Learning to Learn: Hands-on Tutorial on Using and Improving Few-Shot Language Models

Michal Štefánik

Nikola Groverová

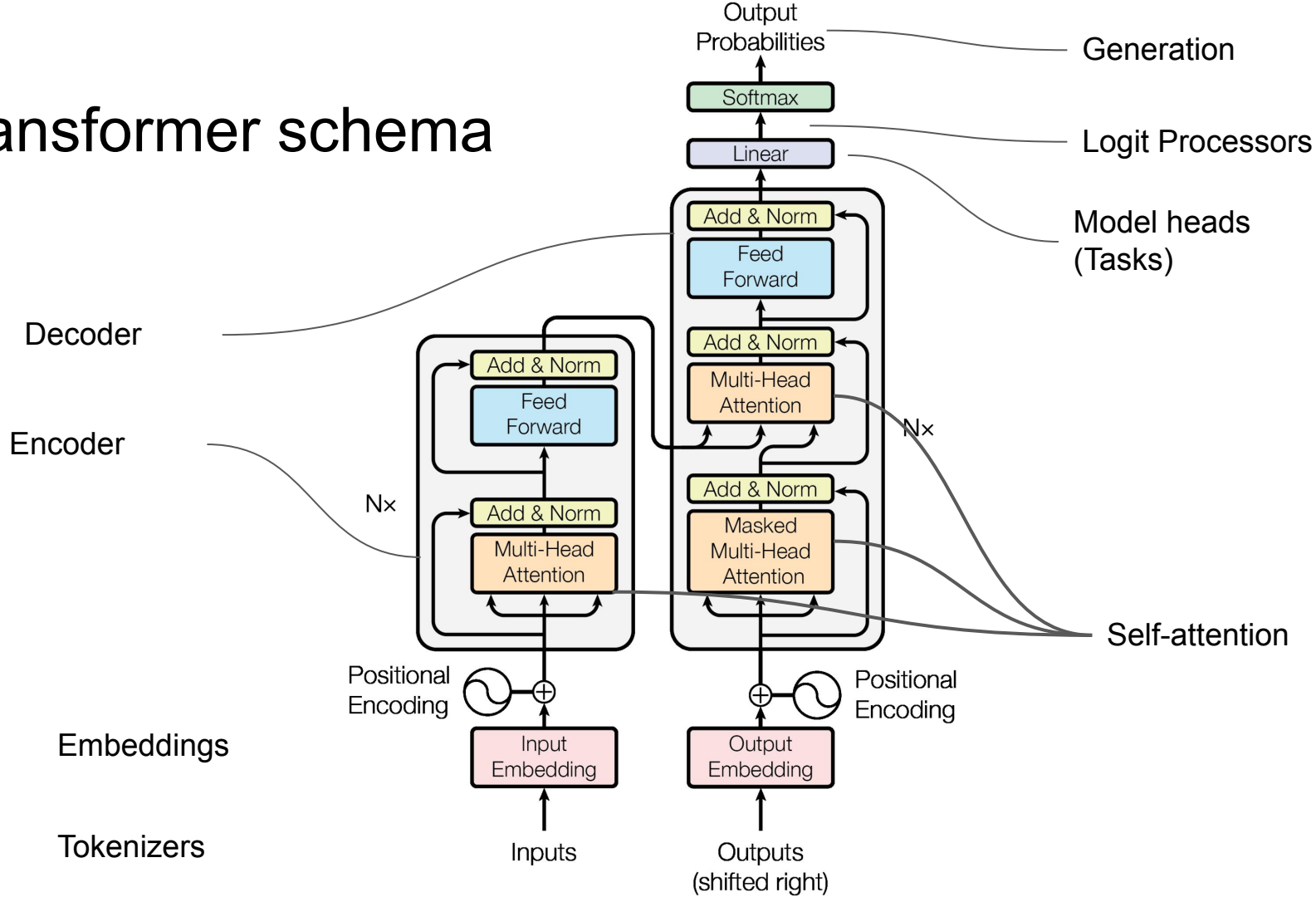
Machine Learning Prague

Natural Language Processing Team | gauss algorithmic

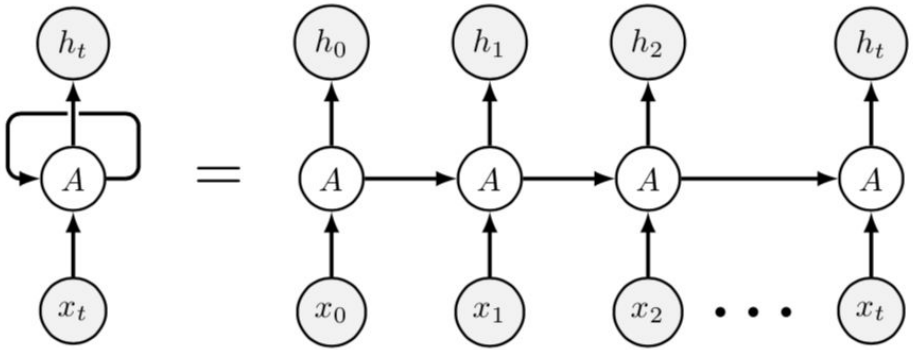
Outline

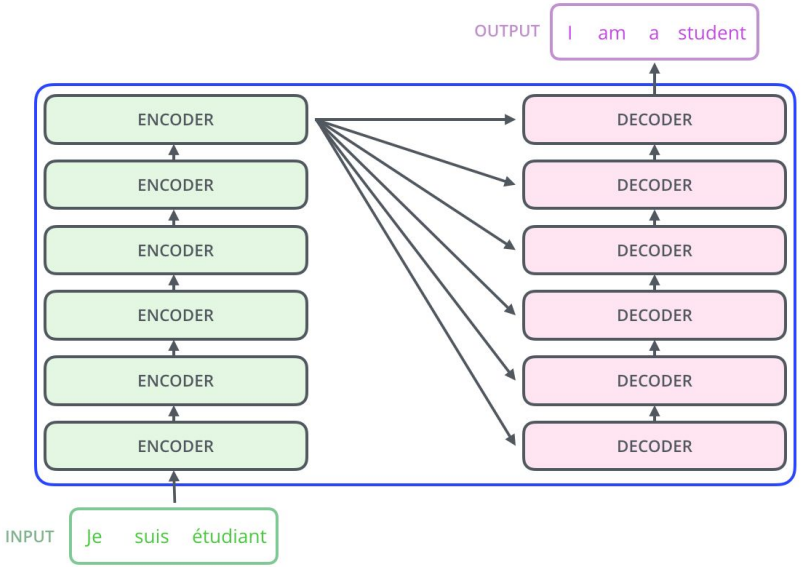
1. **Introduction to using Transformers:**
   a. **Pre-training & Fine-tuning**
   b. **Text generation:** Decoding strategies, Logit processors + **Hands-on session**
2. **In-context learning:**
   a. **LLM or Specialized Models**
   b. **Few-shot in-context learning**
   c. **Prompt design + Hands-on session**
3. **Training few-shot in-context learners**:
   a. **Overview of existing models + Hands-on session**
   b. **Theory behind:** What makes language models in-context learners?
4. **Hands-on on training few-shot ICL model:**
   a. **Training pipeline for training generative transformers:** low & high-level
   b. **Hands-on session**

Transformer schema



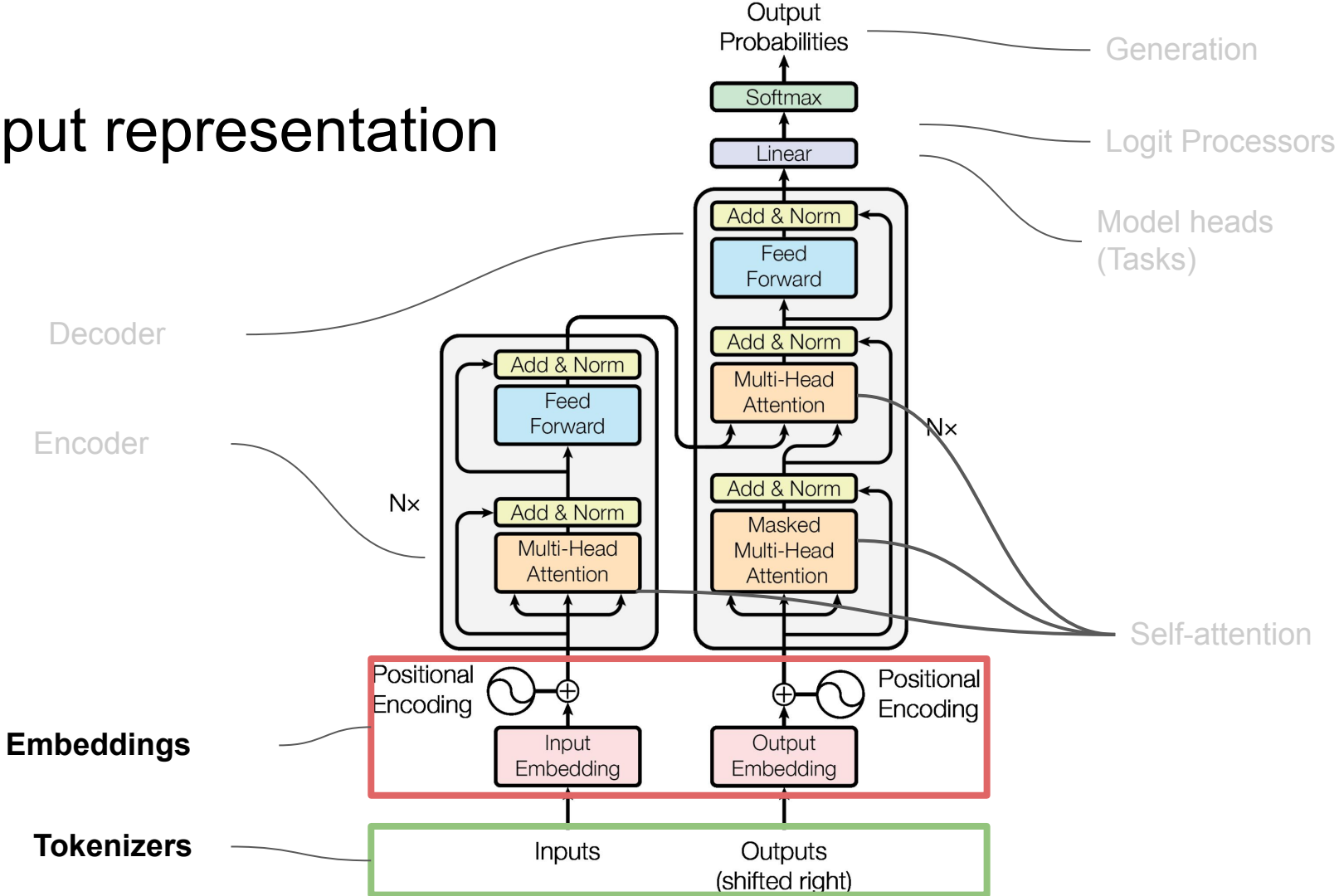
Introduction to Transformers: RNN vs Transformers





[1] https://tex.stackexchange.com/a/494148
[2] Jay Alammar: The Illustrated Transformer

Input representation



Introduction to Transformers: Input representations

Word embedding of input sequence size (9, 512)

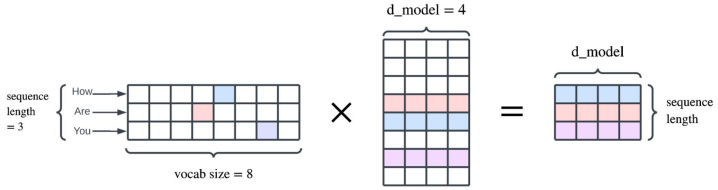
Word Embeddings + Positional Embeddings

1563	31	7	1350	81	445	6892	5	1

_Let	'	s	_talk	_about	_N	LP	.	</s>

Let's talk about NLP.

Word embeddings



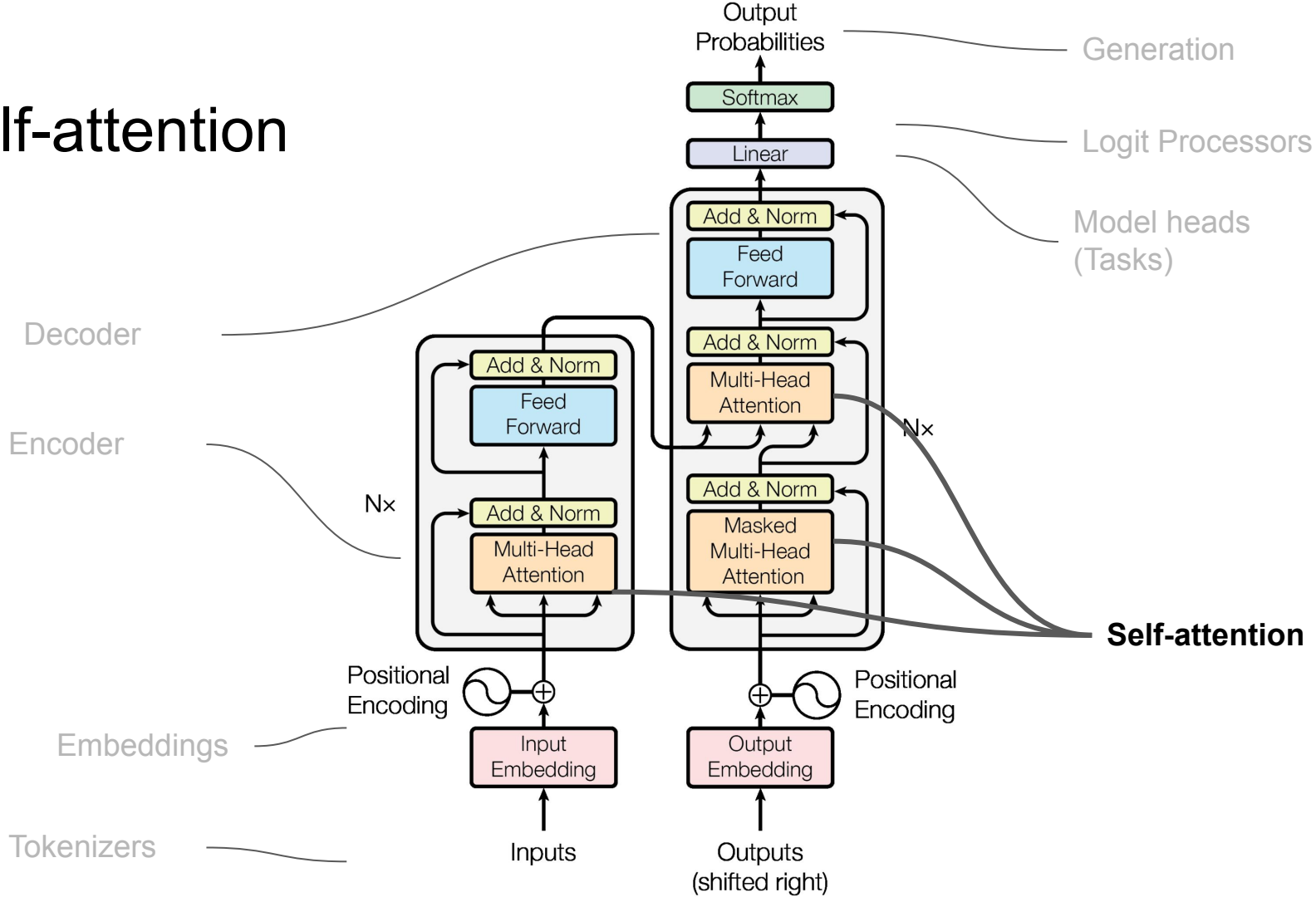
$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{model}}\right)$$

Positional embeddings

[1] https://blog.deepgram.com/visualizing-and-explaining-transformer-models-from-the-ground-up/

