Reinforcement Learning from Human Feedback

Universität St.Gallen



Switching gears… Building Use Cases with AI

Universität St.Gallen

Building Tools with AI

- When building new applications, **we want to**…
  - use as much data as we can
  - train a large–as–needed model
- However, **we don't have**…
  - access to "big data"
  - time and resources to annotate thousands of examples
  - GPUs and energy for millions of Swiss Francs

Universität St.Gallen

Building Tools with AI

- Identify a large pre-trained model that fits your use case
  - These are often called “foundation models”
  - Many available open source (huggingface!)
- Chose a learning strategy
  - Fine-tuning?
  - Zero-/one-/few-shot?
  - RLHF?
- Label a few (hundred) examples
- Deploy!

Terminology

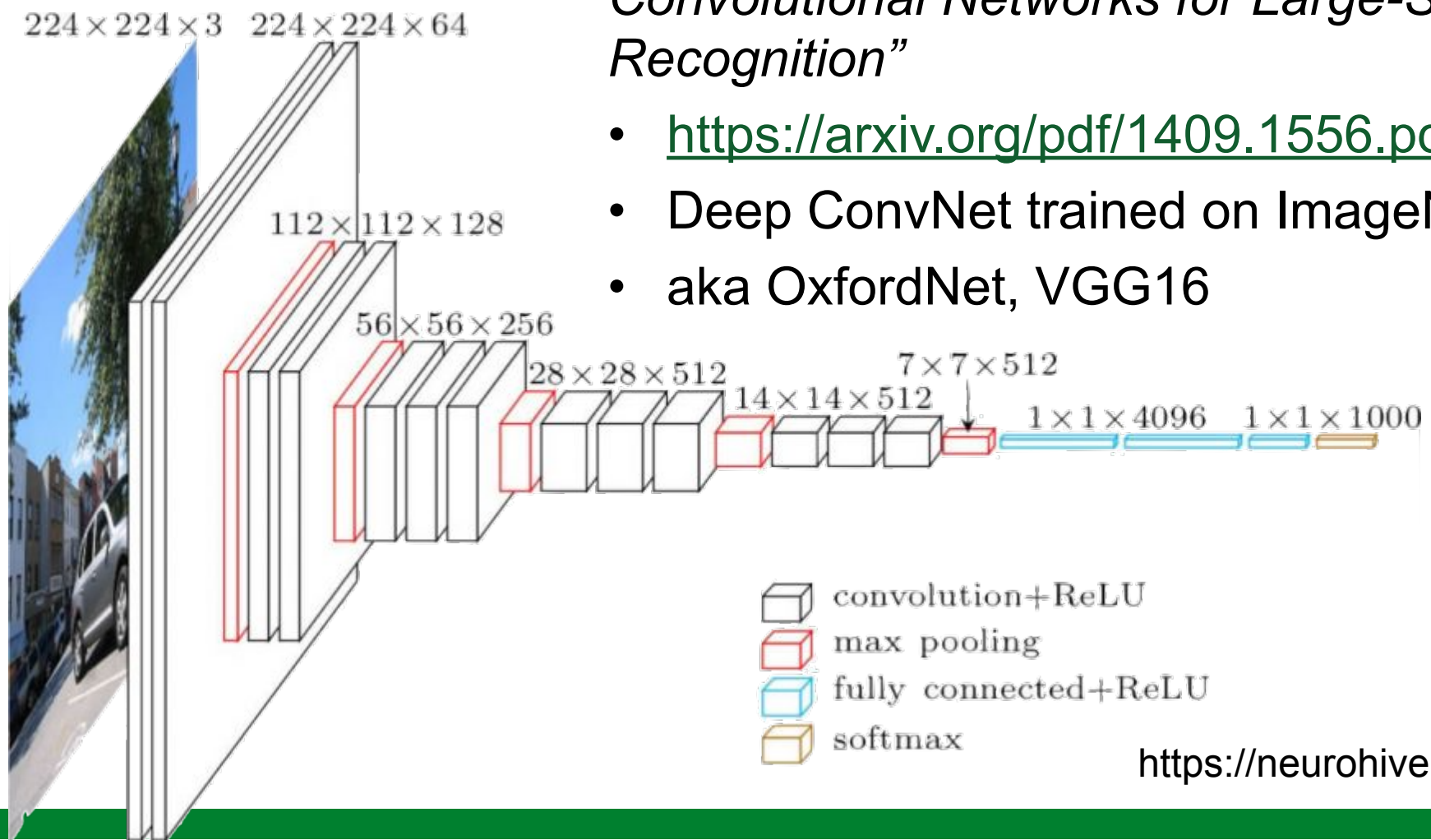
- Pre-training
- Foundation Model
- Representation learning
- Transfer learning

Universität St.Gallen

Image Processing

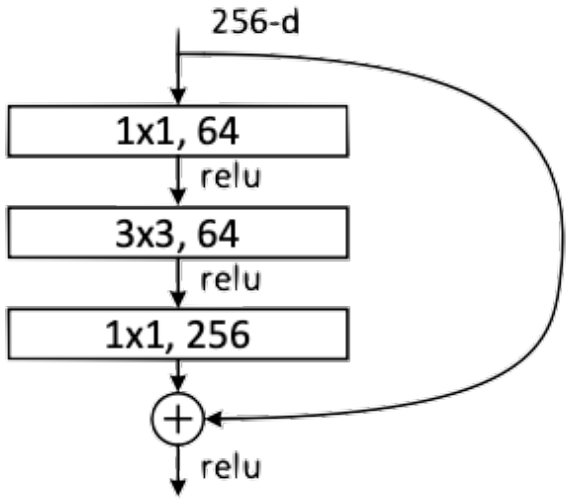
Image Processing

- Simonyan and Zisserman, 2015: *"Very Deep Convolutional Networks for Large-Scale Image Recognition"*
  - https://arxiv.org/pdf/1409.1556.pdf
  - Deep ConvNet trained on ImageNet (winner 2014!)
  - aka OxfordNet, VGG16



https://neurohive.io/en/popular-networks/vgg16/

Image Processing



- He et al., 2015: *"Deep Residual Learning for Image Recognition"*
  - https://arxiv.org/pdf/1512.03385.pdf
  - Deep ConvNet with residual connections
  - aka "ResNet" (resnet50)

