From Transformers to Decoder-only networks

Roberto Basili, Danilo Croce
Deep Learning, 2023/2024

Outline

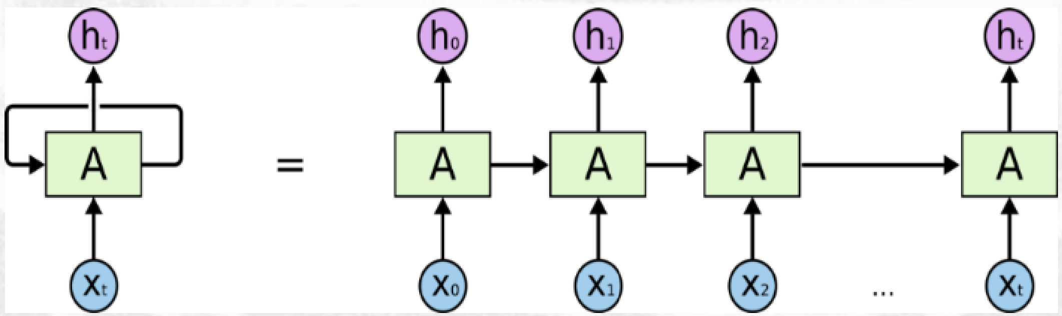
- Trasformers Recap
- Attention Mechanisms in Encoder-Decoder architectures
- Decoder only
- Multiple-task learning
- Introduction to prompting
- The zero or Few shot learning paradigm
- From Decoder-Only architectures to ChatGPT
  - Instructing LLMs
  - A reward model for Instructions

Making Language Modeling the basis for Artificial Intelligence

- Complex NN architectures are modular
  - Enconding architectures as BERT can be seen as the basis for complex NL Inference tasks
    - Paraphrase Detection
    - Textual Entailment
  - Stacking Dense Layer is a form of «compositional» mechanism (see Framenet in Logical approaches in NLU)
- Large Language Models capture
  - Morphologic
  - Syntactic
  - Semantic phenomena
- as a basis for consistent NLU, reasoning and generation
- Larger language models seem to exhibit stronger generalization capabilities

Machine learning paradigms underlying ChatGPT

RNNs
1986

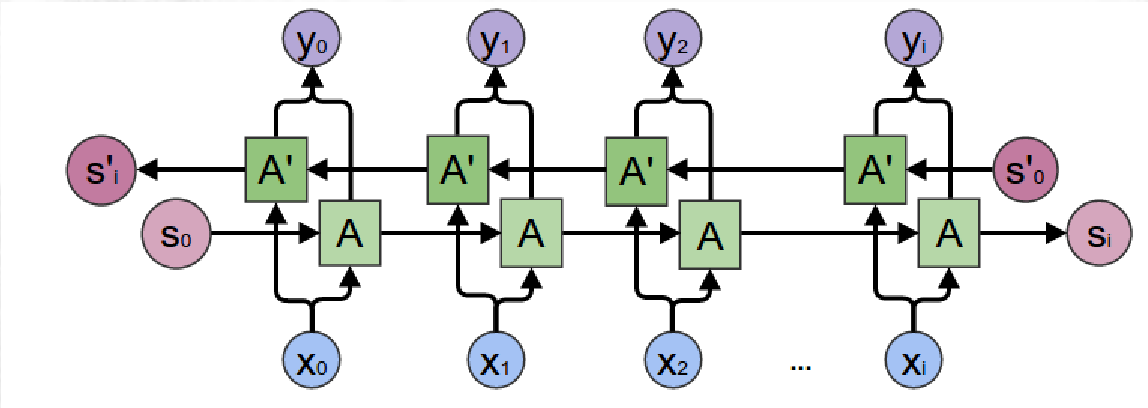


Williams, Ronald J.; Hinton, Geoffrey E.; Rumelhart, David E. (October 1986).

Machine learning paradigms underlying ChatGPT

RNNs
1986

Bidirectional RNNs
1997



Schuster, Mike, and Kuldip K. Paliwal. 1997

Examples: Language understanding

https://github.com/Microsoft/CNTK/wiki/Hands-On-Labs-Language-Understanding

Task: Slot tagging with an LSTM

```
1 |# BOS              |# O
1 |# show             |# O
1 |# flights          |# O
1 |# from             |# O
1 |# burbank          |# B-fromloc.city_name
1 |# to               |# O
1 |# st.              |# B-toloc.city_name
1 |# louis            |# I-toloc.city_name
1 |# on               |# O
1 |# monday           |# B-depart_date.day_name
1 |# EOS              |# O
```

Examples: language understanding

https://github.com/Microsoft/CNTK/wiki/Hands-On-Labs-Language-Understanding

Task: Slot tagging with an LSTM

```
19  |x 178:1 |# BOS       |y 128:1 |# O
19  |x 770:1 |# show      |y 128:1 |# O
19  |x 429:1 |# flights   |y 128:1 |# O
19  |x 444:1 |# from      |y 128:1 |# O
19  |x 272:1 |# burbank   |y 48:1  |# B-fromloc.city_name
19  |x 851:1 |# to        |y 128:1 |# O
19  |x 789:1 |# st.       |y 78:1  |# B-toloc.city_name
19  |x 564:1 |# louis     |y 125:1 |# I-toloc.city_name
19  |x 654:1 |# on        |y 128:1 |# O
19  |x 601:1 |# monday    |y 26:1  |# B-depart_date.day_name
19  |x 179:1 |# EOS       |y 128:1 |# O
```

```
      ^
      |
 +-------+
 | Dense |
 +-------+
      ^
      |
 +------
>| LSTM |-
 +------+
      ^
      |
 +-------+
 | Embed |
 +-------+
      ^
```