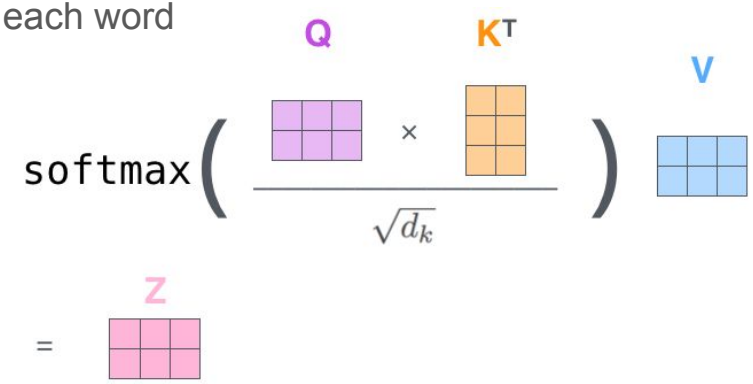
Self-attention



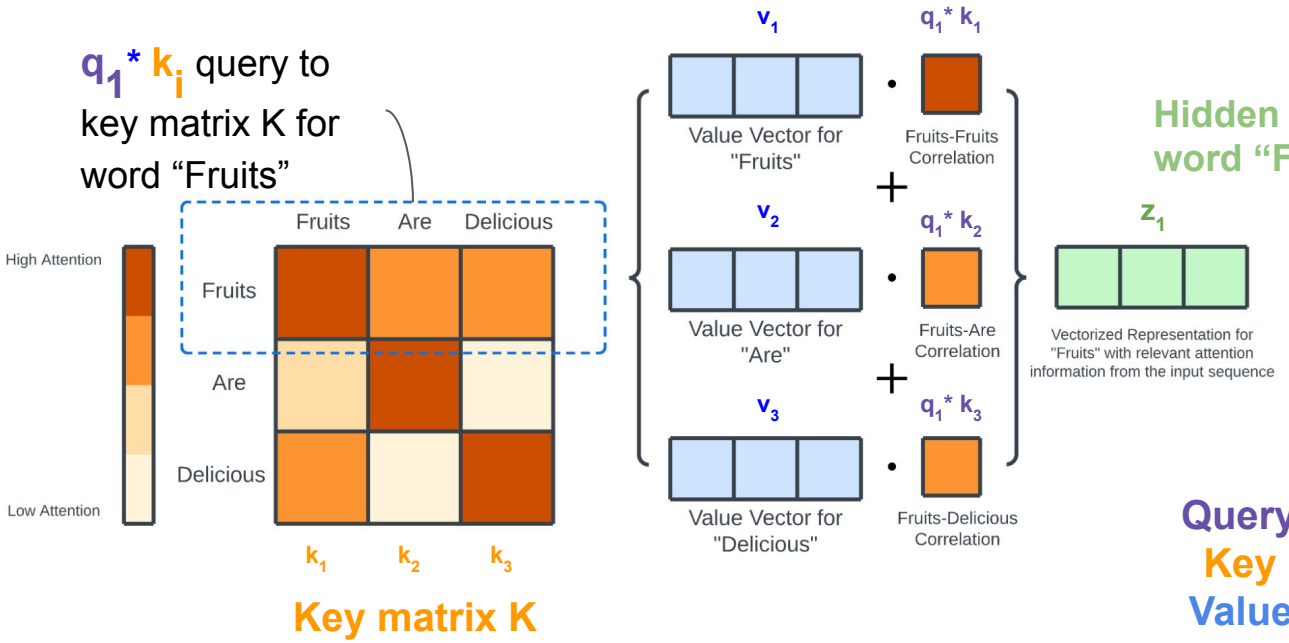
Introduction to Transformers: Self-attention

Query matrix Q
Key matrix K
Value matrix V

- **Q**, **K** and **V** are all weighted representations of the input
  - Each matrix row represents one token/word
  - multiplication Q x KT - attention score between all words
  - Division by a constant - for stability purposes
  - Softmax - creates a probability (each row sum up to 1)
  - Multiplying by V - apply attention score to each word



Introduction to Transformers: Self-attention



Query matrix Q
Key matrix K
Value matrix V

[1] https://blog.deepgram.com/visualizing-and-explaining-transformer-models-from-the-ground-up/

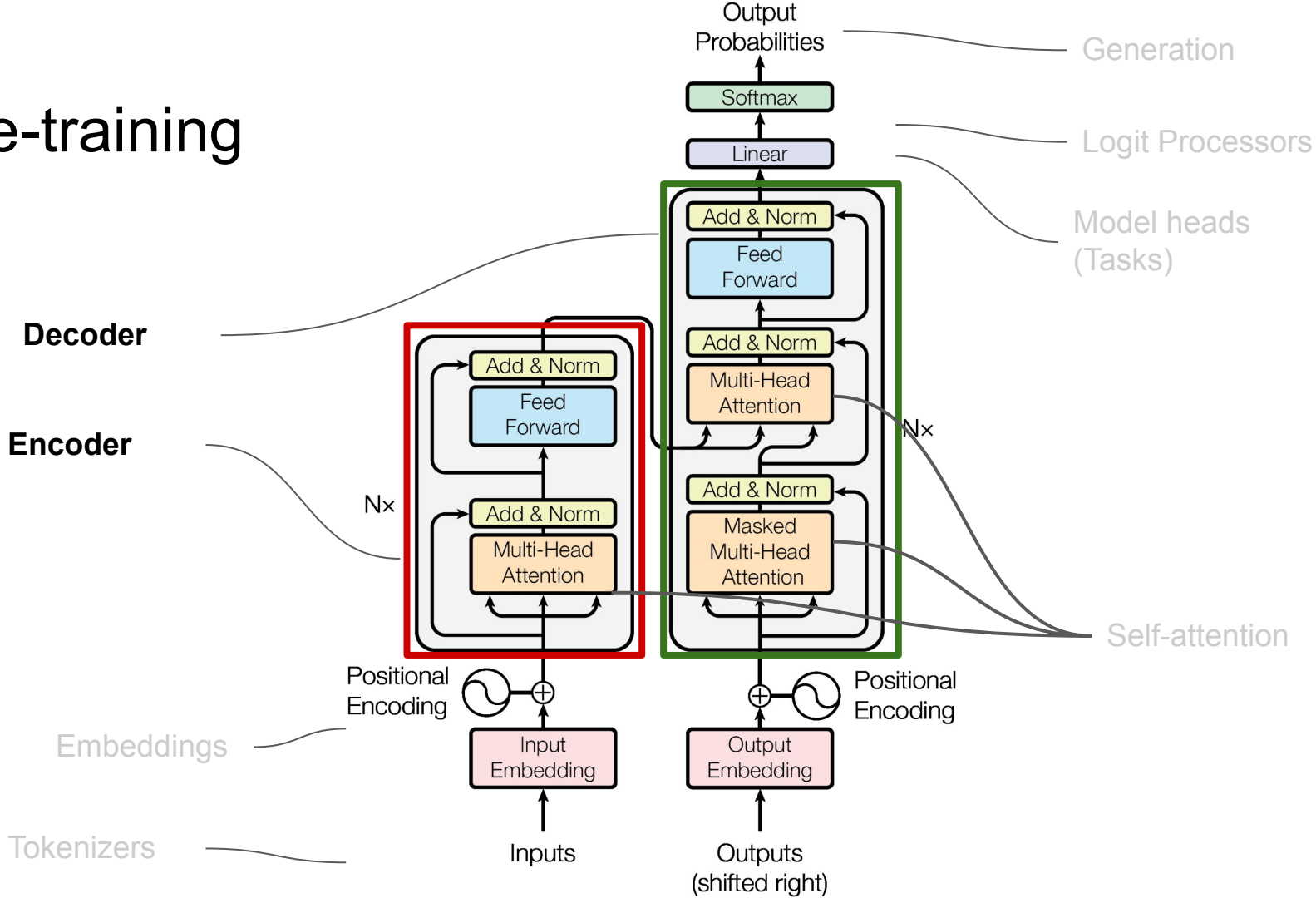
Introduction to Transformers: Multi-head attention

- Each head has its own matrices
- In each head attention score is computed independently
- The results for each head are concatenated back together
- Allowing multiple attention score to be computed we allow for a richer representation.

- When computing the cross-attention between an Encoder and Decoder
  - Q is from the target/output sequence
  - K and V come from the encoder hidden states

[1] Jay Alammar: The Illustrated Transformer

Pre-training



Why do we Pre-train?

- Pre-training a model allows the random initialized model to get preliminary knowledge of the language
- **Unsupervised training**
- Given sufficient capacity, models can be **pre-trained** on a related task, to later **learn** the **end task faster** and **more accurately**
- Pre-training and fine-tuning objective(s) can **differ**: **pre-training** can use **vastly larger volumes of data**

Pre-training: Masked Language Modelling (MLM)

"Odberateľ [MASK] Monika Pataiová, Štúrova 44/B, 90301 Senec, Slovensko"

"Odberateľ: [MASK] Pataiová, Štúrova 44/B, 90301 Senec, Slovensko"

"Odberateľ: Monika [MASK], Štúrova 44/B, 90301 Senec, Slovensko"

```
prob: 99.9023%, token: :
prob: 0.0519%, token: -
prob: 0.0257%, token: je
prob: 0.0105%, token: ,
prob: 0.0011%, token: :

prob: 11.1633%, token: Daniela
prob: 6.7139%, token: Tatiana
prob: 6.1157%, token: Martina
prob: 5.3558%, token: Veronika
prob: 5.3131%, token: Monika

prob: 7.9041%, token: Martina
prob: 7.9041%, token: Horváth
prob: 6.9214%, token: Nová
prob: 3.4271%, token: Tóth
prob: 2.7542%, token: Nagy
```

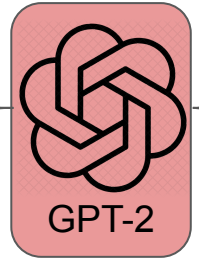
[1] Dávid Meluš: InvoiceBERT Language model.

Pre-training: Causal Language Modelling (CLM)

"Odberateľ [MASK]"

"Odberateľ: [MASK]"

"Odberateľ: Monika [MASK]"

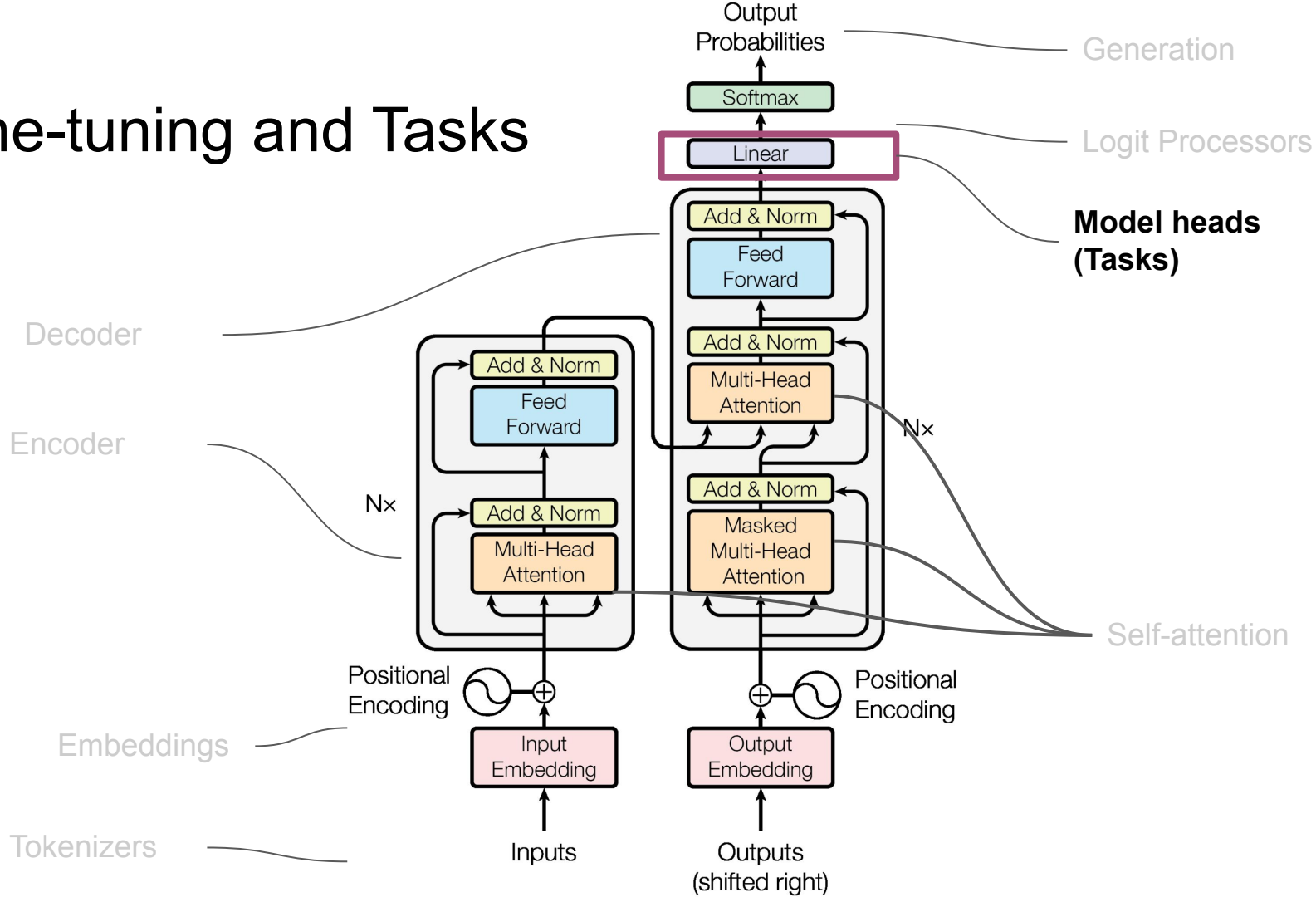


prob: 99.9023%, token: :
prob: 0.0519%, token: -
prob: 0.0257%, token: je
prob: 0.0105%, token: ,
prob: 0.0011%, token: :

prob: 11.1633%, token: Daniela
prob: 6.7139%, token: Tatiana
prob: 6.1157%, token: Martina
prob: 5.3558%, token: Veronika
prob: 5.3131%, token: Monika

prob: 7.9041%, token: Martina
prob: 7.9041%, token: Horváth
prob: 6.9214%, token: Nová
prob: 3.4271%, token: Tóth
prob: 2.7542%, token: Nagy

Fine-tuning and Tasks



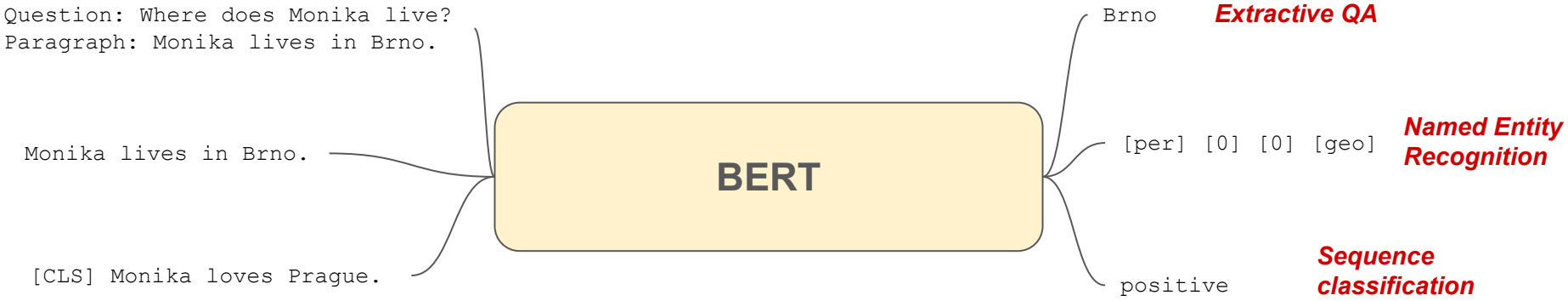
Fine-tuning

- Adding a linear layer (“head”) which processes the last hidden states
- Training the the weights in the linear layer for a specific task

Tasks and Transformer schema

- We can use the Encoder and Decoder either on their own or together
- When thinking of a task you want to solve, think of the schema and pre-training objective which most benefits your use-case

Tasks: Token classification tasks

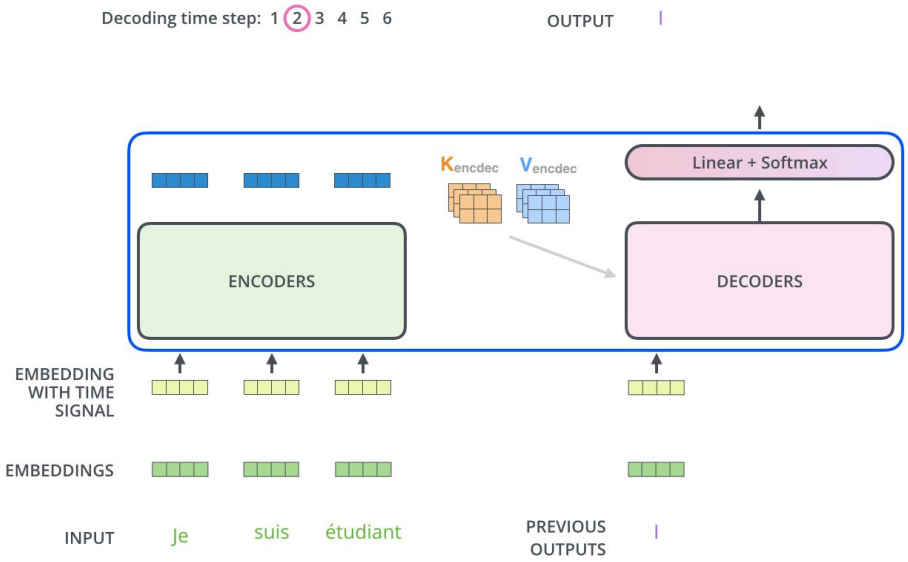


- Models with Encoders
- Pre-training on tasks similar to MLM

[1] Jay Allamar: A Visual Guide to Using BERT for the First Time

Tasks: Text generation tasks

- Text generation
- Sequence to sequence



- Models with Decoders
- Pre-training on tasks similar to CLM

Image source:
[1] Jay Alammar: [The Illustrated Transformer]

Decoder or Encoder-Decoder for text generation?