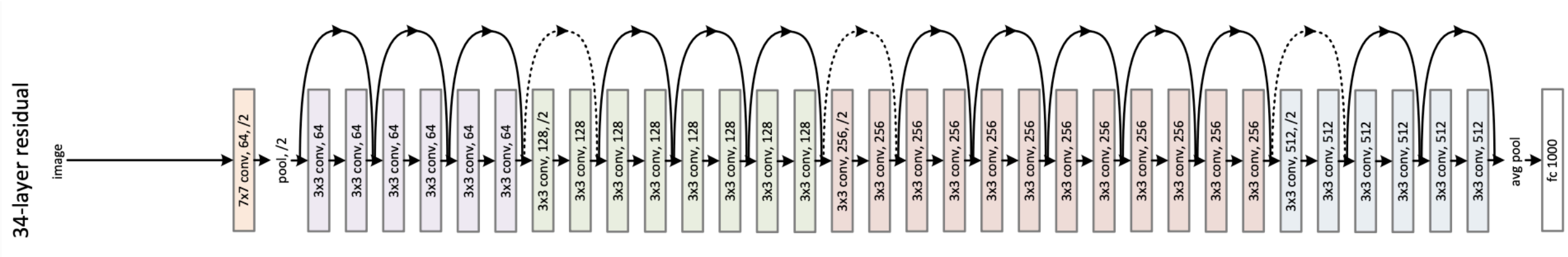
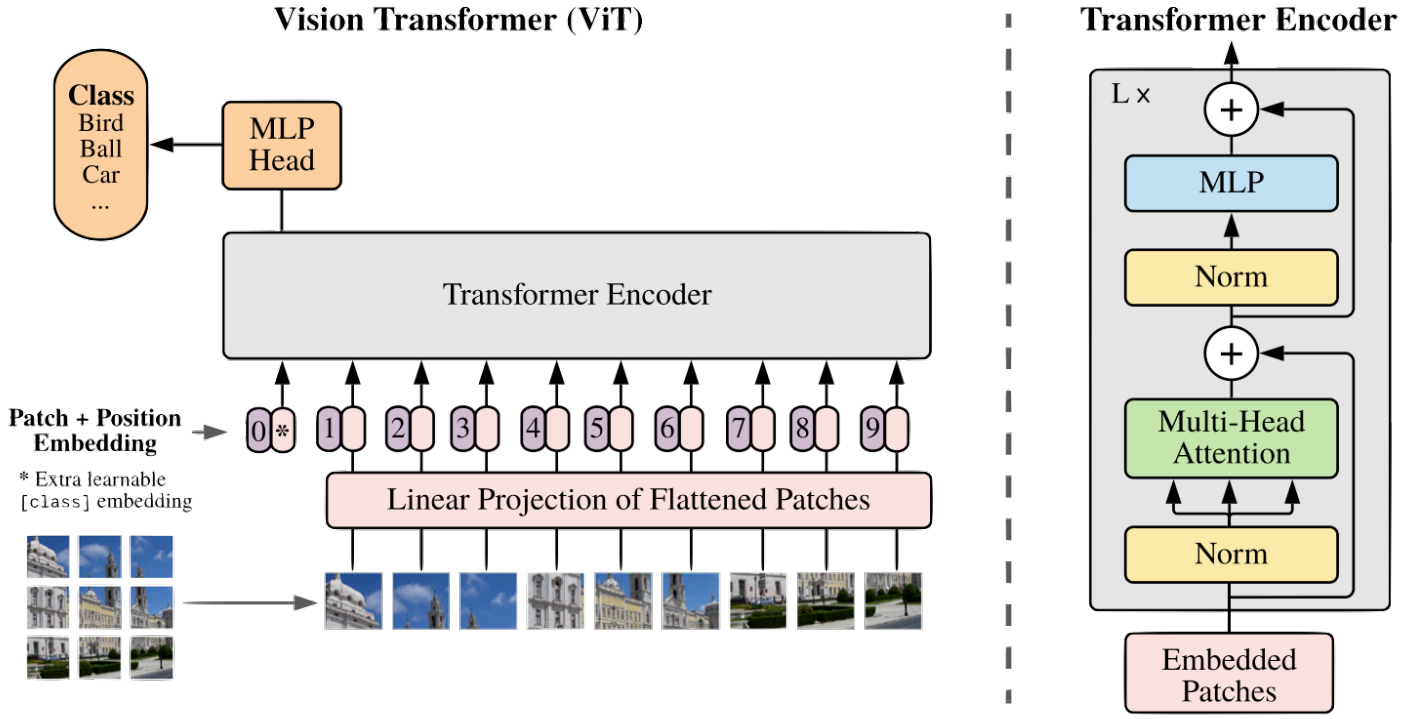


Image Processing

- Dosovotskiy et al. 2021: *"An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale"*
  - https://arxiv.org/pdf/2010.11929.pdf
  - Transformer with patch+position embedding
  - aka "ViT"



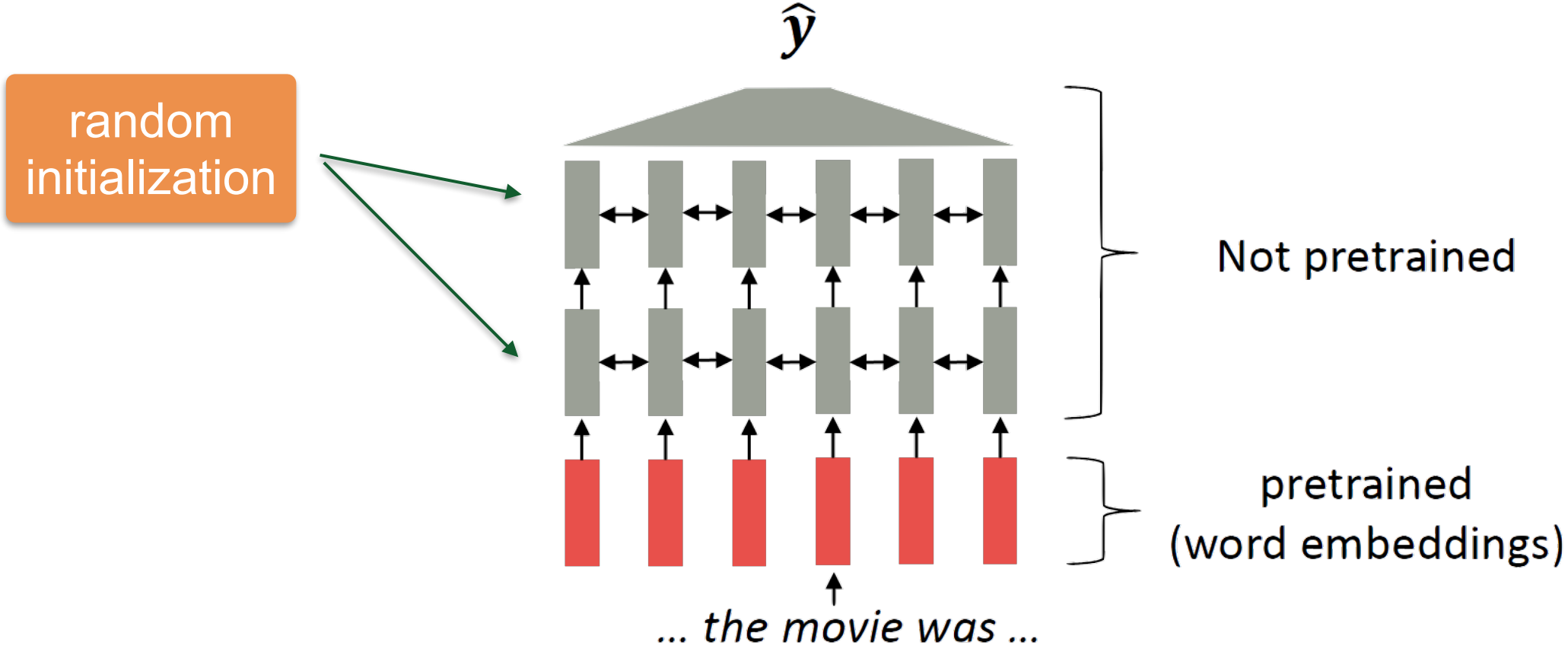
Natural Language Processing

- Unfortunately, there's no ImageNet equivalent :-(
- But there's plenty of text out there!
  - Wikipedia
  - Stackoverflow
  - …