Decoder-only model

- Smaller models
- Can be trained on much more data (unsupervised)

Encoder - Decoder model

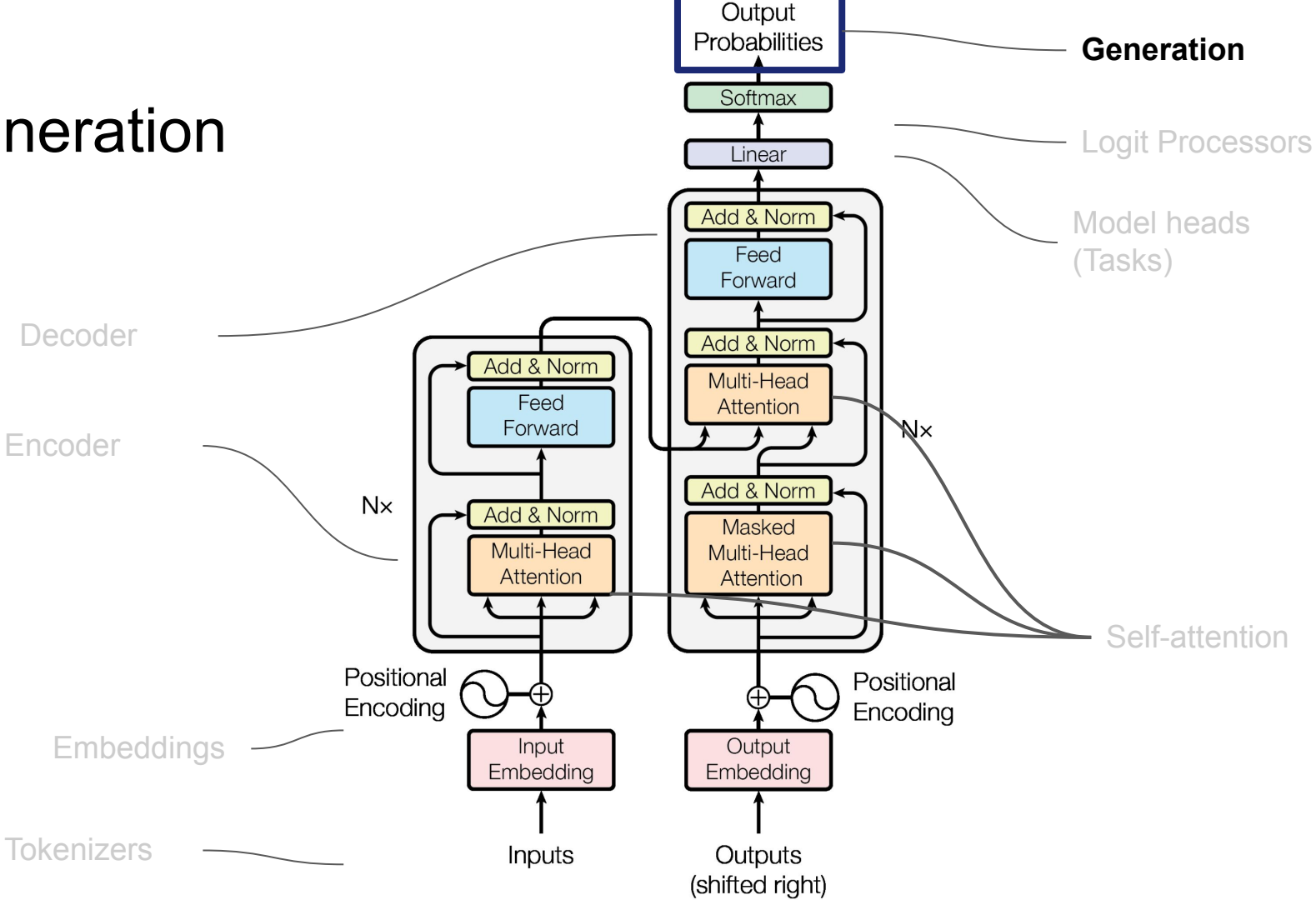
- Information about the input from encoder hidden states
- Fine-tuning on labeled data

The model should be exposed to large amount of data. Diverge from input is not a big deal:
Decoder

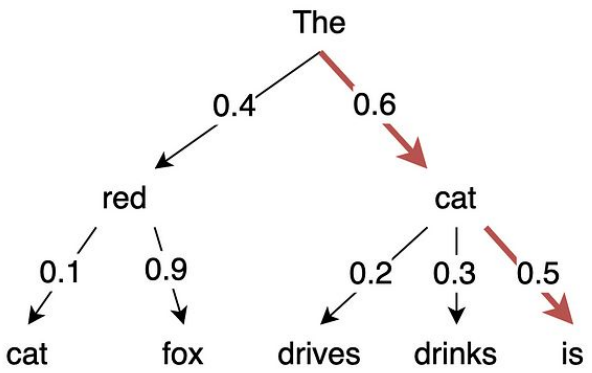
The model should not diverge from input:
Encoder-Decoder

[1] Zihao Fu1 et al.: Decoder-Only or Encoder-Decoder? Interpreting Language Model as a Regularized Encoder-Decoder

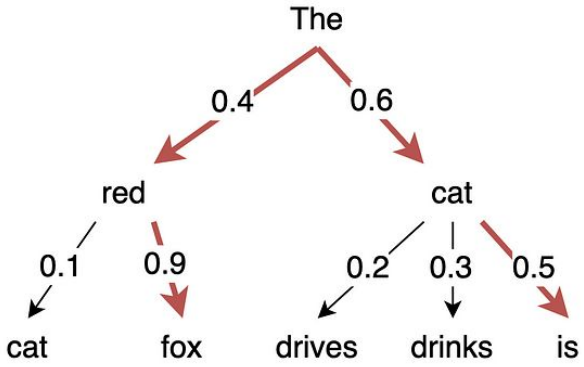
Generation


Output Probabilities
Generation
Softmax
Logit Processors
Linear
Model heads (Tasks)
Add & Norm
Feed Forward
Decoder
Add & Norm
Multi-Head Attention
Nx
Add & Norm
Feed Forward
Encoder
Nx
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Self-attention
Positional Encoding
Positional Encoding
Embeddings
Input Embedding
Output Embedding
Tokenizers
Inputs
Outputs (shifted right)

Decoding strategies: greedy & beam search

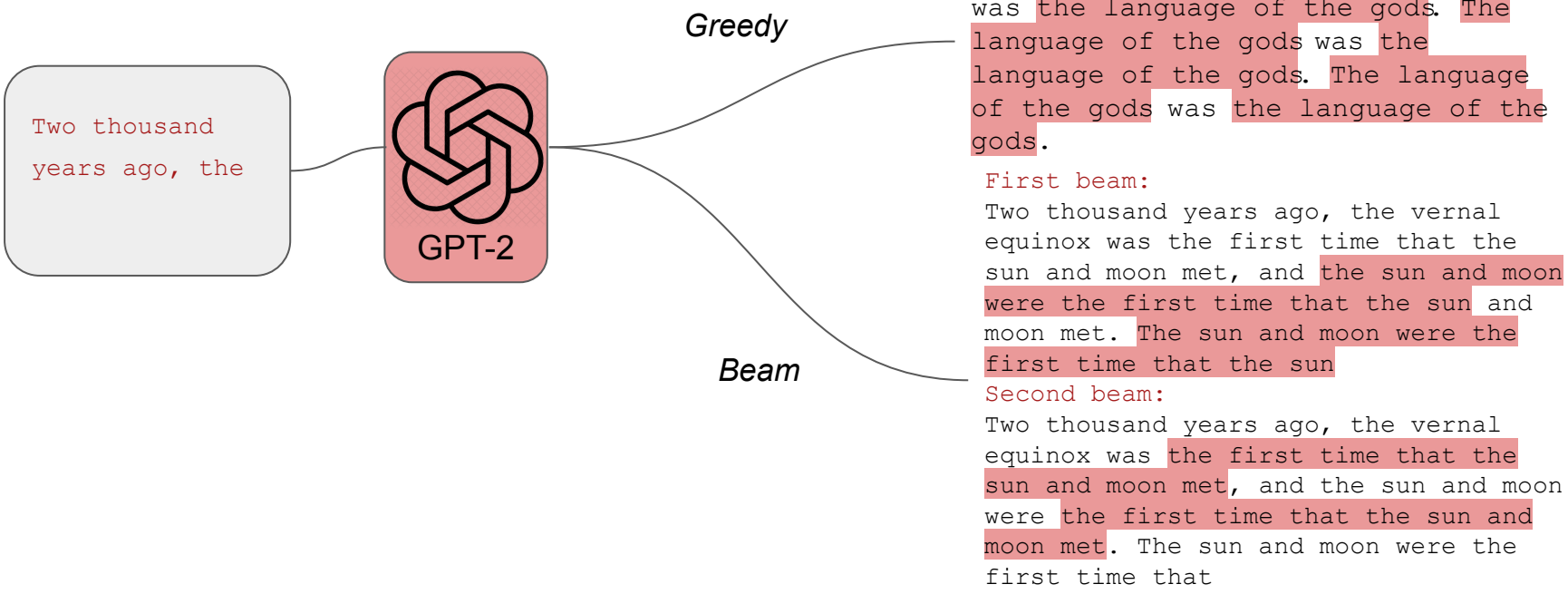


0.6 * 0.5 = 0.3



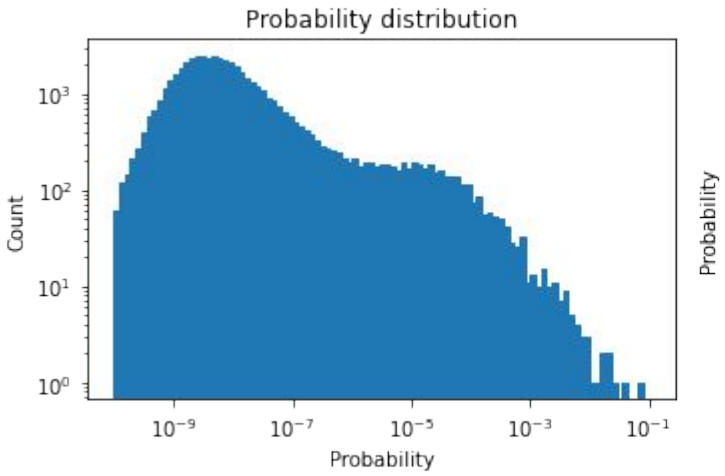
0.4 * 0.9 = 0.36 **0.6 * 0.5 = 0.3**

Decoding strategies: greedy & beam search

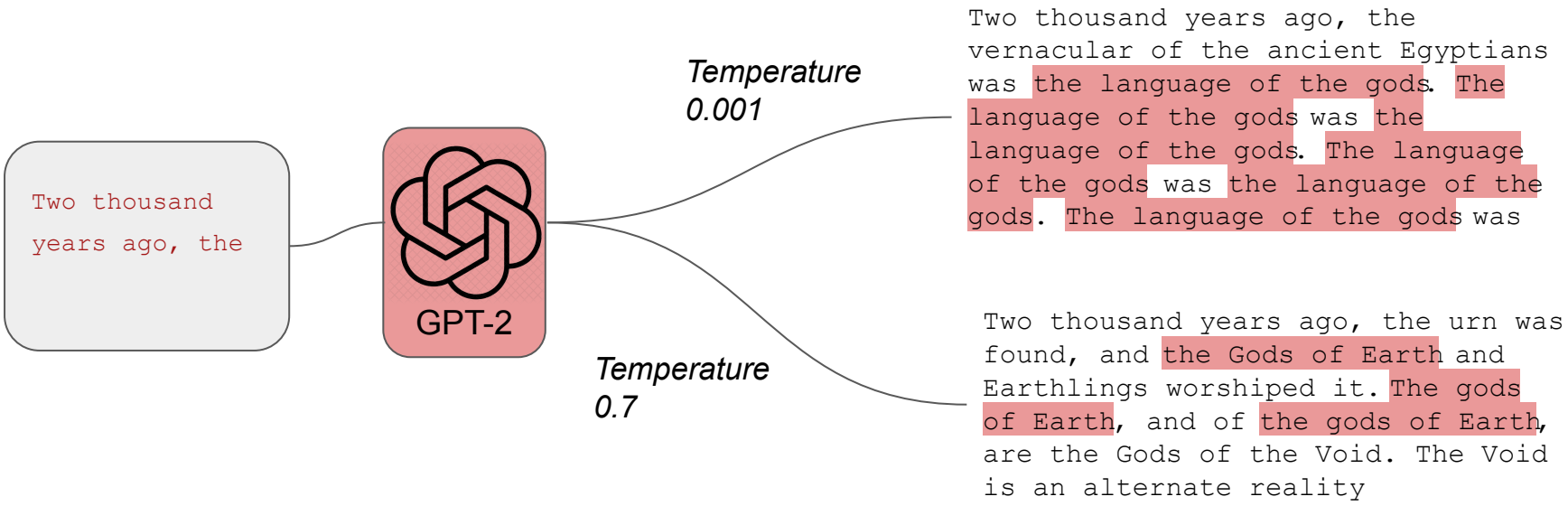


Decoding strategies: Sampling

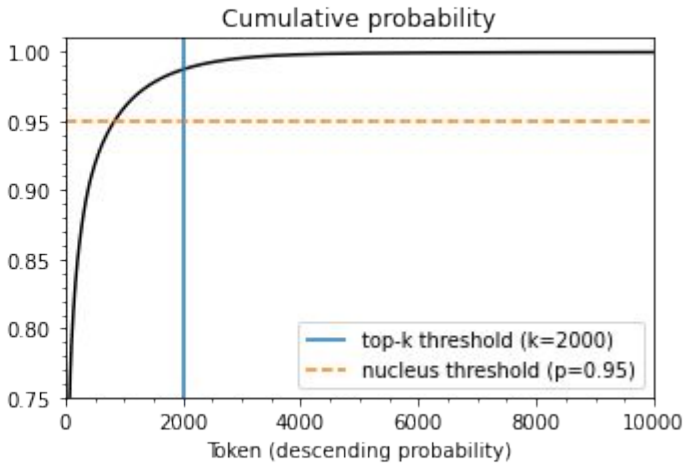
- **NOT DETERMINISTIC**
- Random sampling from a probability distribution based on previously generated tokens
- Randomness adjustments with temperature
- Thresholds on considered samples:
  - Top-k sampling
  - Top-p sampling

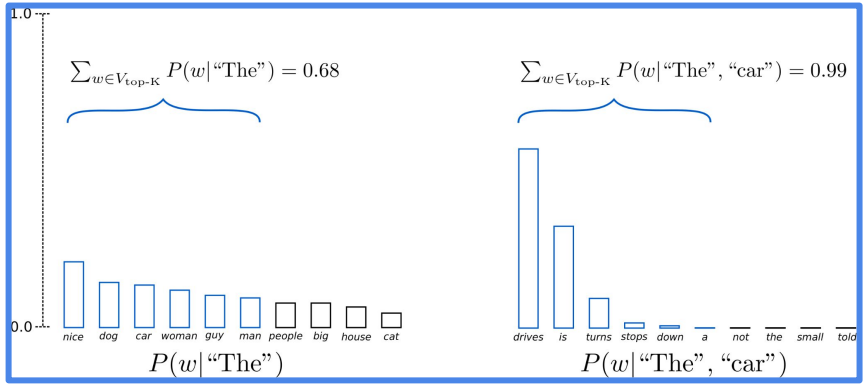


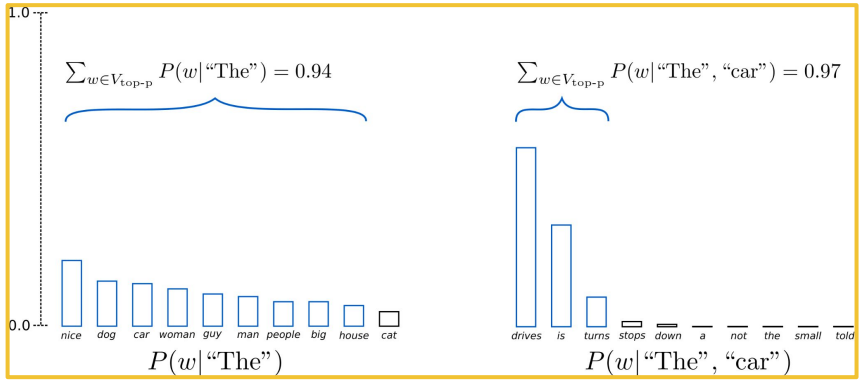
Decoding strategies: Sampling - Temperature



Decoding strategies: Sampling - Thresholds

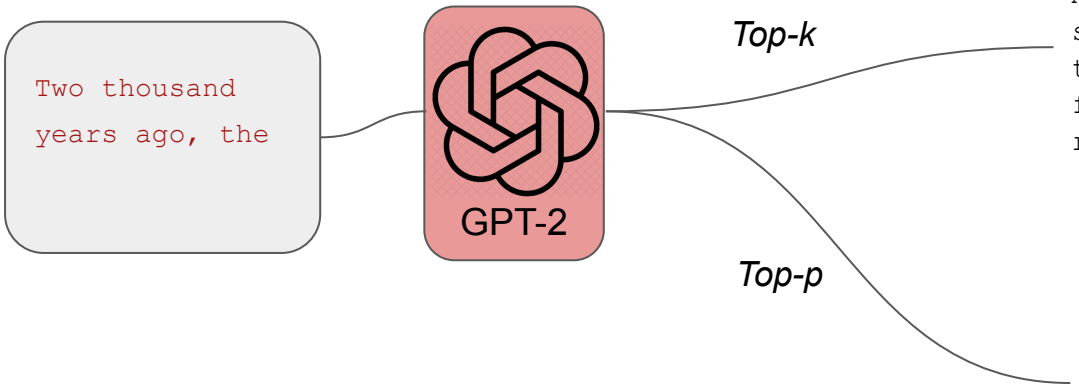






[1] Hugging Face: [How to generate]

Decoding strategies: Sampling - Thresholds



Two thousand years ago, the ichthyologists and palaeobiologists made a breakthrough, by using a method that mimicked the behavior of some animal dinosaurs and allowed them to see how long fossils were formed in a natural system. Their result is

Two thousand years ago, the vernacular (roughly 30 th century) of English as its tongue was English and English. This was partly due to the innovations of English invented by Robert Ellyner of Sud Cheap au Coeur who perfected the

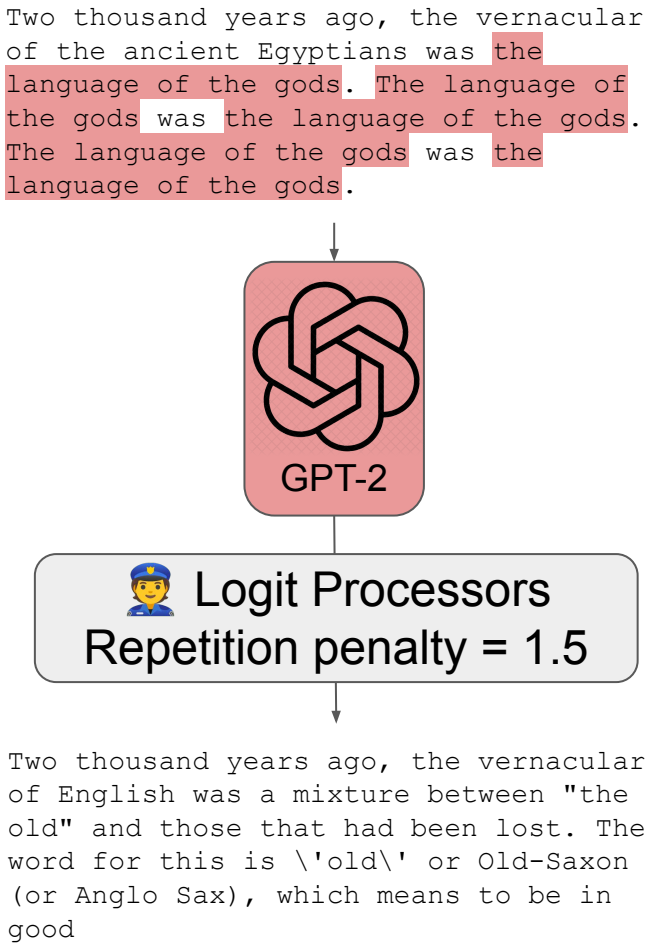
Don’t like the output?

Unwanted behaviour:

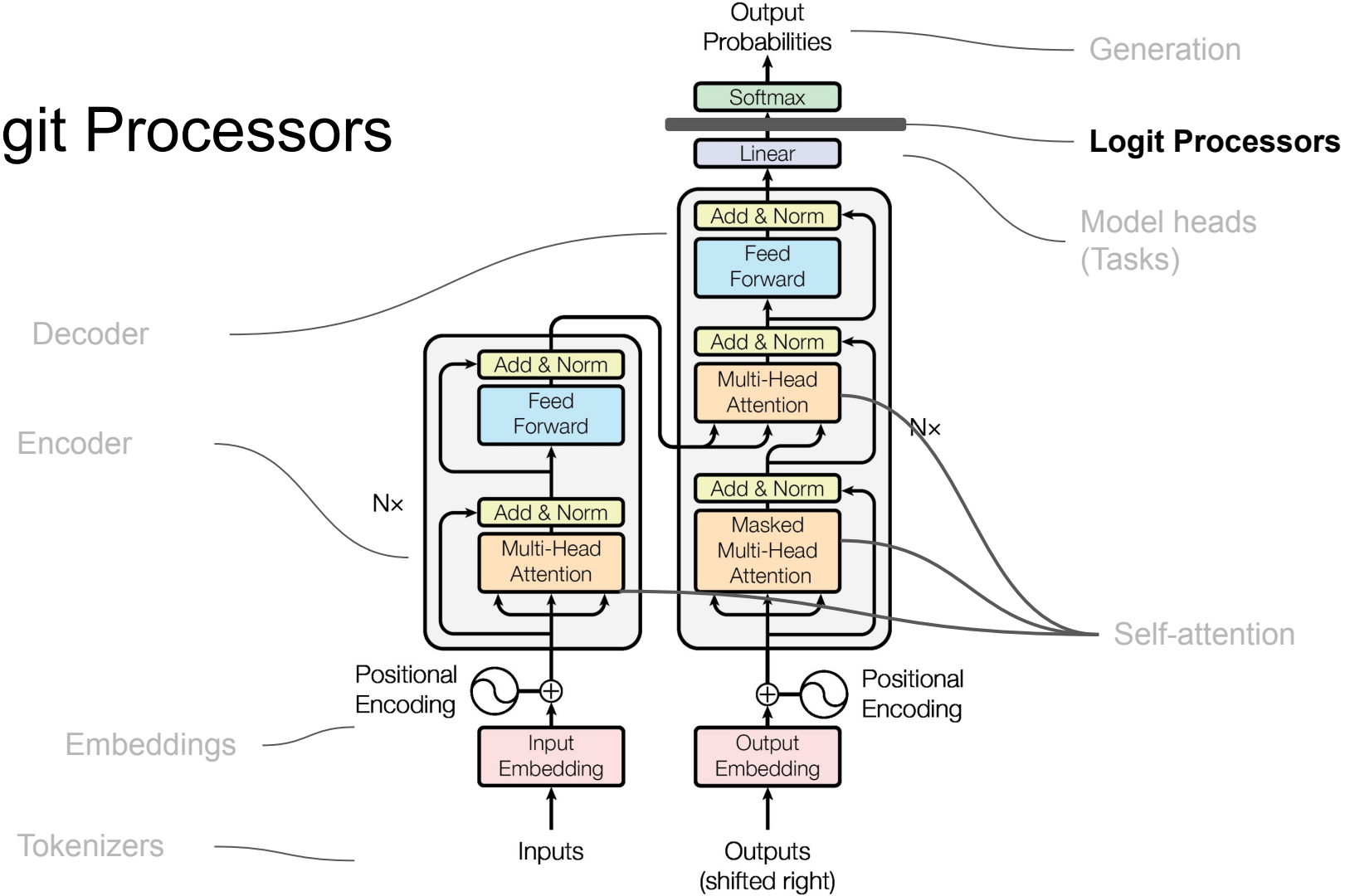
- Repetition
- Hallucinations / Divergence from input text

How to fix it:

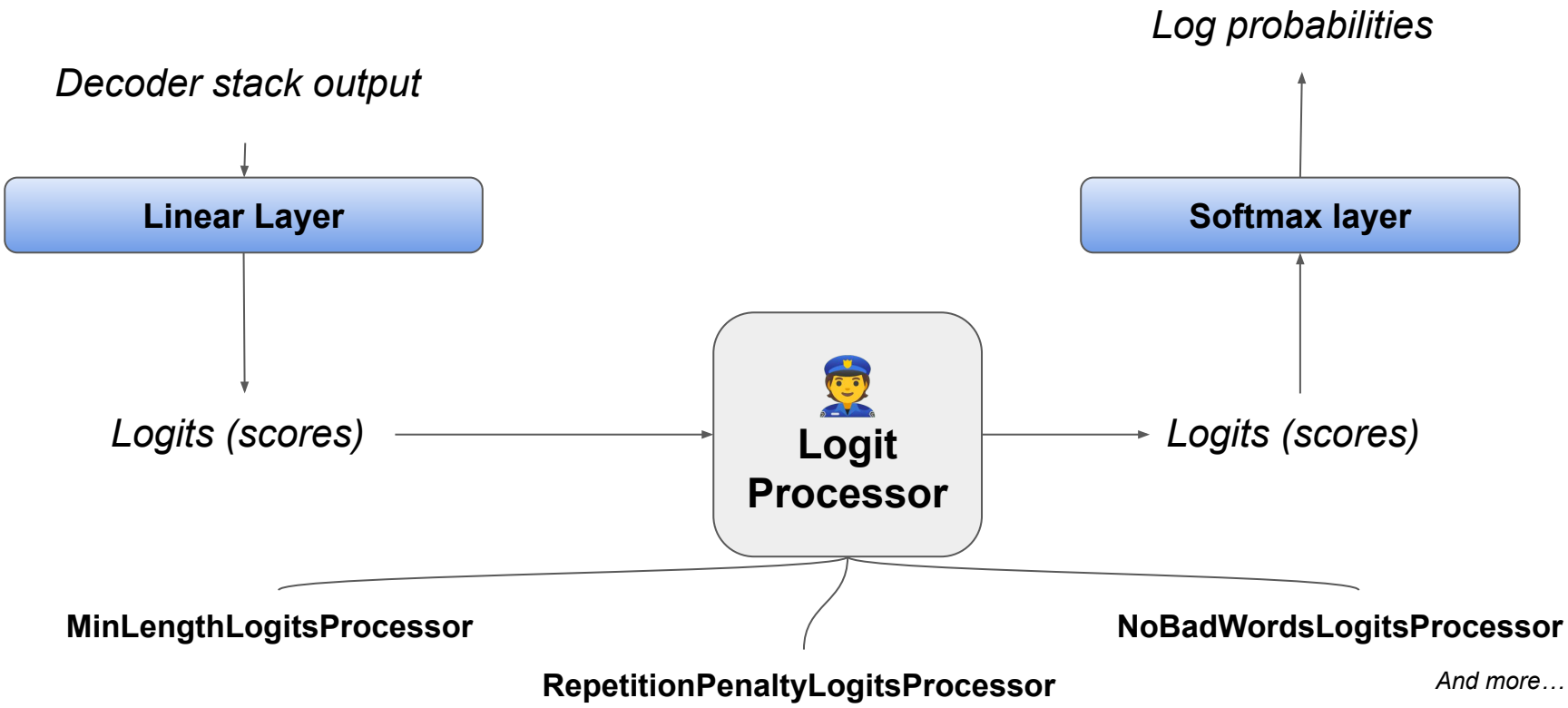
- Retrain a model and hope the behaviour disappears
- Adjust the output logits and constrain the generation



Logit Processors



Logit Processors



Logit Processors: JSONformer library

Only the green content is generated

Generate a random person.

```
{
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "age": {
      "type": "number"
    },
    "city": {
      "type": "string"
    }
  }
}
```

Generate a random person who is 50 years old.

JSONFormer Logit Processors

GPT-2

```
{
  name:
```

```
{
  name: "John Doe"
```

```
{
  name: "John Doe",
  age:
```

```
{
  name: "John Doe",
  age: 50.0
```

```
Generating...
{
  name: "John Doe",
  age: 0.5,
  city: "New York"
}
```

```
Generating...
{
  name: "John Doe",
  age: 50.0,
  city: "New York"
}
```

Logit Processors: Hands-on session

With the hype around text generation it is good to know ways to adjust the output, without needing to gather new data. Sometimes it is better to think of a constraint, than re-training

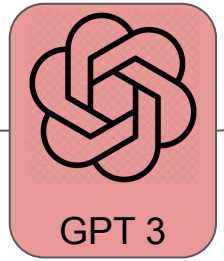
- Take the naive Logit Processor and adjust it for better performance:
  - Enforce not a single token word, but multiple words / sequence of tokens
  - Try fixing the repetition without using repetition penalty
- Experiment with different decoding strategies

Let’s take a short break

What are **Large** Language Models (LLMs)?

- Language model with a **vast amount of parameters** (billions)
- Unsupervised training on a **large amount of data**

"Review: You can get decent sushi for the same price!
Sentiment of this review is "



"Bad"

- LLM are able to **learn** from **input request, aka a “prompt”**
- These models exhibit: **in-context learning** capability:
  - The description of a task is a part of the prompt
  - In-context Learning is the ability to perform this task without the model being exposed to it prior
  - A well composed prompt can increase the likelihood of the correct output

Prompts

- The output of a LLM is greatly influenced by the prompt
- No such thing as a “best” prompt for a given task
  - Prompt performance depends on the model
- **Promptsource** library: Environment for prompt creation and repository of existing prompts
  - **English prompts only**
  - 180 prompted datasets
  - 2085 prompts


```
{
    "idx": 20,
    "label": 0,
    "passage": "Post-office box -- Street
Addressing will have the same street
address of the post office, plus a ``unit
number'' that matches the P.O. Box number.
…",
    "question": "does p o box come before
street address"
}
```


GPT-3 style

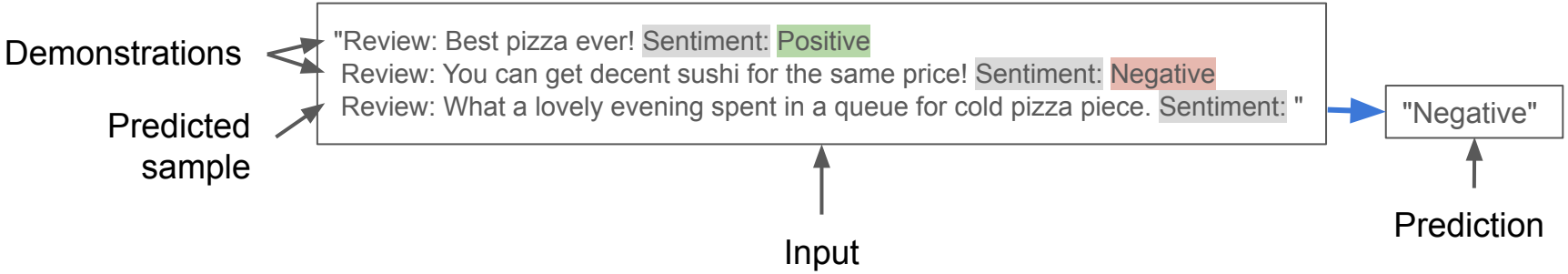
```
Prompt: "Post-office box -- Street
Addressing will have the same street
address of the post office, plus a
``unit number'' that matches the P.O.
Box number…..Question: does p o box come
before street address\nAnswer:"

Label: 'No'
```

After reading

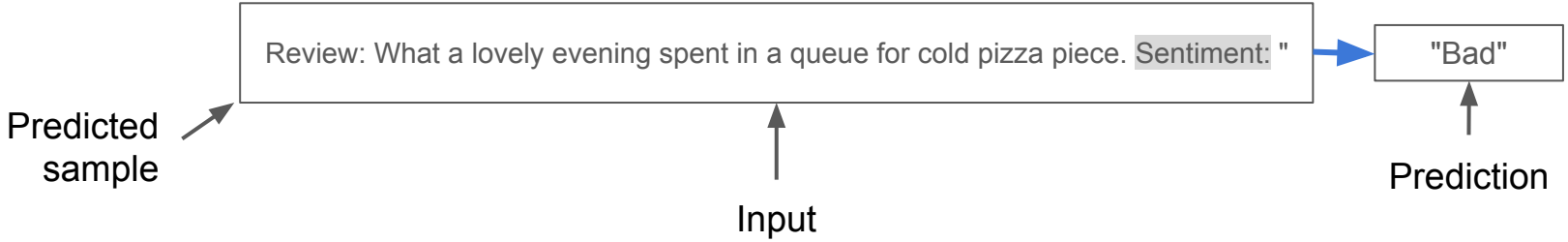
```
Prompt: "Passage: Post-office box --
Street Addressing will have the same
street address of the post office,
plus a ``unit number'' that matches
the P.O. Box number. ….\n\nAfter
reading this passage, I have a
question: does p o box come before
street address? True or False?"

Label: 'False'
```

Zero-shot vs Few-shot prompting

- Few-shot prompting : The instruction is illustrated on multiple demonstrations



- Zero-shot prompting: No demonstrations are given



Two questions

1. Why don’t we only use LLM’s for everything?
2. But why do we care about LLM, when we can train a specialized model for our task?

Let’s compare!

Specialized vs. Large language models

Specialized Models

- Training & inference is faster
- Smaller resources needed
- Smaller models

- **Single prediction** (classification/NER) takes ~**1.2** sec on 4-core CPU
- **Generation** takes ~**0.5** sec per token**

- Annotated data needed
- Task specific
- Any adjustment means re-training

- Classification requires **>100–200** examples/class*
- Named entity recognition requires **>50–100** examples/entity*
- **Generation** (e.g. custom Machine translation) requires **>5000** text pairs**

* Within the "easy" industrial applications & training dataset collected in quality-controlled setting, BERT-base-like models (110M params).
** One-sentence generation length, Transformer-base-like model (80M params)

Specialized vs. Large Language Models

Few-shot "training" data:

- **2-3 samples/class (100-times less)**
- **Prompt engineering (10 prompts ~ 15min)**

Generation with BLOOM (**176B** params):

1. **Deployment price**:
   - **On GPU**: $90,544/month*
   - **On CPU**: $3,602/month**
2. **Prediction time:**
   - **On GPU**: ~5sec/token
   - **On CPU**: several minutes/token

Large Language Models

- **No Annotated data**
- **Not task specific**

- Large resources needed
- ~1TB to persist the model

[1]: Hardware Requirements for CPU / GPU Inference
* 8 x Nvidia A6000 GPUs 40GB [1]
** 768 GB of RAM [1]; In 16-float weight precision, limited context windows (<100 input tokens)

Specialized vs. Large Language Models

Specialized Models

- **Training & inference is faster**
- **Smaller resources needed**
- **Smaller models**

- Annotated data needed
- Task specific
- Any adjustment means re-training

Large Language Models

- **No Annotated data**
- **Not task specific**
- **In-context learning capability**

- Training is slower
- Large resources needed
- Models are bigger

Specialized vs. Large Language Models

Specialized Models

- **Training & inference is faster**
- **Smaller resources needed**
- **Smaller models**

- Annotate
- Task s
- Any ac

Large Language Models

- **No Annotated data**
- **Not task specific**
- **In-context learning capability**

ing is slower
e resources needed
dels are bigger

How can we get benefits from both groups?

Smaller model with in-context learning property

How?

How about using few-shot prompting?