BERT, FLAN-T5, llama, …

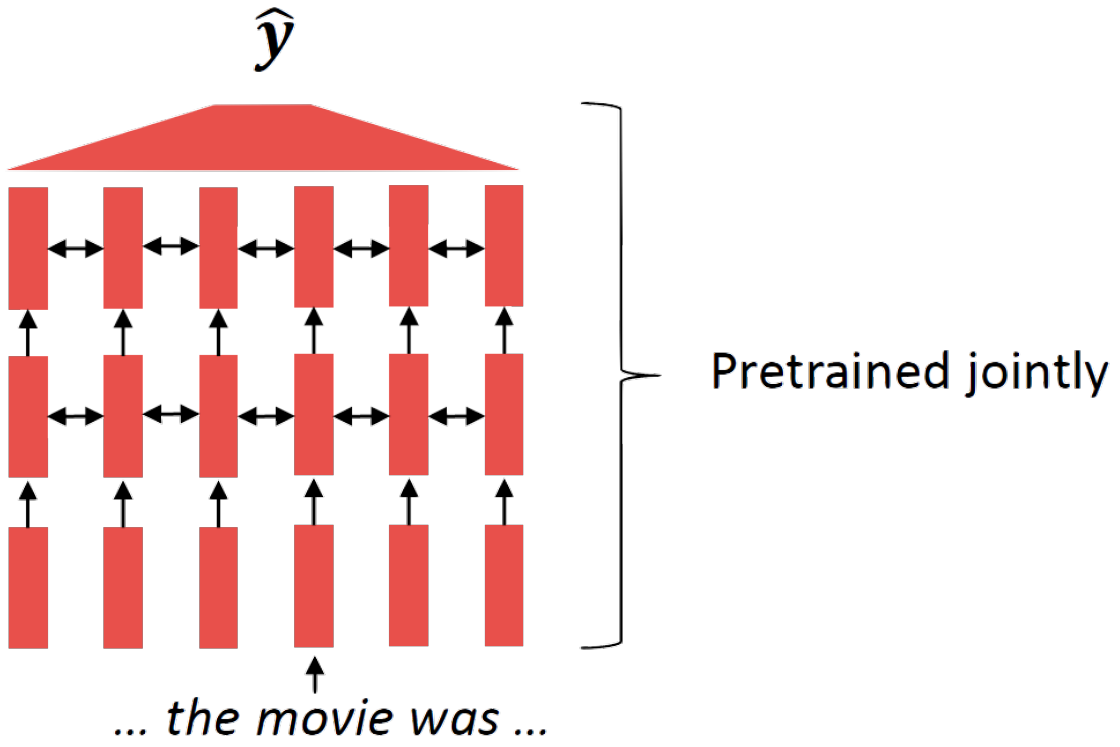
Natural Language Processing

- Up to 2017: start with pre-trained word embeddings (no context!)
- Learn the context in LSTM/Transformer while training to the task



Natural Language Processing

- Today:
  - (Almost) all parameters of the network initialized via pre-training
  - Pre-training methods hide parts of the input and train to reconstruct those parts
- With great success!
  - representations of language
  - parameter initializations for strong NLP models
  - probability distributions over language that we can sample from



Universität St.Gallen

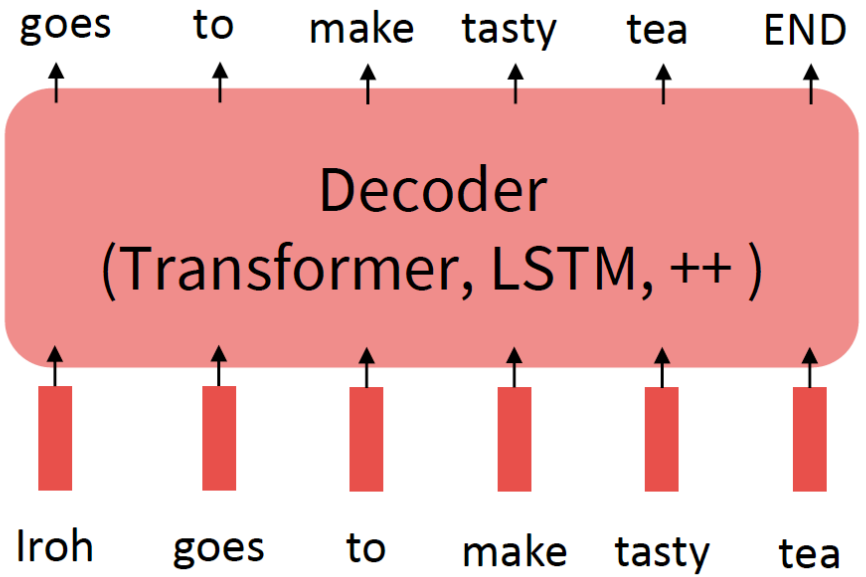
Learning from reconstructing the input

- University of St. Gallen is located in _____, Switzerland.
- I put ____ fork down on the table.
- The man crossed the street, checking for bikes over ___ shoulder.
- I went to the zoo to see the monkeys, giraffes and _____.
- I wasted two hours of my precious lifetime; the movie was _____.

The Pre-Training / Fine-tuning Paradigm

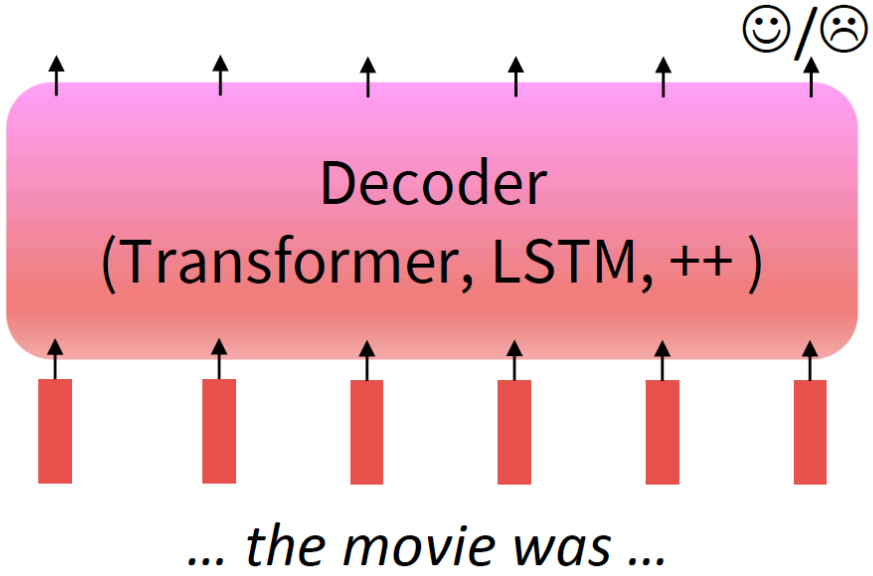
Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