<u>Few-shot in-context learners</u>

Creating a dataset: Hands-on session

Before diving into the theory and training of few-shot in-context learners it is time to experiment with prompts, to get a feel for the performance difference between multiple prompts and zero-shot vs few-shot prompts.

- Download an existing dataset and transform it into a prompt input-label pair
  - Use a dataset which has existing prompts in Promptsource library
  - Or, create your own custom prompt
- Adjust the evaluation script to fit your dataset
- Create a function which will generate a prompt for few-shot learning.
- Evaluate the FLAN base model and see how it performs on your dataset

Coffee break: 10:30 - 11:00

Few-shot learning: Recap & Terminology

Demonstrations

Predicted sample

"Review: Best pizza ever! Sentiment: Positive
Review: You can get decent sushi for the same price! Sentiment: Negative
Review: What a lovely evening spent in a queue for cold pizza piece. Sentiment: "

Input

"Positive"

Prediction

Training = updating

Learning = in-context

Objectives: How to make ICL useful?

We need smaller ICL models!

But is it even possible?

Previous work: Smaller Few-shot ICL

Is this even possible?

Consider how much information GPT-3 remembers

Though this might be useful, if we want models to be able to **learn new tasks**, it is not necessary

Previous work: Smaller Few-shot ICL

Is this even possible? Yes!

After GPT3, Schick & Schutze [1] show that BERT (~110M) can also be fine-tuned to predict [MASK] on ≤ 64 samples, to the GPT3 performance:

Input: "The whole plot was ridiculous. Characters very just random.
In overall, the movie was [MASK]."

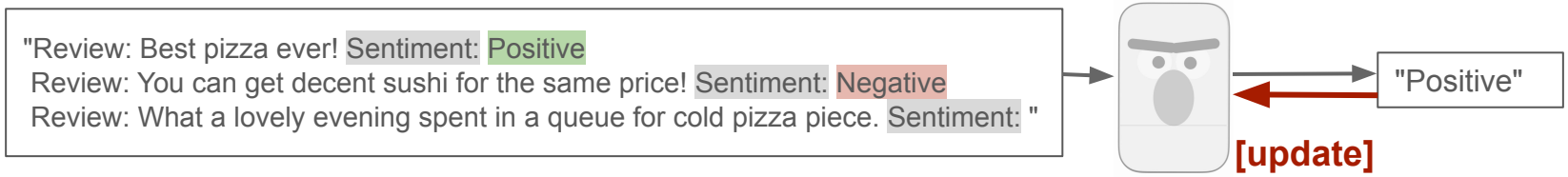
weird	0.347
bad	**0.268**
strange	0.108
…	
okay	**0.028**
fine	0.001

Prediction: **Negative**

[1]: Timo Schick, Hinrich Schütze. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In Proceedings of EACL 2021.

Previous work: Smaller Few-shot ICL

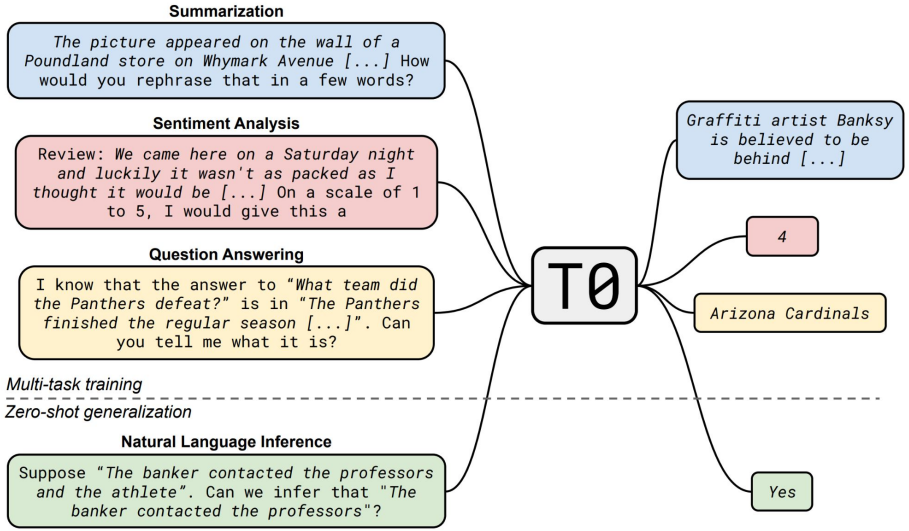
After that, **MetaICL** models of Min et al. [1] were the first to fine-tune **generative** GPT2 (~770M) on a 142-tasks mixture in a **few-shot format** and showed that such model can **generalize** to **unseen tasks.**



[1]: Sewon Min et al.; MetaICL: Learning to Learn In Context. In Proceedings of NAACL 2022.

Previous work: Smaller Few-shot ICL

T0 [T-zero] models of Sanh et al. [1] (3B & 11B) have shown that this approach also works in a **zero-shot** settings with **instructions** (*instruction tuning*)



[1]: Victor Sanh et al.; Multitask Prompted Training Enables Zero-Shot Task Generalization. In Proceedings of ICLR 2022.

Previous work: Smaller Few-shot ICL

Tk-Instruct models of Wang et al. [1] (400M – 11B) combine *few-shot fine-tuning* and *instruction-tuning* and simulate the tasks **as presented to the annotators**.

- They create a Natural-Instruction collection of **1600+ tasks** casted to this format, incl. **576 non-English tasks** in 55 languages (but with English instructions)
- **mTk-Instruct** model also includes non-English tasks in the training

Task Instruction

Definition

"... Given an utterance and recent dialogue context containing past 3 utterances (wherever available), output 'Yes' if the utterance contains the small-talk strategy, otherwise output 'No'. Small-talk is a cooperative negotiation strategy. It is used for discussing topics apart from the negotiation, to build a rapport with the opponent."

Positive Examples

- **Input:** "Context: … *'That's fantastic, I'm glad we came to something we both agree with.'* Utterance: *'Me too. I hope you have a wonderful camping trip.'*"
- **Output:** "Yes"
- **Explanation:** "The participant engages in small talk when wishing their opponent to have a wonderful trip."

Negative Examples

- **Input:** "Context: … *'Sounds good, I need food the most, what is your most needed item?!'* Utterance: *'My item is food too'*."
- **Output:** "Yes"
- **Explanation:** "The utterance only takes the negotiation forward and there is no side talk. Hence, the correct answer is 'No'."

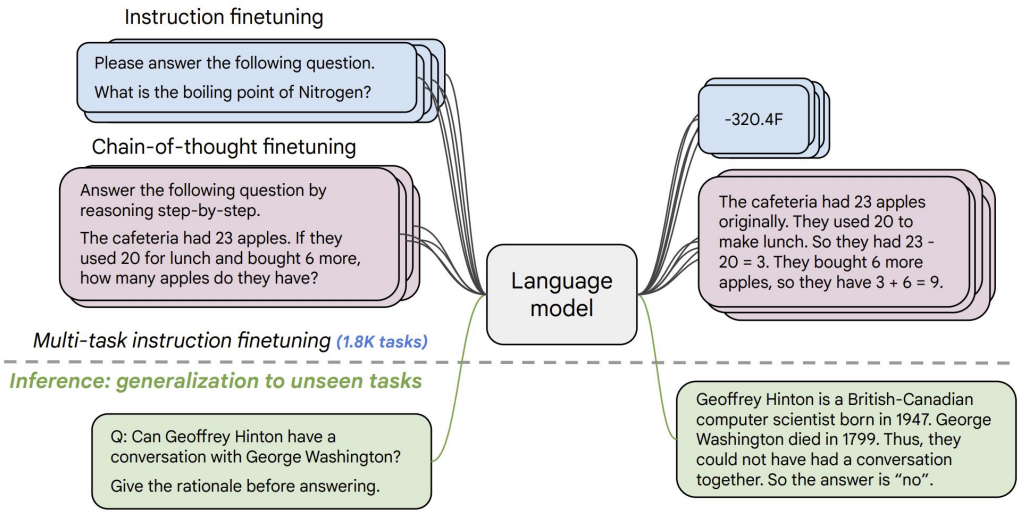
[1]: Yizhong Wang et al.; Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. In Proceedings of EMNLP 2022.

Previous work: Smaller Few-shot ICL

FLAN models of Chung et al. [1] (400M – 11B) further **extend** Natural-Instructions training collection with **Chain-of-Thought** (CoT)

- LM is trained to generate a **sequence of steps** deducing the correct answer
- This brings quite large improvements in ICL on new tasks

Why?



[1]: Hyung Won Chung et al.; Scaling Instruction-Finetuned Language Models. Arxiv preprint:2210.11416. Dec. 2022.

Previous work: Smaller Few-shot ICL

So we do not need model scale, but **data scale**?

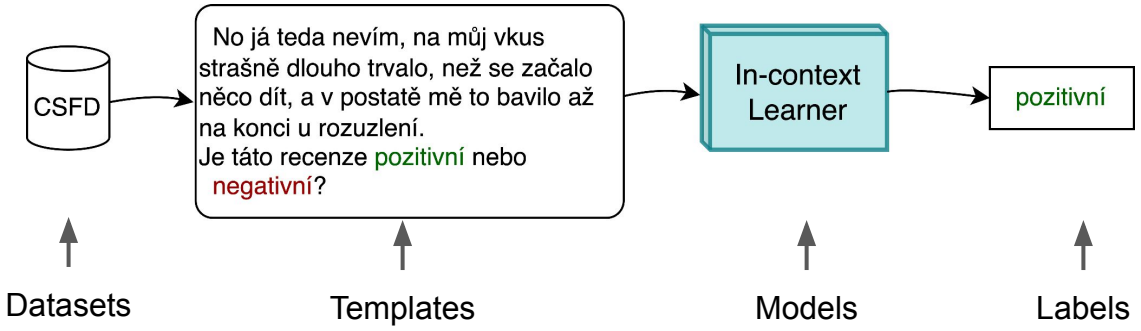
If so, we can not train ICLs e.g. for non-English languages, with little data!

Our work - Can ICL work in **Multilingual** settings?

- Previous work fine-tuned Few-shot ICLs on tasks outside English, but the **prompts were always in English**
- It never evaluated new ICLs in target-language tasks + interaction
- Can ICL outside English work in practically-usable quality?

Our work - Can ICL work in **Multilingual** settings?

- In our work [1], we **collected** 50+ target-language templates for 14 tasks in {Czech, Polish, Russian} by native speakers of each language
- We evaluated existing ICLs for interaction **solely** in target language
- We **trained** new ICLs **specialized** to each target language



[1]: Michal Štefánik et al.; Resources and Few-shot Learners for In-context Learning in Slavic Languages. In Proc. of EACL SlavicNLP 2023.

Our work - Multilingual templates

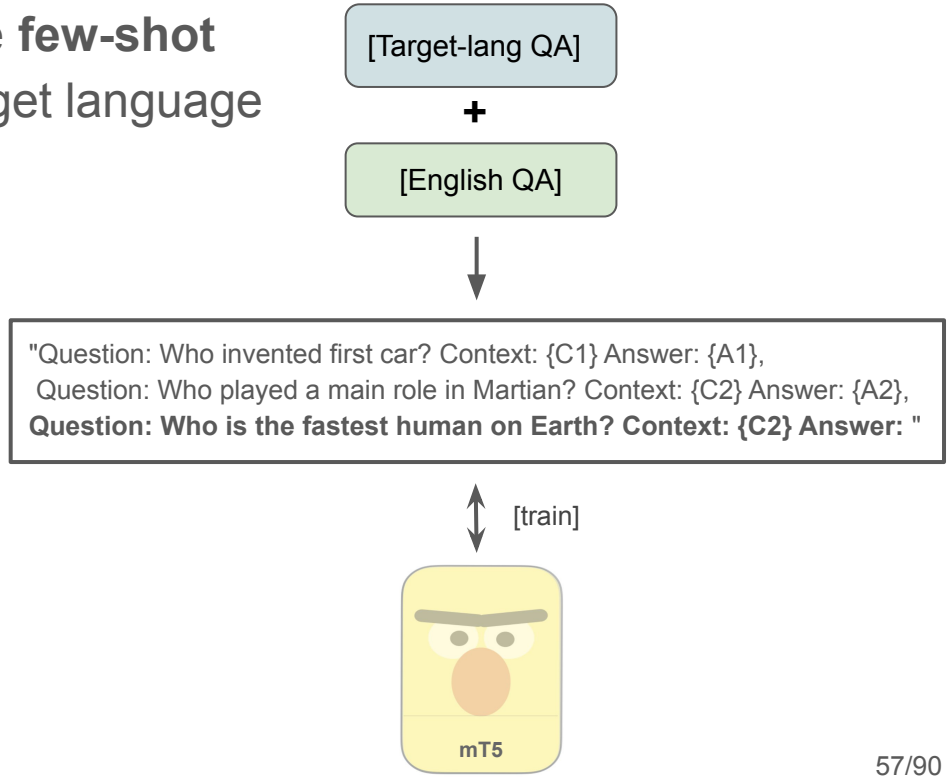
Lang	Task	Template
cs	NER	{{text}} Jaká entita typu {{label_type}} se nachází v předchozím odstavci?
cs	Clf.	{{comment}} Je tato recenze {{"pozitivní, neutrální nebo negativní"}}?
cs	QA	{{context}} Q: {{question}} S odkazem na sekci výše je správná odpověď na danou otázku
cs	NLI	Za předpokladu, že {{evidence}} vyplývá, že {{claim}}? Ano, ne, nebo možná?
pl	clf.	"{{text}}" Ten tekst jest pozytywny, negatywny, neutralny czy dwuznaczny?
pl	NLI	Oceń czy poniższe zdania są zgodne ze sobą - tak, nie czy nie wiadomo? Zdanie A: {{premise}} Zdanie B: {{hypothesis}} Zgodność:
pl	NER	Jaka encja typu {{label_type_selected}} znajduje się w następującym tekście? "{{text}}"
ru	NER	{{text}} Какой объект типа {{label_type}} находится в предыдущем абзаце?
ru	NLI	Примите за истину следующее: {{premise}} Тогда следующее утверждение: "{{hypothesis}}" есть "правда", "ложь" или "неубедительно"?
ru	QA	Посмотрите на абзац ниже и ответьте на следующий вопрос: Абзац: {{context}} Вопрос: {{question}}
ru	Clf.	{{text}} Каково настроение этого обзора? радость, печаль, удивление, страх или гнев?

Table 2: Examples of instruction templates for each of the language + task pair that we collect in this work. A full list of templates collected in this work by our native speakers can be found in Appendix A Table 6.

[1]: Michal Štefánik et al.; Resources and Few-shot Learners for In-context Learning in Slavic Languages. In Proc. of EACL SlavicNLP 2023.

Our work - Training Multilingual ICL Models

- We fine-tune the mT5 models in the **few-shot format**, on samples of English+Target language **QA datasets**
- We evaluate performance on unseen tasks using our templates



Our work - Multilingual evaluation

Dataset + task / Model	Czech: CNEC NER	Czech: CSFD Clf.	Czech: FBCom Clf.	Czech: MALL Clf.	Czech: SQAD QA	Czech: CTKFacts NLI	Russian: Polyglot NER	Russian: CEDR Clf.	Russian: SberQAD QA	Russian: XNLI NLI	Polish: PoliticAds NER	Polish: KPWR NER	Polish: Polemo Clf.	Polish: CDSC NLI
Supervised (mT5-1B)	67.9± 9.1	82.4±4.5	49.3±10.3	42.8±10.8	88.3±5.3	56.1±10.9	54.3±10.8	48.6±9.6	86.4±6.5	51.5±11.5	5.9±5.1	63.3±10.3	51.9±9.9	75.5±8.5
TK-Instruct (700M)	15.3± 6.7	14.1±7.1	25.2± 7.2	25.5± 8.4	5.6±4.8	**54.7**±8.2	0.1±0.5	12.2±6.8	0.6±1.1	12.9± 6.9	5.6±4.3	8.6±5.4	28.3±8.6	52.3±8.2
TK-Instruct (3B)	32.8± 9.1	20.9±8.1	23.0± 7.4	25.1± 6.9	34.0±9.0	47.8±9.8	3.6±3.9	17.7±8.3	8.1±4.1	22.2± 8.2	17.6±8.1	54.6±11.2	19.5±8.4	67.8±8.8
T5-FLAN (700M)	41.1±10.0	0.0± 0.0	0.0± 0.0	0.0± 0.0	46.5±8.4	30.3±9.3	1.0±1.6	15.1±6.1	11.4±4.8	13.8± 6.2	6.8±5.5	33.8±9.8	24.3±8.6	10.0±6.4
T5-FLAN (3B)	49.6±10.4	0.0± 0.0	0.0± 0.0	0.1± 0.1	51.6±9.1	34.7±10.7	2.0±2.5	24.4±7.4	19.6±5.6	26.0± 9.0	18.4±7.3	60.5±7.8	**43.0**±9.0	**71.5**±9.0
mTK-Instruct (3B)	62.5± 8.9	**90.2**±4.2	10.8±6.2	9.9±7.0	67.9±8.6	44.0±10.1	57.6±11.2	**33.0**±9.9	73.7±6.7	**35.3**±10.3	**32.1**±9.6	**67.6**±8.4	25.4±8.6	70.6±8.2
mTK-QA$_{none}$(1B)	72.0± 9.0	45.9±9.1	29.2±8.2	32.1±8.9	85.0±7.0*	35.4±10.5	53.3±8.4	17.9±8.1	**89.1**±5.2*	19.6± 7.5				
mTK-QA$_{SQUAD}$(1B)	72.3± 9.1	72.9±6.6	**32.1**±9.0	**34.7**±9.2	**87.8**±5.3*	46.9±10.1	50.3±9.3	7.5±4.5	84.6±6.0*	23.8± 8.8				
mTK-QA$_{AQA}$(1B)	**77.0**± 7.8	59.8±8.8	27.6±8.6	29.8±9.9	87.1±6.6*	42.7±10.7	**66.3**±10.9	27.0±9.9	86.0±5.6*	32.3± 8.3				

[1]: Michal Štefánik et al.; Resources and Few-shot Learners for In-context Learning in Slavic Languages. In Proc. of EACL SlavicNLP 2023.

Our work - Multilingual evaluation

Model \ Dataset + task	Czech: CNEC NER	Czech: CSFD Clf.	Czech: FBCom Clf.	Czech: MALL Clf.	Czech: SQAD QA	Czech: CTKFacts NLI	Russian: Polyglot NER	Russian: CEDR Clf.	Russian: SberQAD QA	Russian: XNLI NLI	Polish: PoliticAds NER	Polish: KPWR NER	Polish: Polemo Clf.	Polish: CDSC NLI
Supervised (mT5-1B)	67.9± 9.1	82.4±4.5	49.3±10.3	42.8±10.8	88.3±5.3	56.1±10.9	54.3±10.8	48.6±9.6	86.4±6.5	51.5±11.5	5.9±5.1	63.3±10.3	51.9±9.9	75.5±8.5
TK-Instruct (700M)	15.3± 6.7	14.1±7.1	25.2± 7.2	25.5± 8.4	5.6±4.8	**54.7**±8.2	0.1±0.5	12.2±6.8	0.6±1.1	12.9± 6.9	5.6±4.3	8.6±5.4	28.3±8.6	52.3±8.2
TK-Instruct (3B)	32.8± 9.1	20.9±8.1	23.0± 7.4	25.1± 6.9	34.0±9.0	47.8±9.8	3.6±3.9	17.7±8.3	8.1±4.1	22.2± 8.2	17.6±8.1	54.6±11.2	19.5±8.4	67.8±8.8
T5-FLAN (700M)	41.1±10.0	0.0± 0.0	0.0± 0.0	0.0± 0.0	46.5±8.4	30.3±9.3	1.0±1.6	15.1±6.1	11.4±4.8	13.8± 6.2	6.8±5.5	33.8±9.8	24.3±8.6	10.0±6.4
T5-FLAN (3B)	49.6±10.4	0.0± 0.0	0.0± 0.0	0.1± 0.1	51.6±9.1	34.7±10.7	2.0±2.5	24.4±7.4	19.6±5.6	26.0± 9.0	18.4±7.3	60.5±7.8	**43.0**±9.0	**71.5**±9.0
mTK-Instruct (3B)	62.5± 8.9	**90.2**±4.2	10.8±6.2	9.9±7.0	67.9±8.6	44.0±10.1	57.6±11.2	**33.0**±9.9	73.7±6.7	**35.3**±10.3	**32.1**±9.6	**67.6**±8.4	25.4±8.6	70.6±8.2
mTK-QA$_{none}$(1B)	72.0± 9.0	45.9±9.1	29.2±8.2	32.1±8.9	85.0±7.0*	35.4±10.5	53.3±8.4	17.9±8.1	**89.1**±5.2*	19.6± 7.5				
mTK-QA$_{SQUAD}$(1B)	72.3± 9.1	72.9±6.6	**32.1**±9.0	**34.7**±9.2	**87.8**±5.3*	46.9±10.1	50.3±9.3	7.5±4.5	84.6±6.0*	23.8± 8.8				
mTK-QA$_{AQA}$(1B)	**77.0**± 7.8	59.8±8.8	27.6±8.6	29.8±9.9	87.1±6.6*	42.7±10.7	**66.3**±10.9	27.0±9.9	86.0±5.6*	32.3± 8.3				

Multilingual ICLs

[1]: Michal Štefánik et al.; Resources and Few-shot Learners for In-context Learning in Slavic Languages. In Proc. of EACL SlavicNLP 2023.

Our work - Multilingual evaluation

Dataset + task / Model	Czech: CNEC NER	CSFD Clf.	FBCom Clf.	MALL Clf.	SQAD QA	CTKFacts NLI	Russian: Polyglot NER	CEDR Clf.	SberQAD QA	XNLI NLI	Polish: PoliticAds NER	KPWR NER	Polemo Clf.	CDSC NLI
Supervised (mT5-1B)	67.9± 9.1	82.4±4.5	49.3±10.3	42.8±10.8	88.3±5.3	56.1±10.9	54.3±10.8	48.6±9.6	86.4±6.5	51.5±11.5	5.9±5.1	63.3±10.3	51.9±9.9	75.5±8.5
TK-Instruct (700M)	15.3± 6.7	14.1±7.1	25.2± 7.2	25.5± 8.4	5.6±4.8	**54.7**±8.2	0.1±0.5	12.2±6.8	0.6±1.1	12.9± 6.9	5.6±4.3	8.6±5.4	28.3±8.6	52.3±8.2
TK-Instruct (3B)	32.8± 9.1	20.9±8.1	23.0± 7.4	25.1± 6.9	34.0±9.0	47.8±9.8	3.6±3.9	17.7±8.3	8.1±4.1	22.2± 8.2	17.6±8.1	54.6±11.2	19.5±8.4	67.8±8.8
T5-FLAN (700M)	41.1±10.0	0.0± 0.0	0.0± 0.0	0.0± 0.0	46.5±8.4	30.3±9.3	1.0±1.6	15.1±6.1	11.4±4.8	13.8± 6.2	6.8±5.5	33.8±9.8	24.3±8.6	10.0±6.4
T5-FLAN (3B)	49.6±10.4	0.0± 0.0	0.0± 0.0	0.1± 0.1	51.6±9.1	34.7±10.7	2.0±2.5	24.4±7.4	19.6±5.6	26.0± 9.0	18.4±7.3	60.5±7.8	**43.0**±9.0	**71.5**±9.0
mTK-Instruct (3B)	62.5± 8.9	**90.2**±4.2	10.8±6.2	9.9±7.0	67.9±8.6	44.0±10.1	57.6±11.2	**33.0**±9.9	73.7±6.7	**35.3**±10.3	**32.1**±9.6	**67.6**±8.4	25.4±8.6	70.6±8.2
mTK-QA$_{none}$(1B)	72.0± 9.0	45.9±9.1	29.2±8.2	32.1±8.9	85.0±7.0*	35.4±10.5	53.3±8.4	17.9±8.1	**89.1**±5.2*	19.6± 7.5				
mTK-QA$_{SQUAD}$(1B)	72.3± 9.1	72.9±6.6	**32.1**±9.0	**34.7**±9.2	**87.8**±5.3*	46.9±10.1	50.3±9.3	7.5±4.5	84.6±6.0*	23.8± 8.8				
mTK-QA$_{AQA}$(1B)	**77.0**± 7.8	59.8±8.8	27.6±8.6	29.8±9.9	87.1±6.6*	42.7±10.7	**66.3**±10.9	27.0±9.9	86.0±5.6*	32.3± 8.3				

Multilingual ICLs

[1]: Michal Štefánik et al.; Resources and Few-shot Learners for In-context Learning in Slavic Languages. In Proc. of EACL SlavicNLP 2023.

Our work - Multilingual evaluation

Dataset + task / Model	Czech: CNEC NER	Czech: CSFD Clf.	Czech: FBCom Clf.	Czech: MALL Clf.	Czech: SQAD QA	Czech: CTKFacts NLI	Russian: Polyglot NER	Russian: CEDR Clf.	Russian: SberQAD QA	Russian: XNLI NLI	Polish: PoliticAds NER	Polish: KPWR NER	Polish: Polemo Clf.	Polish: CDSC NLI
Supervised (mT5-1B)	67.9± 9.1	82.4±4.5	49.3±10.3	42.8±10.8	88.3±5.3	56.1±10.9	54.3±10.8	48.6±9.6	86.4±6.5	51.5±11.5	5.9±5.1	63.3±10.3	51.9±9.9	75.5±8.5
TK-Instruct (700M)	15.3± 6.7	14.1±7.1	25.2± 7.2	25.5± 8.4	5.6±4.8	**54.7**±8.2	0.1±0.5	12.2±6.8	0.6±1.1	12.9± 6.9	5.6±4.3	8.6±5.4	28.3±8.6	52.3±8.2
TK-Instruct (3B)	32.8± 9.1	20.9±8.1	23.0± 7.4	25.1± 6.9	34.0±9.0	47.8±9.8	3.6±3.9	17.7±8.3	8.1±4.1	22.2± 8.2	17.6±8.1	54.6±11.2	19.5±8.4	67.8±8.8
T5-FLAN (700M)	41.1±10.0	0.0± 0.0	0.0± 0.0	0.0± 0.0	46.5±8.4	30.3±9.3	1.0±1.6	15.1±6.1	11.4±4.8	13.8± 6.2	6.8±5.5	33.8±9.8	24.3±8.6	10.0±6.4
T5-FLAN (3B)	49.6±10.4	0.0± 0.0	0.0± 0.0	0.1± 0.1	51.6±9.1	34.7±10.7	2.0±2.5	24.4±7.4	19.6±5.6	26.0± 9.0	18.4±7.3	60.5±7.8	**43.0**±9.0	**71.5**±9.0
mTK-Instruct (3B)	62.5± 8.9	**90.2**±4.2	10.8±6.2	9.9±7.0	67.9±8.6	44.0±10.1	57.6±11.2	**33.0**±9.9	73.7±6.7	**35.3**±10.3	**32.1**±9.6	**67.6**±8.4	25.4±8.6	70.6±8.2
mTK-QA$_{none}$(1B)	72.0± 9.0	45.9±9.1	29.2±8.2	32.1±8.9	85.0±7.0*	35.4±10.5	53.3±8.4	17.9±8.1	**89.1**±5.2*	19.6± 7.5				
mTK-QA$_{SQUAD}$(1B)	72.3± 9.1	72.9±6.6	**32.1**±9.0	**34.7**±9.2	**87.8**±5.3*	46.9±10.1	50.3±9.3	7.5±4.5	84.6±6.0*	23.8± 8.8				
mTK-QA$_{AQA}$(1B)	**77.0**± 7.8	59.8±8.8	27.6±8.6	29.8±9.9	87.1±6.6*	42.7±10.7	**66.3**±10.9	27.0±9.9	86.0±5.6*	32.3± 8.3				

Instruction tuning

+ Chain-of-Thought

[1]: Michal Štefánik et al.; Resources and Few-shot Learners for In-context Learning in Slavic Languages. In Proc. of EACL SlavicNLP 2023.

Our work - Multilingual evaluation

Model \ Dataset + task	Czech: CNEC NER	Czech: CSFD Clf.	Czech: FBCom Clf.	Czech: MALL Clf.	Czech: SQAD QA	Czech: CTKFacts NLI	Russian: Polyglot NER	Russian: CEDR Clf.	Russian: SberQAD QA	Russian: XNLI NLI	Polish: PoliticAds NER	Polish: KPWR NER	Polish: Polemo Clf.	Polish: CDSC NLI
Supervised (mT5-1B)	67.9± 9.1	82.4±4.5	49.3±10.3	42.8±10.8	88.3±5.3	56.1±10.9	54.3±10.8	48.6±9.6	86.4±6.5	51.5±11.5	5.9±5.1	63.3±10.3	51.9±9.9	75.5±8.5
Tk-Instruct (700M)	15.3± 6.7	14.1±7.1	25.2± 7.2	25.5± 8.4	5.6±4.8	**54.7**±8.2	0.1±0.5	12.2±6.8	0.6±1.1	12.9± 6.9	5.6±4.3	8.6±5.4	28.3±8.6	52.3±8.2
Tk-Instruct (3B)	32.8± 9.1	20.9±8.1	23.0± 7.4	25.1± 6.9	34.0±9.0	47.8±9.8	3.6±3.9	17.7±8.3	8.1±4.1	22.2± 8.2	17.6±8.1	54.6±11.2	19.5±8.4	67.8±8.8
T5-FLAN (700M)	41.1±10.0	0.0± 0.0	0.0± 0.0	0.0± 0.0	46.5±8.4	30.3±9.3	1.0±1.6	15.1±6.1	11.4±4.8	13.8± 6.2	6.8±5.5	33.8±9.8	24.3±8.6	10.0±6.4
T5-FLAN (3B)	49.6±10.4	0.0± 0.0	0.0± 0.0	0.1± 0.1	51.6±9.1	34.7±10.7	2.0±2.5	24.4±7.4	19.6±5.6	26.0± 9.0	18.4±7.3	60.5±7.8	**43.0**±9.0	**71.5**±9.0
mTk-Instruct (3B)	62.5± 8.9	**90.2**±4.2	10.8±6.2	9.9±7.0	67.9±8.6	44.0±10.1	57.6±11.2	**33.0**±9.9	73.7±6.7	**35.3**±10.3	**32.1**±9.6	**67.6**±8.4	25.4±8.6	70.6±8.2
mTk-QA$_{none}$(1B)	72.0± 9.0	45.9±9.1	29.2±8.2	32.1±8.9	85.0±7.0*	35.4±10.5	53.3±8.4	17.9±8.1	**89.1**±5.2*	19.6± 7.5				
mTk-QA$_{SQuAD}$(1B)	72.3± 9.1	72.9±6.6	**32.1**±9.0	**34.7**±9.2	**87.8**±5.3*	46.9±10.1	50.3±9.3	7.5±4.5	84.6±6.0*	23.8± 8.8				
mTk-QA$_{AQA}$(1B)	**77.0**± 7.8	59.8±8.8	27.6±8.6	29.8±9.9	87.1±6.6*	42.7±10.7	**66.3**±10.9	27.0±9.9	86.0±5.6*	32.3± 8.3				

only target-lang data

+ transfer-lang data

[1]: Michal Štefánik et al.; Resources and Few-shot Learners for In-context Learning in Slavic Languages. In Proc. of EACL SlavicNLP 2023.

Our work - Multilingual evaluation

Model \ Dataset + task	Czech: CNEC NER	Czech: CSFD Clf.	Czech: FBCom Clf.	Czech: MALL Clf.	Czech: SQAD QA	Czech: CTKFacts NLI	Russian: Polyglot NER	Russian: CEDR Clf.	Russian: SberQAD QA	Russian: XNLI NLI	Polish: PoliticAds NER	Polish: KPWR NER	Polish: Polemo Clf.	Polish: CDSC NLI
Supervised (mT5-1B)	67.9± 9.1	82.4±4.5	49.3±10.3	42.8±10.8	88.3±5.3	56.1±10.9	54.3±10.8	48.6±9.6	86.4±6.5	51.5±11.5	5.9±5.1	63.3±10.3	51.9±9.9	75.5±8.5
Tk-Instruct (700M)	15.3± 6.7	14.1±7.1	25.2± 7.2	25.5± 8.4	5.6±4.8	**54.7**±8.2	0.1±0.5	12.2±6.8	0.6±1.1	12.9± 6.9	5.6±4.3	8.6±5.4	28.3±8.6	52.3±8.2
Tk-Instruct (3B)	32.8± 9.1	20.9±8.1	23.0± 7.4	25.1± 6.9	34.0±9.0	47.8±9.8	3.6±3.9	17.7±8.3	8.1±4.1	22.2± 8.2	17.6±8.1	54.6±11.2	19.5±8.4	67.8±8.8
T5-FLAN (700M)	41.1±10.0	0.0± 0.0	0.0± 0.0	0.0± 0.0	46.5±8.4	30.3±9.3	1.0±1.6	15.1±6.1	11.4±4.8	13.8± 6.2	6.8±5.5	33.8±9.8	24.3±8.6	10.0±6.4
T5-FLAN (3B)	49.6±10.4	0.0± 0.0	0.0± 0.0	0.1± 0.1	51.6±9.1	34.7±10.7	2.0±2.5	24.4±7.4	19.6±5.6	26.0± 9.0	18.4±7.3	60.5±7.8	**43.0**±9.0	**71.5**±9.0
mTk-Instruct (3B)	62.5± 8.9	**90.2**±4.2	10.8±6.2	9.9±7.0	67.9±8.6	44.0±10.1	57.6±11.2	**33.0**±9.9	73.7±6.7	**35.3**±10.3	**32.1**±9.6	**67.6**±8.4	25.4±8.6	70.6±8.2
mTk-QA$_{none}$(1B)	72.0± 9.0	45.9±9.1	29.2±8.2	32.1±8.9	85.0±7.0*	35.4±10.5	53.3±8.4	17.9±8.1	**89.1**±5.2*	19.6± 7.5				
mTk-QA$_{SQUAD}$(1B)	72.3± 9.1	72.9±6.6	**32.1**±9.0	**34.7**±9.2	**87.8**±5.3*	46.9±10.1	50.3±9.3	7.5±4.5	84.6±6.0*	23.8± 8.8				
mTk-QA$_{AQA}$(1B)	**77.0**± 7.8	59.8±8.8	27.6±8.6	29.8±9.9	87.1±6.6*	42.7±10.7	**66.3**±10.9	27.0±9.9	86.0±5.6*	32.3± 8.3				

Training on 1600+ tasks

Single-task training (QA)

[1]: Michal Štefánik et al.; Resources and Few-shot Learners for In-context Learning in Slavic Languages. In Proc. of EACL SlavicNLP 2023.

Our work - Can ICL work in **Multilingual** settings?