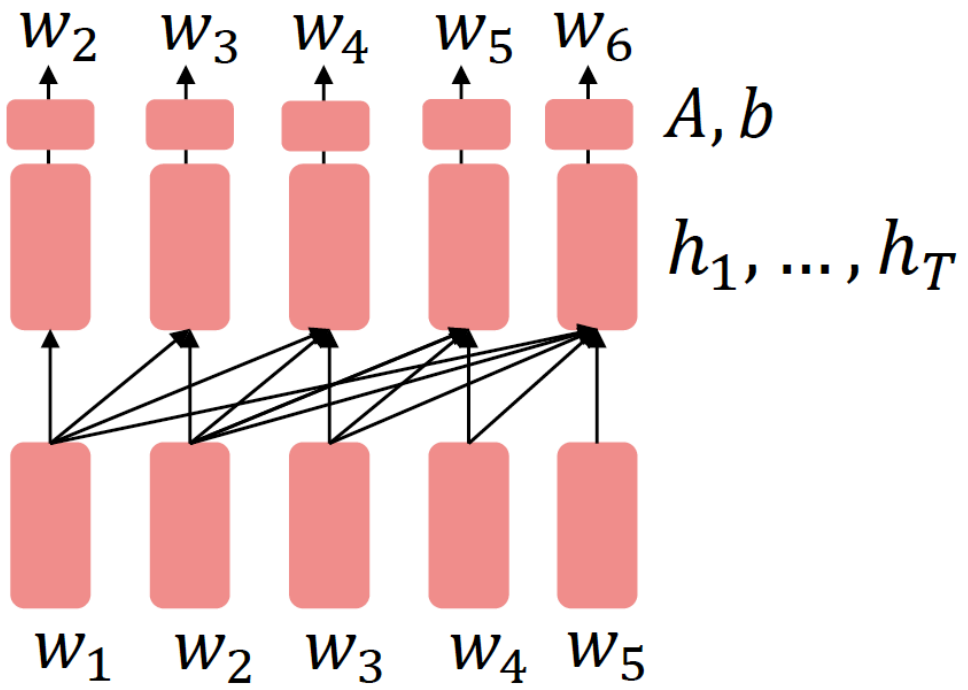
Step 2: Finetune (on your task)

Not many labels; adapt to the task!



Pre-training Decoders

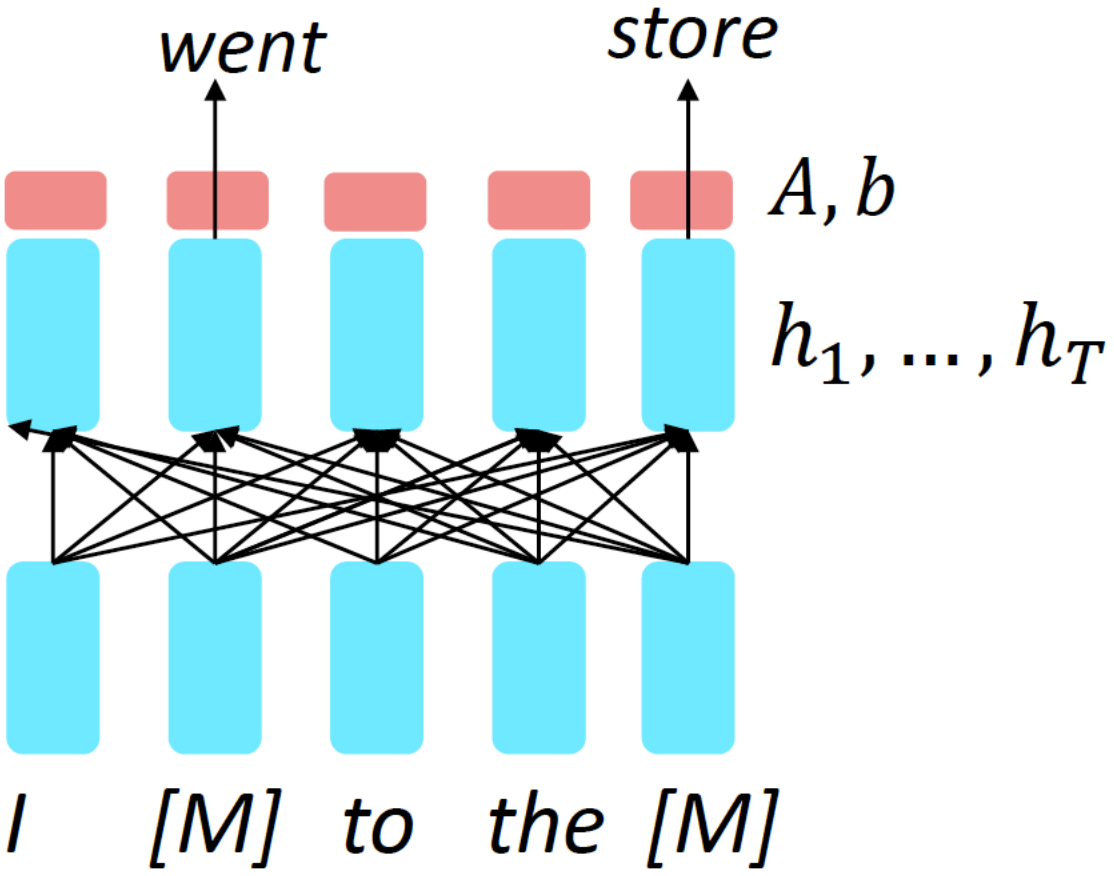
- Pre-train as language models
- …fine-tune on target sequences
- Helpful where input and output share the vocabulary
- Mask/ignore future!
- Dialog, summarization, …



[Note how the linear layer has been pretrained.]

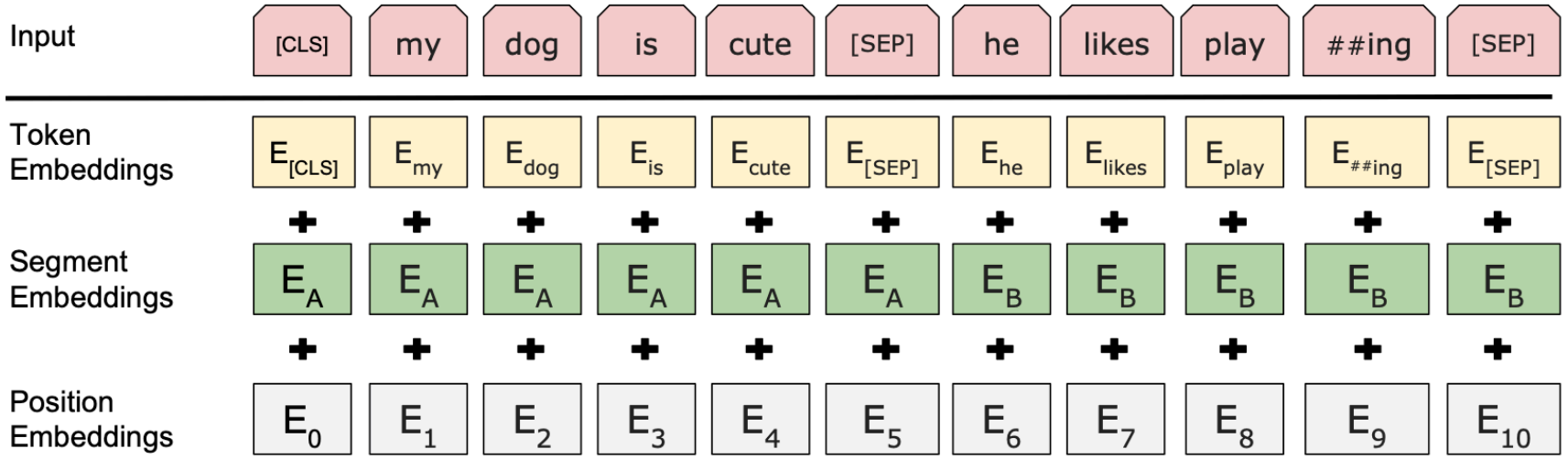
Pre-training Encoders

- Full input accessible — can condition on the future!
- Replace fraction of words with special [MASK] symbol
- ..train networks to reconstruct, predict those words



Ground-breaking Paper & Model

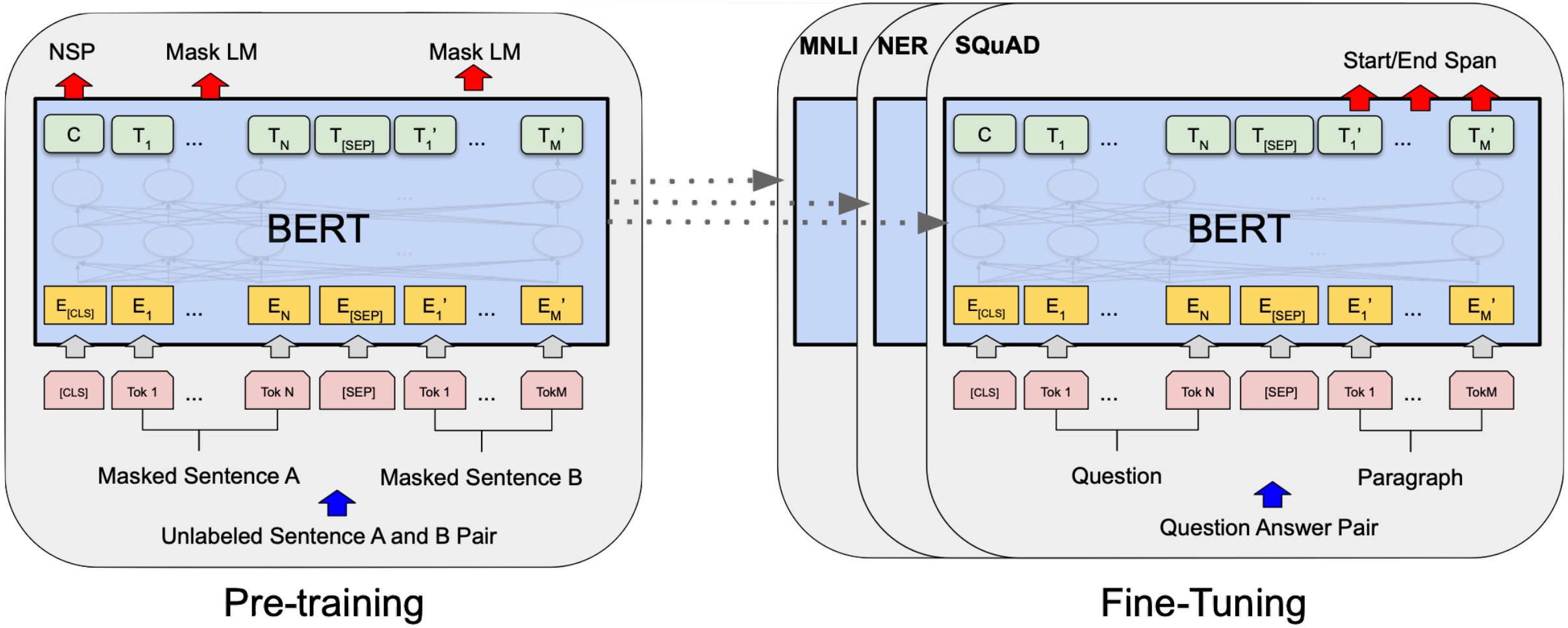
- Devlin et al., 2018: BERT: Bi-directional Encoder Representations from Transformer (https://arxiv.org/abs/1810.04805)
- During training: *masked* LM: randomly set some of the inputs to a placeholder (dropout)
- Fine-tune based on output corresponding to CLS



BERT's success

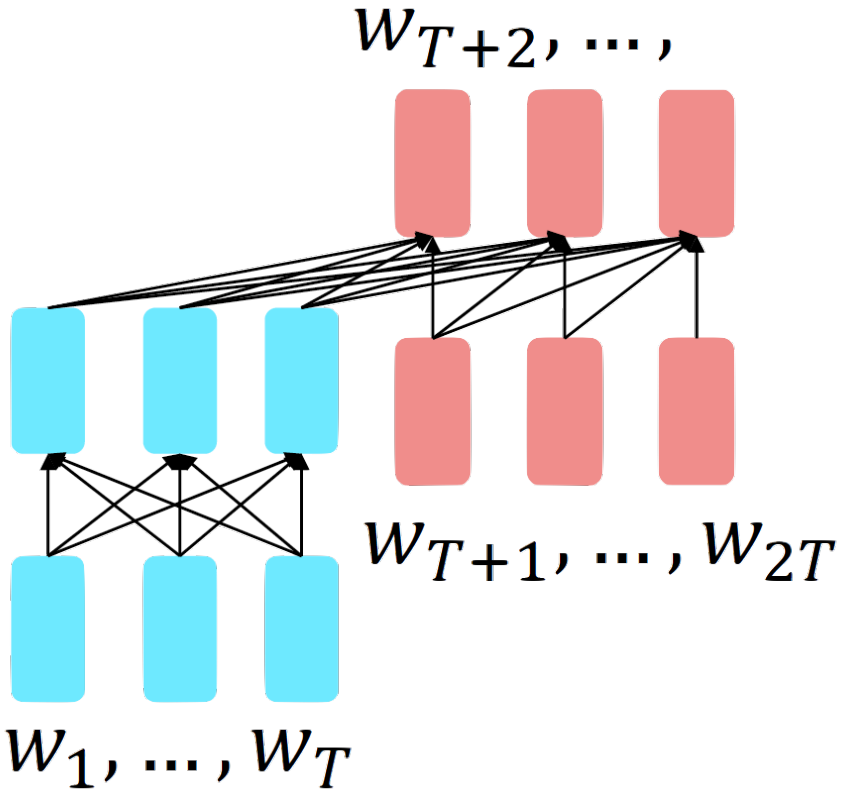
- Trained on huge corpus using masked LM
- In 2018, very same model achieved state-of-the-art values, for
  - Quora Question Pairs
  - CoLA (lingustic acceptability)
  - …and many mode.