- **Yes!**
- Quality largely **depends on a task**
- But for some tasks, a quality of Multilingual ICL can already **exceed** classic supervised fine-tuning on a full dataset!
- ICLs can **benefit** also **from English data** & training strategies

  => Future improvements in ICL are relevant also for other languages

[1]: Michal Štefánik et al.; Resources and Few-shot Learners for In-context Learning in Slavic Languages. In Proc. of EACL SlavicNLP 2023.

So what makes LMs to *Learn to Learn*?

- It's not parameters scale [Previous work on smaller models]
- It's not scale of data either:
  1. Our work [1] (previous slides) use only two QA datasets and outperforms mTk-Instruct (1600+ tasks) in 7/10 evaluations
  2. Zhou et al. [2] train a LIMA LM on only 1,000 carefully curated prompts and find responses of LIMA preferred over GPT4 in 43% of cases!

[1]: Michal Štefánik et al.; Resources and Few-shot Learners for In-context Learning in Slavic Languages. In Proc. of EACL SlavicNLP 2023.
[2]: Chunting Zhou et al.; LIMA: Less Is More for Alignment. ArXiv:2305.11206. May 2023.

Theory behind: Are ICL Models actually *learning*?

- Lu et al. [1] first report on the sensitivity of LMs to the **specific formulation** of the prompt
- Liu et al. [2] find LMs' surprising sensitivity to the **ordering of demonstrations**
- Min et al. [3] find that if you **shuffle demonstrations' labels**, ICL models perform almost exactly the same well!

"Review: Best pizza ever! Sentiment: ~~Positive~~ Negative
Review: You can get decent sushi for the same price! Sentiment: ~~Negative~~ Positive
Review: What a lovely evening spent in a queue for cold pizza piece. Sentiment: "

✅ "Positive"

- Wei et al. [4] explain this by LMs' **reliance** on labels' **semantics** (meaning)

[1]: Yao Lu et al.; Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity. In Proc. ACL 2022.
[2]: Jiachang Liu et al.; What Makes Good In-Context Examples for GPT-3? In Proc. ACL DeeLIO 2022.
[3]: Sewon Min et al.; Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In Proc. EMNLP 2022.
[4]: Jerry Wei et al.; Larger language models do in-context learning differently. ArXiv:2303.03846. March 2023.

Theory behind: Are ICL Models actually *learning*?

- Lu et al. [1] first report on the sensitivity of LMs to the **specific formulation** of the prompt
- Liu et al. [2] find LMs' surprising sensitivity to the **ordering of demonstrations**
- Min et al. [3] find that if you **shuffle demonstrations' labels**, ICL models perform almost exactly the same well!

= **LMs do not always perform in-context *learning* when we expect them to!**

[1]: Yao Lu et al.; Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity. In Proc. ACL 2022.
[2]: Jiachang Liu et al.; What Makes Good In-Context Examples for GPT-3? In Proc. ACL DeeLIO 2022.
[3]: Sewon Min et al.; Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In Proc. EMNLP 2022.
[4]: Jerry Wei et al.; Larger language models do in-context learning differently. ArXiv:2303.03846. March 2023.

Theory behind: Are ICL Models actually *learning*?

Not always, but in many cases **there is no other explanation**!

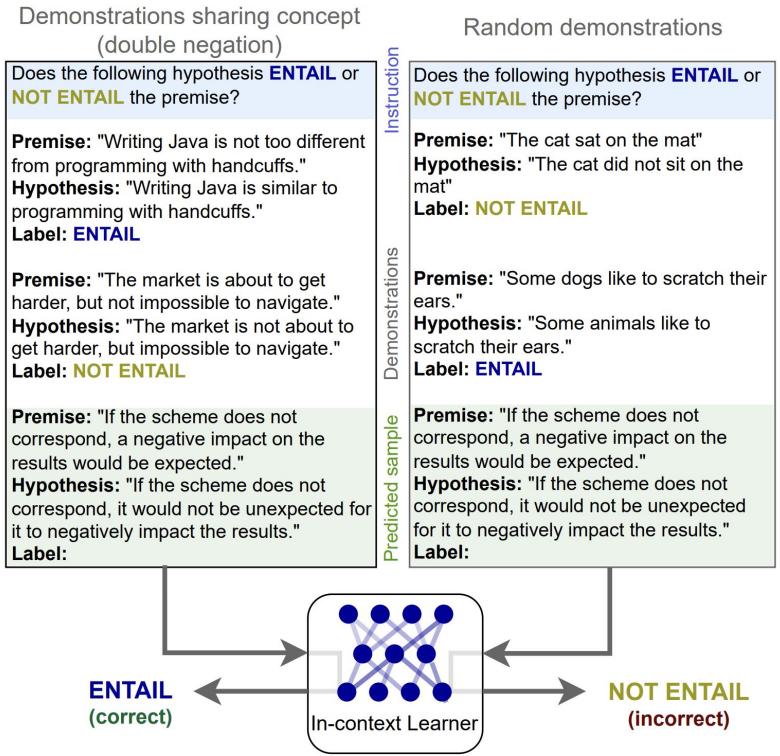
- E.g. Akyürek [1] or Li [2] show that Transformers can in-context *fit* ML algorithms, like Linear Regression parameters or can even choose an algorithm from collection!

[1]: Ekin Akyürek et al.; What learning algorithm is in-context learning? Investigations with linear models. In Proceedings of ICLR 2023.
[2]: Yingcong Li et al.; Transformers as Algorithms: Generalization and Stability in In-context Learning ArXiv:2301.07067 Feb. 2023.

Theory behind: What can ICL Models learn from context?

In [1], we extracted a set of **analogical reasoning concepts** from the explanations of selected NLP datasets

- We presented LMs with demonstrations exhibiting analogical reasoning and measured **if they can learn** the presented reasoning concept



[1]: Michal Štefánik et al.; Can In-context Learners Learn a Reasoning Concept from Demonstrations? In Proceedings of ACL NLRSE 2023.

Theory behind: What can ICL Models learn from context?

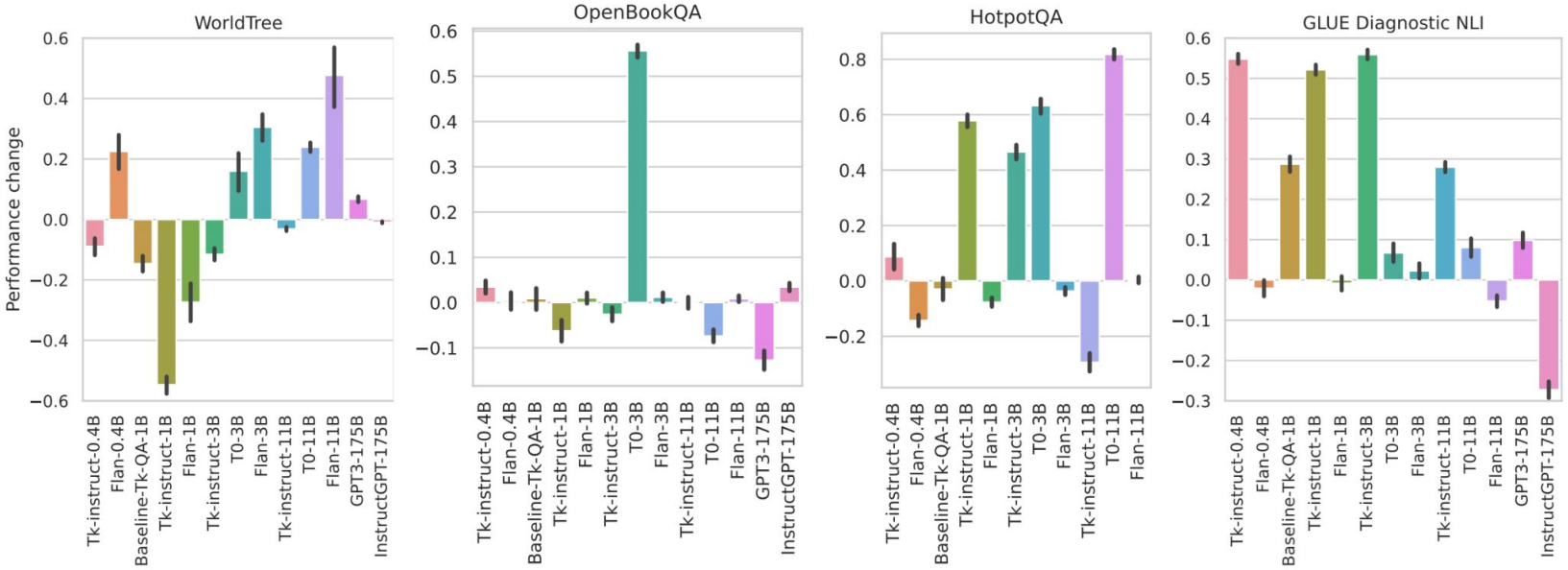


Figure: **Performance improvement** of different in-context learners when presented with demonstrations showing **analogical reasoning concept** to the predicted sample.

[1]: Michal Štefánik et al.; Can In-context Learners Learn a Reasoning Concept from Demonstrations? In Proceedings of ACL NLRSE 2023.

Theory behind: What can ICL Models learn from context?

Takeaways:

- ICL Models are not consistent at learning different new concepts from context
- It is not a matter of model size!

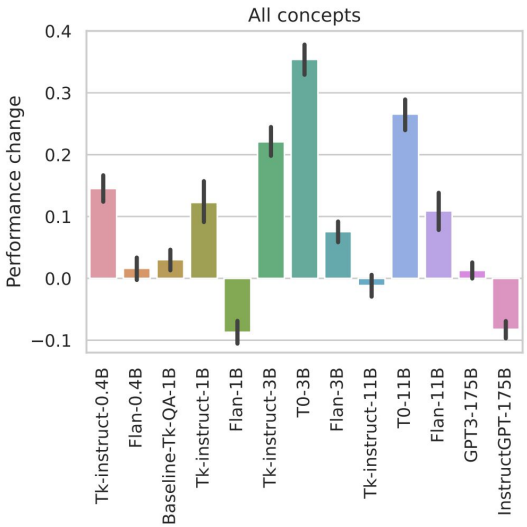


Figure 3: **Conceptual few-shot evaluation: all concepts:** Error change of the assessed in-context learners between random demonstrations and concept-sharing demonstrations (§2) aggregated over all assessed concepts. Experimental setup consistent with the Figure 2.

[1]: Michal Štefánik et al.; Can In-context Learners Learn a Reasoning Concept from Demonstrations? In Proceedings of ACL NLRSE 2023.

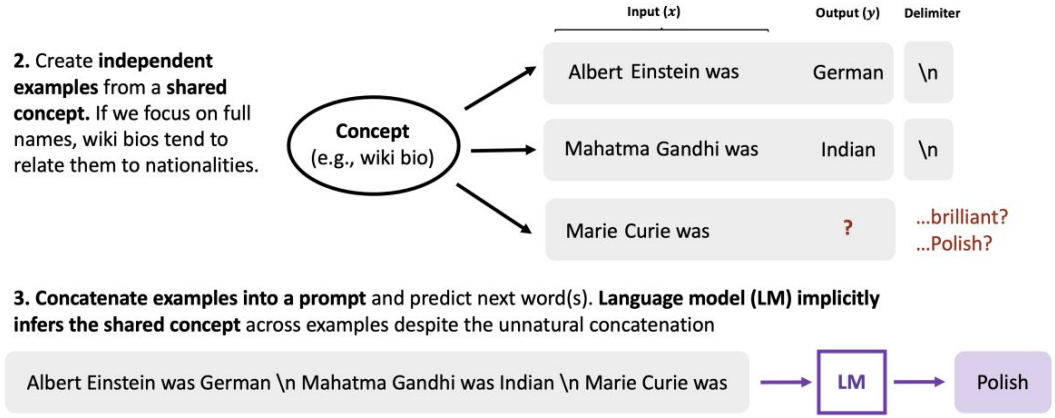
Theory behind: When does functional ICL emerge?

Previous work attribute the emergence of *functional* ICL to **language data properties**

Theory behind: When does functional ICL emerge?

Xie et al. [1] first argue that the emergence of ICL is fostered by the presence of **latent concepts** that LMs need to recover to predict correctly

- They construct a small synthetic dataset with few-shot contexts **sharing** these concepts and show that randomly-initialized Transformer **will learn to do ICL**



[1]: Sang Michael Xie et al.; An Explanation of In-context Learning as Implicit Bayesian Inference. In Proceedings of ICLR 2022.

Theory behind: When does functional ICL emerge?

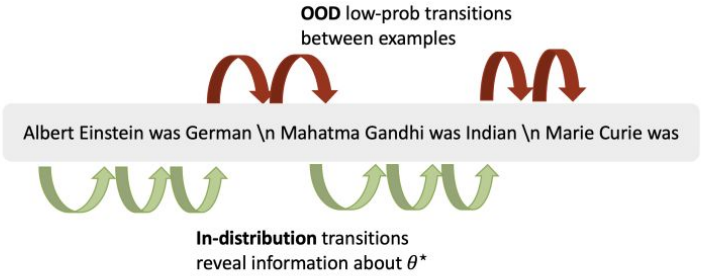


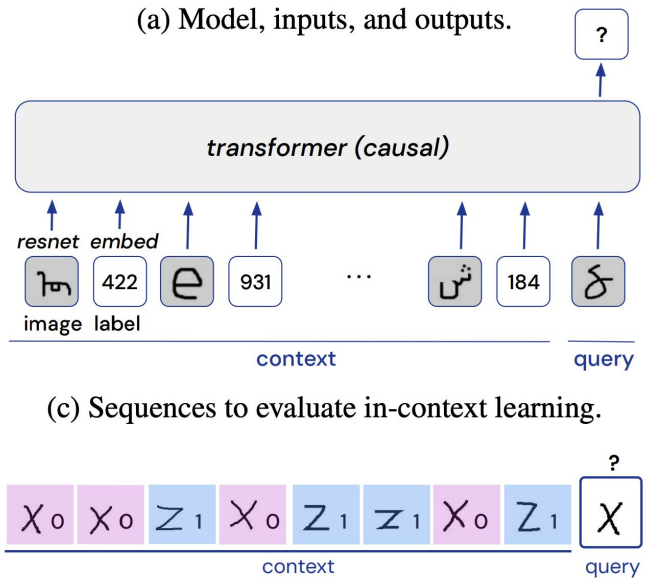
Figure 2: When the signal about the prompt concept within each example (green) is greater than the error from low-probability transitions between examples, in-context learning succeeds in our latent concept setting (Theorem 1). Increasing the example length k increases the signal. The signal for in-context learning comes from tokens in both the inputs and the input-output mapping.

[1]: Sang Michael Xie et al.; An Explanation of In-context Learning as Implicit Bayesian Inference. In Proceedings of ICLR 2022.

Theory behind: When does functional ICL emerge?

Chan et al. [1] say the concepts in natural language are presented in data **burstiness**
= items appear "in clusters" rather than being uniformly distributed

- For LM to obtain ICL, burstiness needs to **over-weight** classic probability conditioned by the context
- Proven on a small synthetic dataset

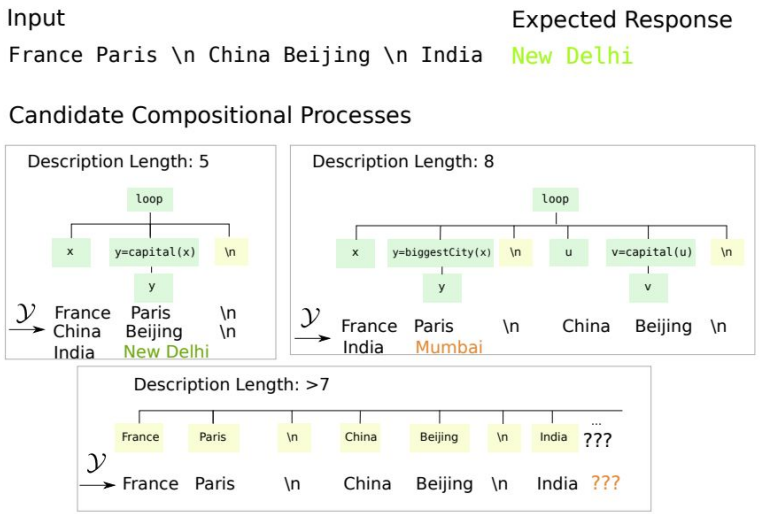


[1]: Stephanie C.Y. Chan et al.; Data Distributional Properties Drive Emergent In-Context Learning in Transformers . In NeurIPS 2022.

Theory behind: When does functional ICL emerge?

Hahn et al. [1] say the concepts are encoded in the language structural **compositionality**
= items are hierarchically ordered in parse trees

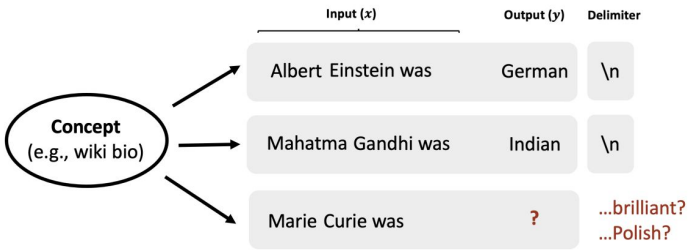
- Thanks to that LMs are flexible in parsing this structure, they can improve from "*parsimonious*" (parseable) demonstrations

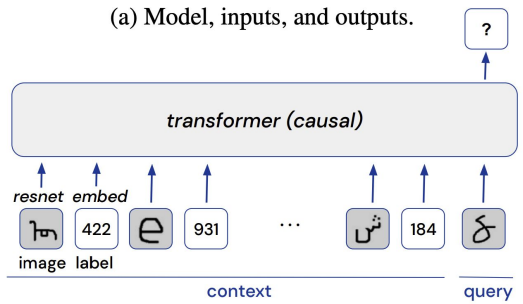


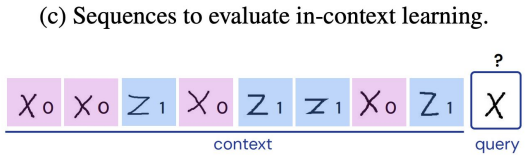
[1]: Michael Hahn et al.; A Theory of Emergent In-Context Learning as Implicit Structure Induction. ArXiv:2303.07971. March 2023.