BERT Transfer Learning



Pre-training Encoder-Decoders

- Raffel et al. 2018: “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer: Use sequence prefix to feed encoder”
  - https://arxiv.org/pdf/1910.10683.pdf
  - Use decoder part of the sequence to learn language modeling
  - Use span corruption instead of just masks



Universität St.Gallen

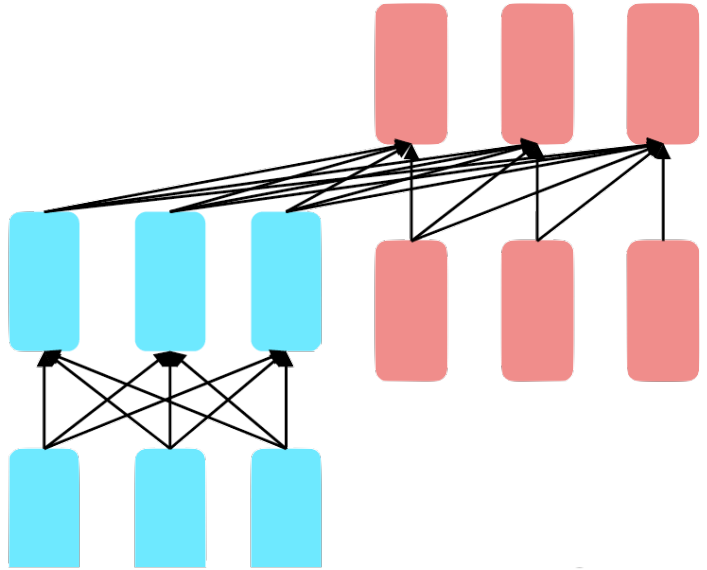
Span corruption

Original text

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

Targets

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



Inputs

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

"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."

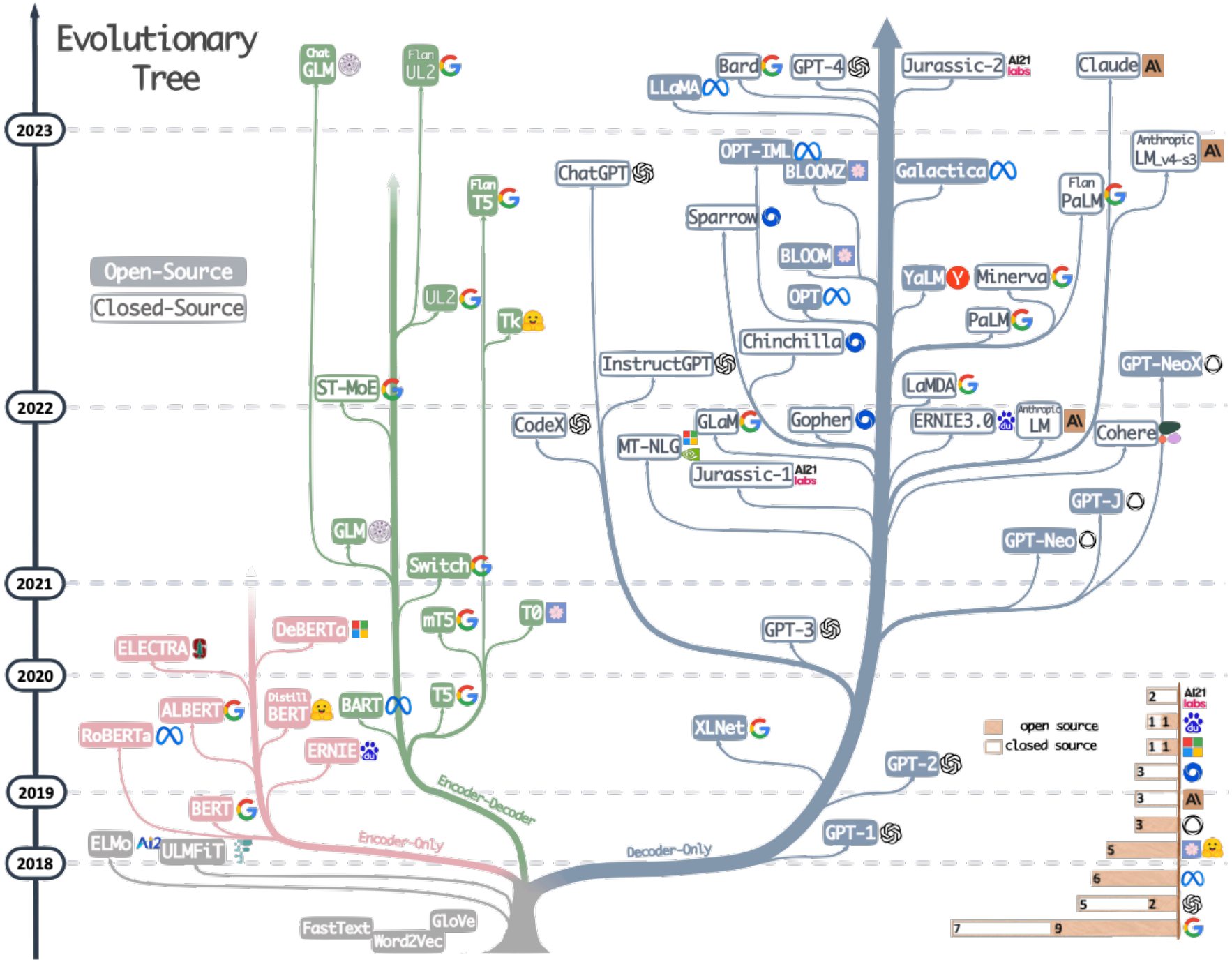
"not acceptable"

"3.8"

"six people hospitalized after a storm in attala county."

Universität St.Gallen

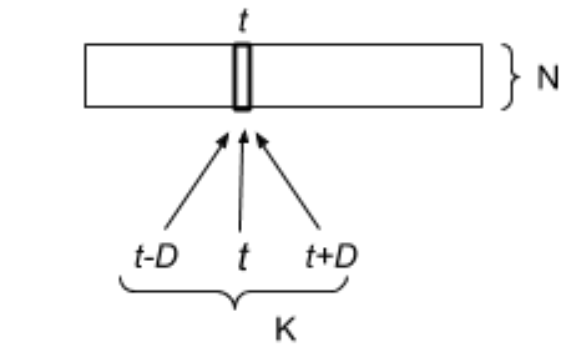
Too many LLMs to talk about…



Audio/Speech Processing

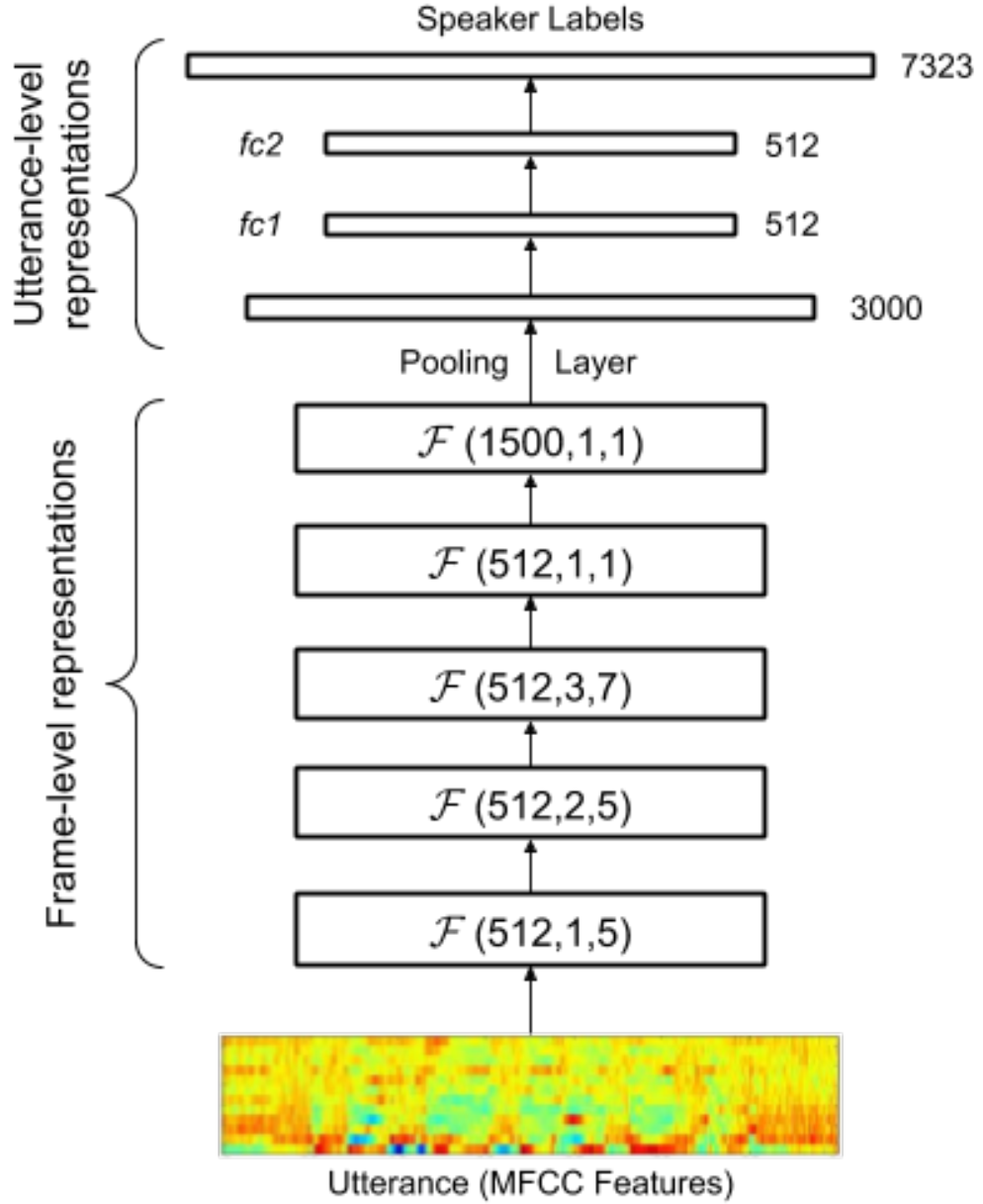
- Snyder et al., 2018: “X-Vectors: Robust DNN Embeddings for Speaker Recognition”.
  - https://www.danielpovey.com/files/2018_icassp_xvectors.pdf
  - Use TDNN features and global pooling
  - Classification of Age/Sex, medical conditions, etc.

X-Vectors



Time Delay Layer: $\mathcal{F}$ (N,D,K)

(a)



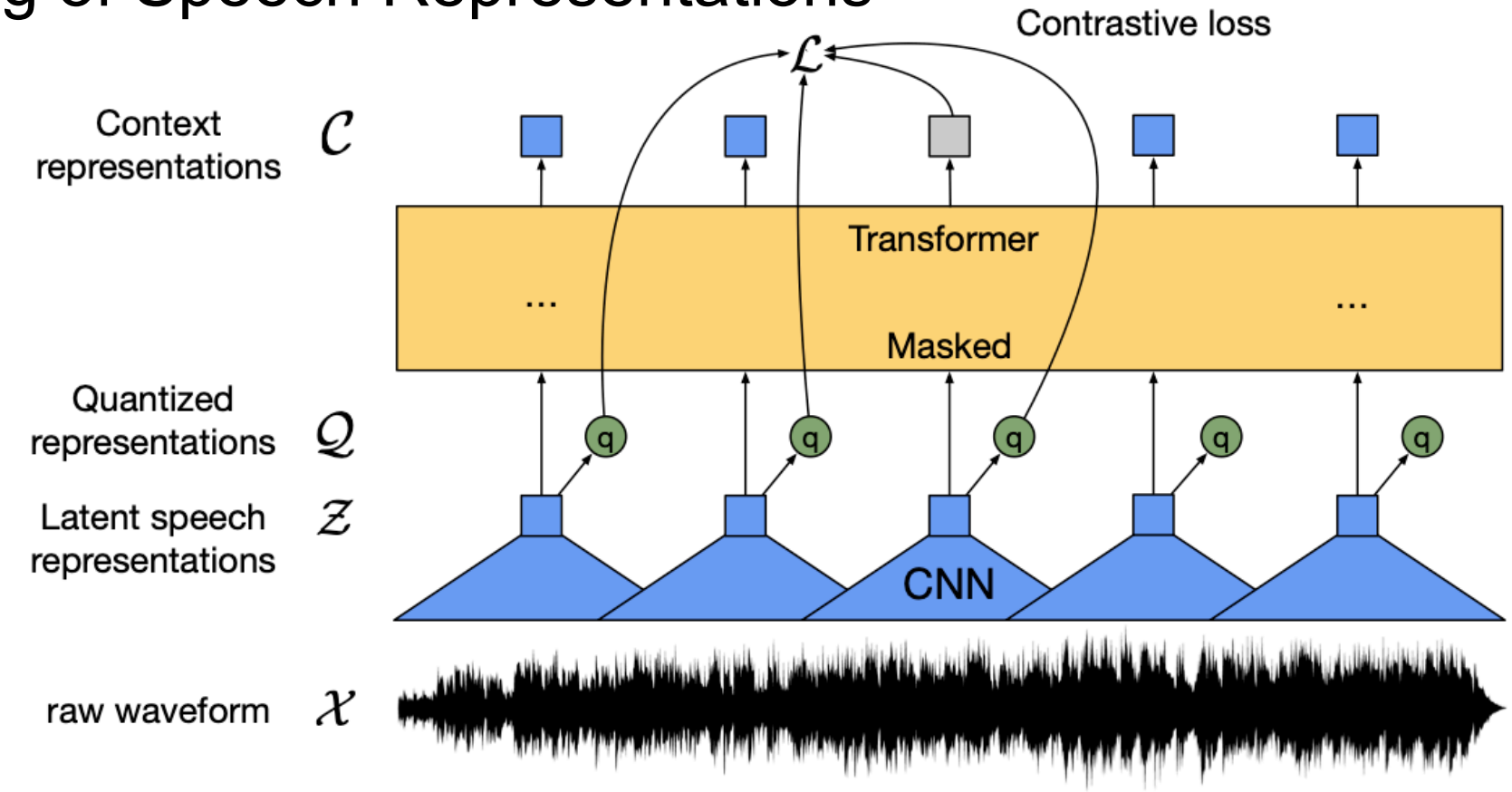
X-vector Model (Baseline)