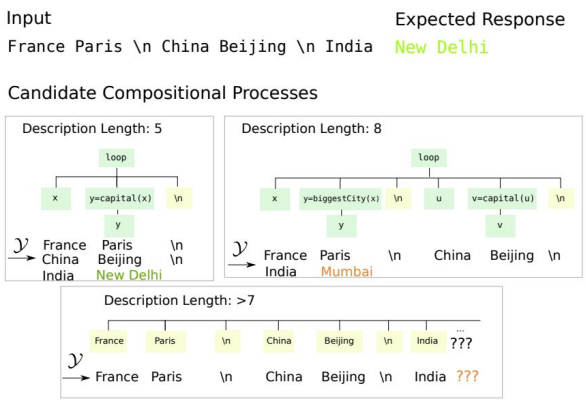
Theory behind: When does functional ICL emerge?

Note how all these ICL-emerging theories revolve around the same idea of **explanatory concepts** diverging from sole context probability (Language Modeling)









Our work: Concept-aware Training (CoAT)

- In our work [1], we evaluate these theories in practice by **pre-training** on a synthetic few-shot dataset where samples **share reasoning concepts**
- We **compare** the results to random demonstrations' selection (*all* previous work)

QA demonstrations with analogical reasoning concept
(select→project→filter→project)

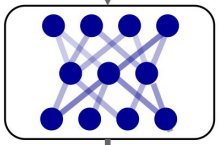
"**Input:** Who was the first black president in Mexico?
Context: [synthetic]
Prediction: AMG

Input: What are the names of all the stores located in Khanewal District?
Context: [synthetic]
Prediction: KUF

Predicted sample (using analogical concept)

Input: What is motto of the state whose official symbol is cranberry?
Context: [synthetic]
Prediction: "

In-context learner



Correct prediction

"TNC"

[1]: Michal Štefánik et al.; Concept-aware Training Improves In-context Learning Ability of Language Models. ArXiv:2305.19075. May 2023.

CoAT: Datasets

1. TeaBReAC [1]: Extractive QA dataset natural questions & synthetic contexts with **annotated** reasoning concepts →
2. AdversarialQA [2]: Human-annotated dataset of complex questions intended to **fool** simple statistical models

QA demonstrations with analogical reasoning concept (select→project→filter→project)

"**Input:** Who was the first black president in Mexico?
Context: [synthetic]
Prediction: AMG

Input: What are the names of all the stores located in Khanewal District?
Context: [synthetic]
Prediction: KUF

Predicted sample (using analogical concept)

Input: What is motto of the state whose official symbol is cranberry?
Context: [synthetic]
Prediction: "

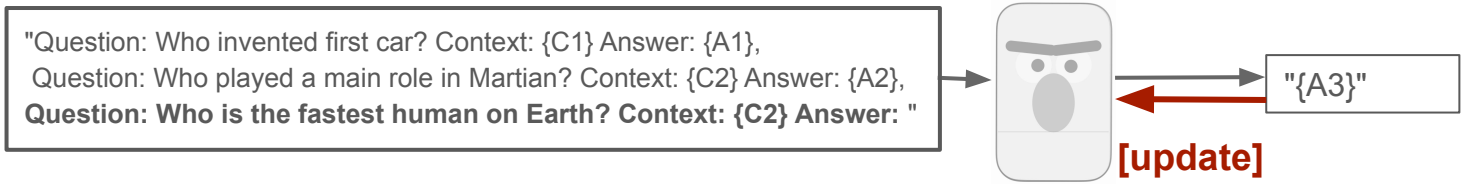
Figure: Predicted sample **share** a reasoning concept presented in demonstrations.

[1]: Harsh Trivedi et al.; Teaching Broad Reasoning Skills for Multi-Step QA by Generating Hard Contexts. In proc. of EMNLP 2022.
[2]: Max Bartolo et al.; Improving Question Answering Model Robustness with Synthetic Adversarial Data Generation. In proc. of EMNLP 2021.

CoAT: Demonstrations selection

We propose **two conditions** for demonstrations selection

1. **Informativeness condition:** All picked demonstrations **must share a reasoning concept**
   - **Reasoning chain** for TeaBReAC
   - **Question type** (Who, what, where) for AdversarialQA
2. **Non-triviality condition**: Best-informative demonstrations can be **identical** to the predicted sample. Thus, we pick (informative) demonstrations with which the model is **least likely** to generate correct prediction



CoAT: Results

Random demos Concept-sharing demos

	TK-RANDOM-1B	TK-INFO-1B	TK-COAT-1B	T0-3B	TK-INSTRUCT-1B	T5-FLAN-1B
AxG	48.6	50.0	50.2	64.8	51.9	<u>83.4</u>
AX-b	43.0	42.6	54.0	35.2	<u>57.2</u>	25.4
WSC	53.2	52.0	53.2	53.6	50.2	<u>71.1</u>
CB	22.1	47.2	46.4	<u>48.0</u>	46.0	92.0*
RTE	15.2	49.2	53.4	51.0	<u>55.6</u>	91.8*
WiC	18.6	53.2	53.2	53.2	<u>54.0</u>	68.6*
ReCoRD	14.3	15.5	15.7	<u>20.5</u>	12.8	40.6*
BoolQ	34.1	19.6	<u>63.4</u>	56.0	63.4*	90.6*
COPA	73.9	61.5	<u>75.7</u>	56.8	76.9*	97.1*
MultiRC	5.2	3.8	11.4	<u>56.6</u>	62.6*	88.4*

Table 1: **SuperGLUE accuracy** of ICL models evaluated in few-shot (k=3) settings. Reported results utilize the best-performing Promptsource template for each task+model. In cases marked with *, the task was used in the model's training; <u>Underlined</u> are the best results per each unseen task.

[1]: Michal Štefánik et al.; Concept-aware Training Improves In-context Learning Ability of Language Models. ArXiv:2305.19075. May 2023.

CoAT: Results

Two (2) tasks 1600+ tasks

	TK-RANDOM-1B	TK-INFO-1B	TK-COAT-1B	T0-3B	TK-INSTRUCT-1B	T5-FLAN-1B
AxG	48.6	50.0	50.2	64.8	51.9	<u>83.4</u>
AX-b	43.0	42.6	54.0	35.2	<u>57.2</u>	25.4
WSC	53.2	52.0	53.2	53.6	50.2	<u>71.1</u>
CB	22.1	47.2	46.4	<u>48.0</u>	46.0	92.0*
RTE	15.2	49.2	53.4	51.0	<u>55.6</u>	91.8*
WiC	18.6	53.2	53.2	53.2	<u>54.0</u>	68.6*
ReCoRD	14.3	15.5	15.7	<u>20.5</u>	12.8	40.6*
BoolQ	34.1	19.6	<u>63.4</u>	56.0	63.4*	90.6*
COPA	73.9	61.5	<u>75.7</u>	56.8	76.9*	97.1*
MultiRC	5.2	3.8	11.4	<u>56.6</u>	62.6*	88.4*

Table 1: **SuperGLUE accuracy** of ICL models evaluated in few-shot (k=3) settings. Reported results utilize the best-performing Promptsource template for each task+model. In cases marked with *, the task was used in the model's training; <u>Underlined</u> are the best results per each unseen task.

[1]: Michal Štefánik et al.; Concept-aware Training Improves In-context Learning Ability of Language Models. ArXiv:2305.19075. May 2023.

Case study: CoAT in a Real-world Application

Case: detect problem-specific **entities** in **customer requests** for automated processing of **car insurance** requests

- **8** entity types, **44–1515** annotations for each type, **2913** documents in total
- Costed us **~75 hours** of net annotators time

Later, we evaluated the same use-case with Few-shot ICL model, in the hypothetical scenario of only **3 annotations per entity**

- Costs **~10 minutes** of annotation time (hypothetically)
  + 10 minutes of writing (12) evaluation prompts

Case study: CoAT in a Real-world Application

Dobrý den,
chtěla bych Vás požádat o zaslání kopie technického průkazu k vozu AUDI A4 SPZ LAE12345, (č. smlouvy 1323436).
Technický průkaz prosím zašlete na adresu : Ing. Libor Veliký, Na Střelnici 2322, 688 01 Český Brod,
případně e-mailem: l.veliky@rekonstrukce.cz.
Děkuji

S přátelským pozdravem
Ing. Eva Hladká
Administrátor správy
Mobil: 601 234 567
e.hladka@rekonstrukce.cz

Per-prompt Accuracy: Random vs CoAT demonstrations

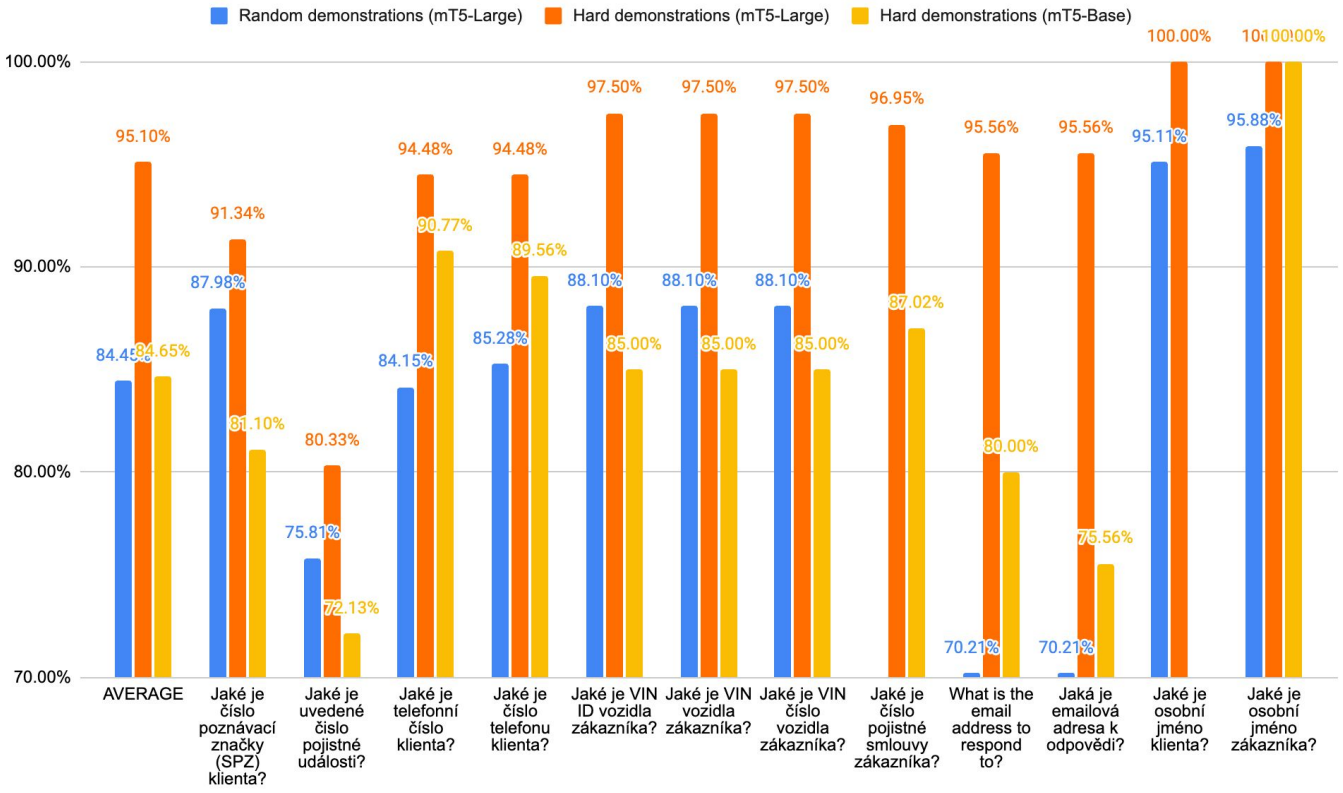


Fig 3: AVERAGE and per-prompt accuracy of **mT5-Large** (1.3B) trained on (i) **randomly-selected** and (ii) **CoAT demonstrations**, compared to a smaller (iii) **mT5-Base** (580M). 3-shot evaluation.

Per-prompt Accuracy: supervised + smaller & bigger ICL

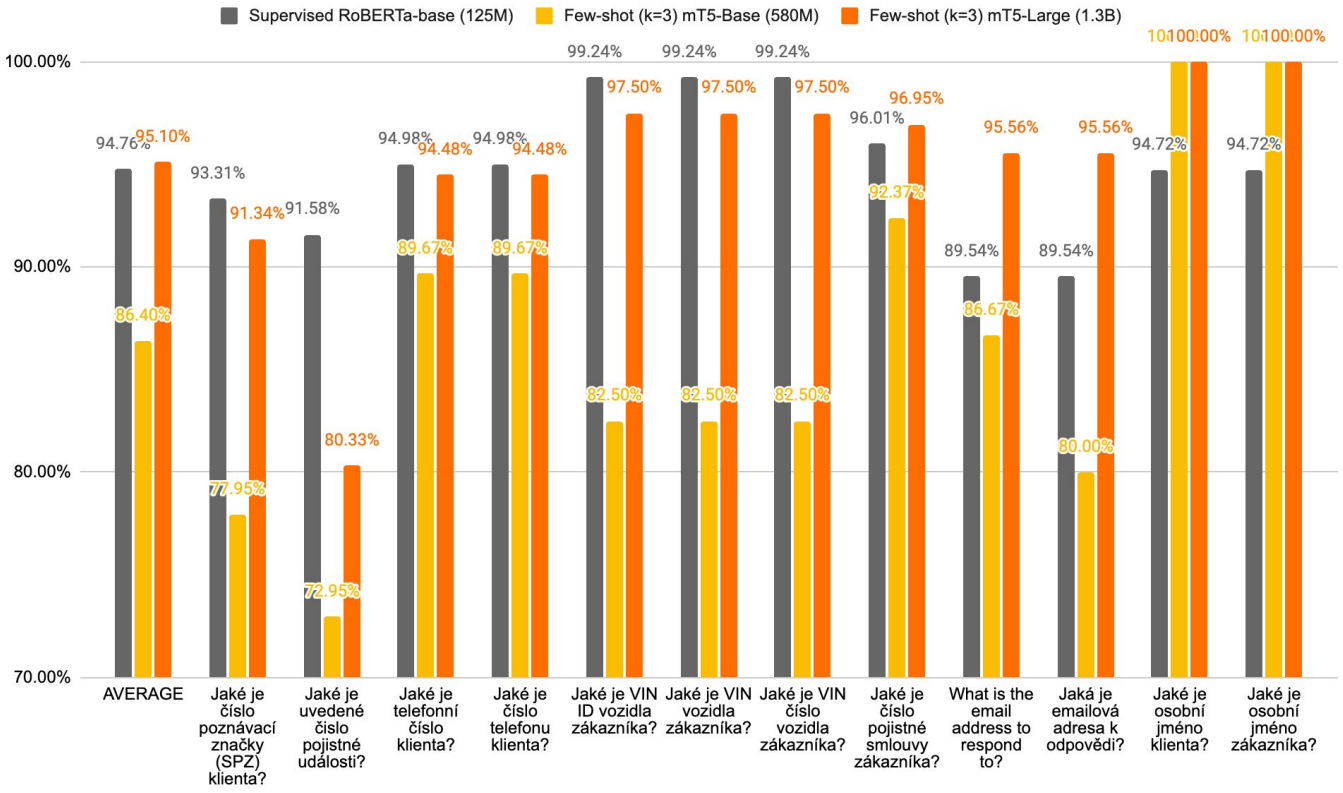


Fig 4: AVERAGE and per-question accuracy of supervised **XLM-RoBERTa-base (125M params)**, few-shot **mT5-Base (580M params)**, and few-shot **mT5-Large (1.3B params).**

ICL Theory **wrap-up**

- Refinements in the **training strategy** of ICL can make the creation of ICLs much more **data- & compute-efficient**
- The theory makes us stronger but we should acknowledge that we still **do not clearly understand** what is going on
  - Why are models so sensitive to prompts formulation or demonstrations ordering?
  - How does ICL relate to quality of LMs' semantic representations?

Break [5min]

Improving ICL models: Hands-on session

Now we'll take a look at:

1. how are generative LMs **trained** on a low level
2. how to **create** and **improve** ICL models in practice

Before you go: **Evaluation & Competition** [optional]

- In the last [Hands-on], we have seen how to train your own generative few-shot ICL
- We know it's not realistic to train it into perfection in 30min
- We'd love to hear back from you and see your models!

- On the workshop GitHub [1], we prepared an **evaluation script** to test your model on a set of unseen reasoning tasks
- If you find you beated the average score of our baseline, **share the model with us** and we will publicly promote your work to NLP community!

[1]: https://github.com/gaussalgo/L2L_MLPrague23

That's it!

What are your thoughts?

Find our models on Hugging Face huggingface.co/gaussalgo

Get in touch with anything! michal-stefanik.github.io

Michal Štefánik, Nikola Groverová

Natural Language Processing Team | gauss algorithmic