(b)

https://arxiv.org/pdf/2006.11477.pdf

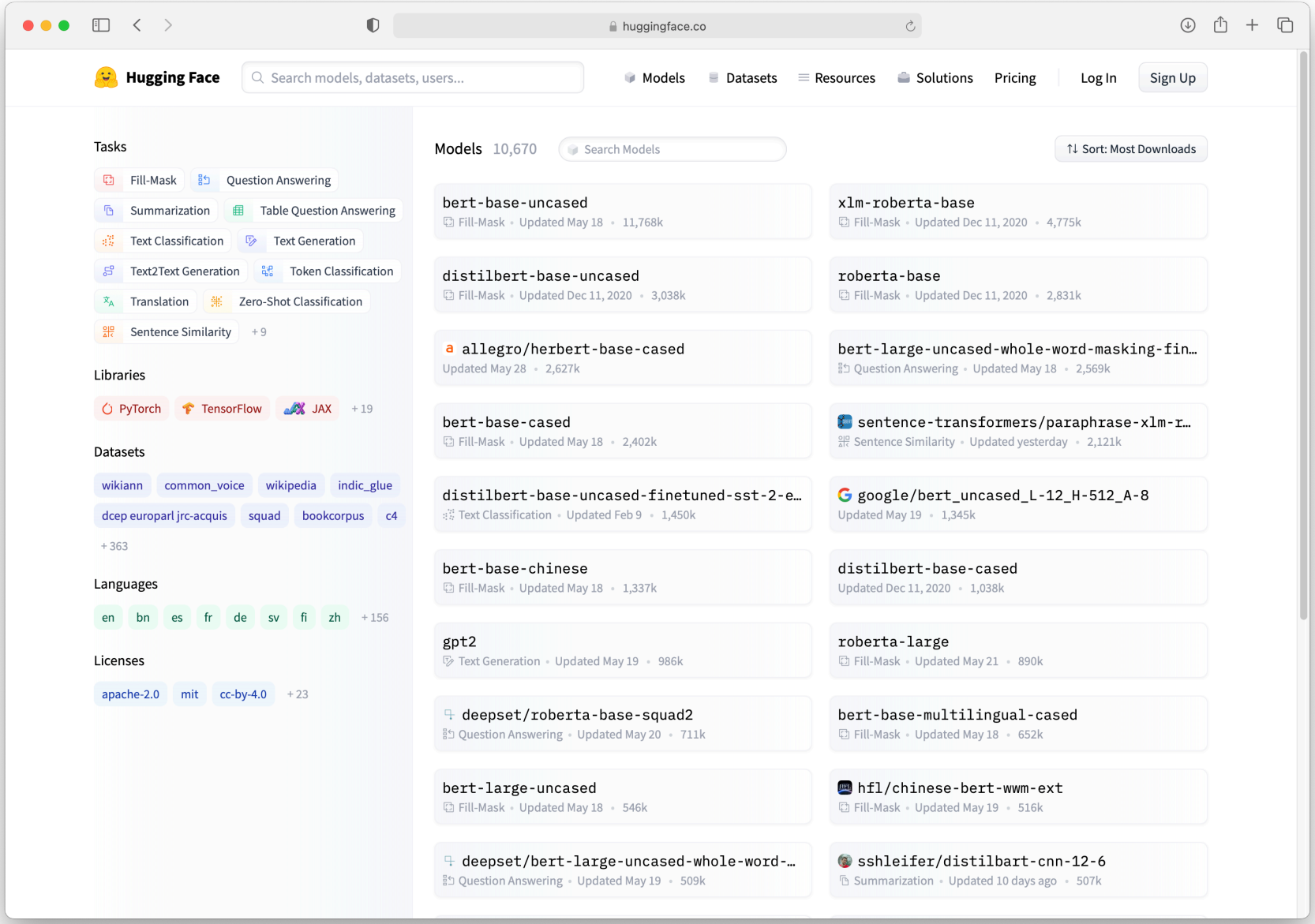
Audio/Speech Processing

- Baevski et al. 2020, “wav2vec2.0: A Framework for Self-Supervised Learning of Speech Representations”



Universität St.Gallen

Huggingface



Summary

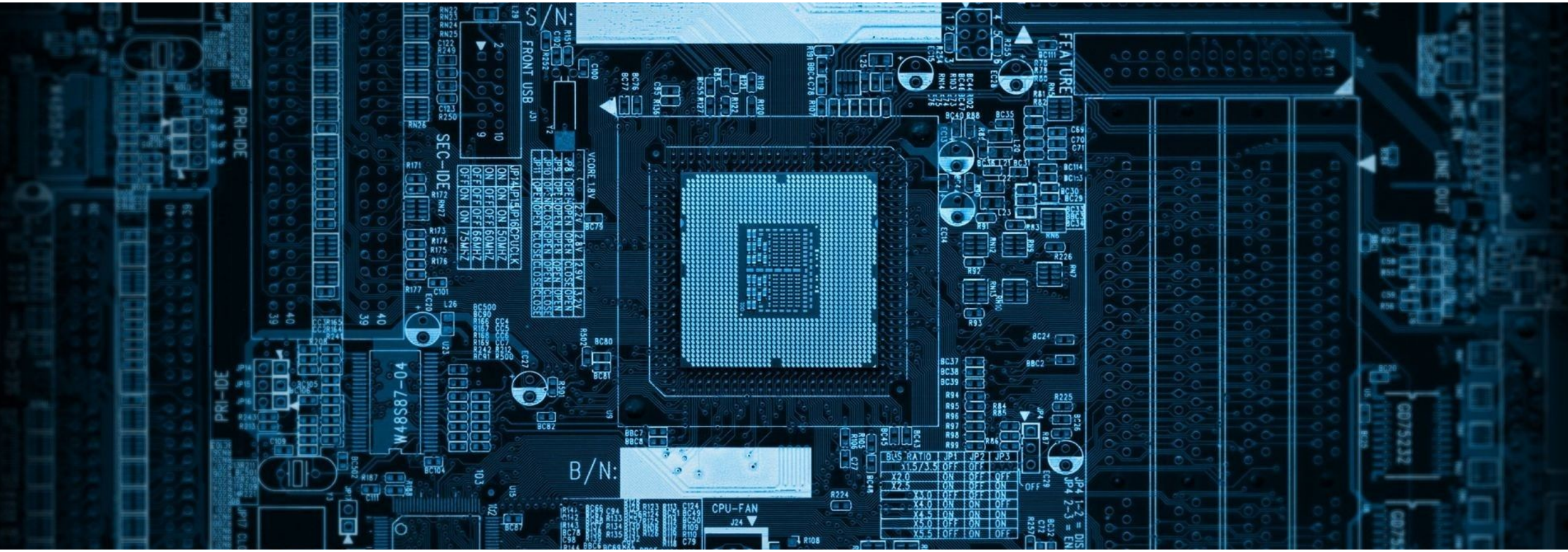
- We can **learn representations** from large corpora
  - using dedicated labels (eg. ImageNet, VoxCeleb)
  - using *surrogate* labels, ie. by reconstructing masked inputs (c4)
- For (most) of the models we discussed, **fine-tuning to the actual task only involves training a single final layer** (or maybe a few iterations of BP)
- Auto-encoder bottle-necks are also (basic) representations
  - typically much smaller/simpler architectures
  - resulting representations require typically much more sophisticated architectures for the actual taks
- **LLMs are tempting**, but should only be used if approximate answers are ok!

Universität St.Gallen

Recommendations

- If you can avoid it, do not train the discussed models yourself
  - …it's complicated, time-consuming and expensive
  - Download base (and adapted) models from Huggingface
- For image classification tasks, use **ResNet** or **ViT** models
- For NLP tasks, use transformer models such as BERT or FLAN-T5
- For audio/speech classification tasks regarding speaker properties, use **x-vectors**
- For general audio/speech classification tasks, use **wav2vec2**

Universität St.Gallen



Questions?