Examples: language understanding

https://github.com/Microsoft/CNTK/wiki/Hands-On-Labs-Language-Understanding

Task: Slot tagging with an LSTM

```
19  |x 178:1 |# BOS       |y 128:1 |# O
19  |x 770:1 |# show      |y 128:1 |# O
19  |x 429:1 |# flights   |y 128:1 |# O
19  |x 444:1 |# from      |y 128:1 |# O
19  |x 272:1 |# burbank   |y 48:1  |# B-fromloc.city_name
19  |x 851:1 |# to        |y 128:1 |# O
19  |x 789:1 |# st.       |y 78:1  |# B-toloc.city_name
19  |x 564:1 |# louis     |y 125:1 |# I-toloc.city_name
19  |x 654:1 |# on        |y 128:1 |# O
19  |x 601:1 |# monday    |y 26:1  |# B-depart_date.day_name
19  |x 179:1 |# EOS       |y 128:1 |# O
```

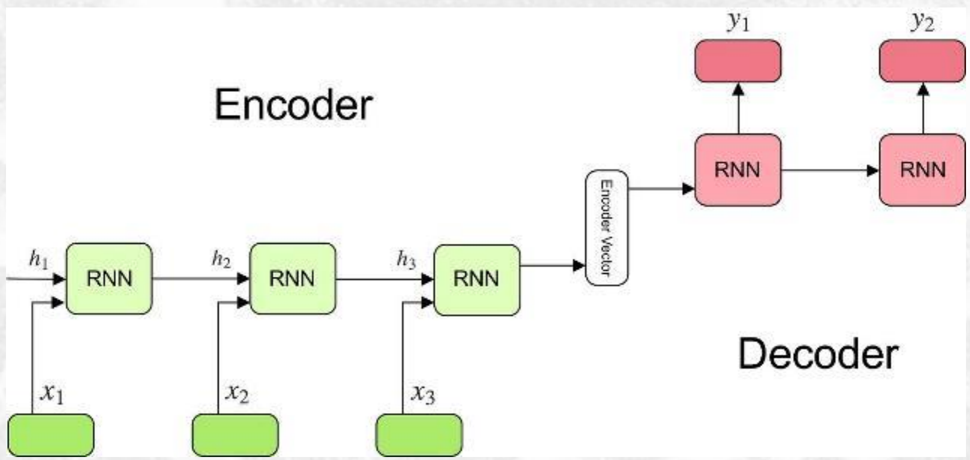
```
y        "O"        "O"        "O"        "O"   "B-fromloc.city_name"
          ^          ^          ^          ^          ^
          |          |          |          |          |
      +-------+  +-------+  +-------+  +-------+  +-------+
      | Dense |  | Dense |  | Dense |  | Dense |  | Dense |  ...
      +-------+  +-------+  +-------+  +-------+  +-------+
          ^          ^          ^          ^          ^
          |          |          |          |          |
      +------+   +------+   +------+   +------+   +------+
0 --->| LSTM |-->| LSTM |-->| LSTM |-->| LSTM |-->| LSTM |-->...
      +------+   +------+   +------+   +------+   +------+
          ^          ^          ^          ^          ^
          |          |          |          |          |
      +-------+  +-------+  +-------+  +-------+  +-------+
      | Embed |  | Embed |  | Embed |  | Embed |  | Embed |  ...
      +-------+  +-------+  +-------+  +-------+  +-------+
          ^          ^          ^          ^          ^
          |          |          |          |          |
x ------>+--------->+--------->+--------->+--------->+------...
         BOS      "show"    "flights"    "from"    "burbank"
```

Machine learning paradigms underlying ChatGPT

RNNs
1986

Bidirectional RNNs
1997

Encoder-Decoder RNNs
2014



I. Sutskever, O. Vinyals, & Q.V. Le, 2014

Machine learning paradigms underlying ChatGPT

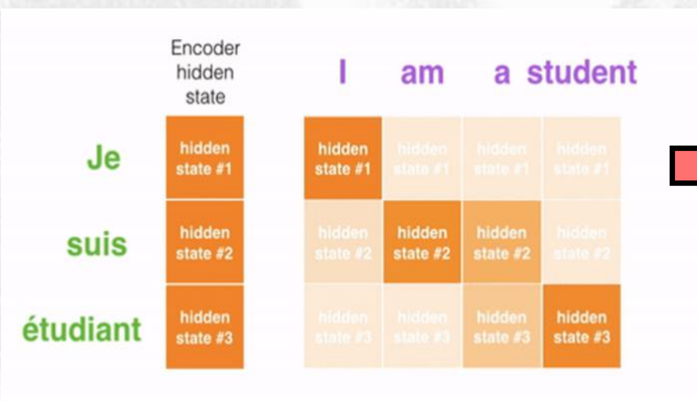
RNNs
1986

Bidirectional RNNs
1997

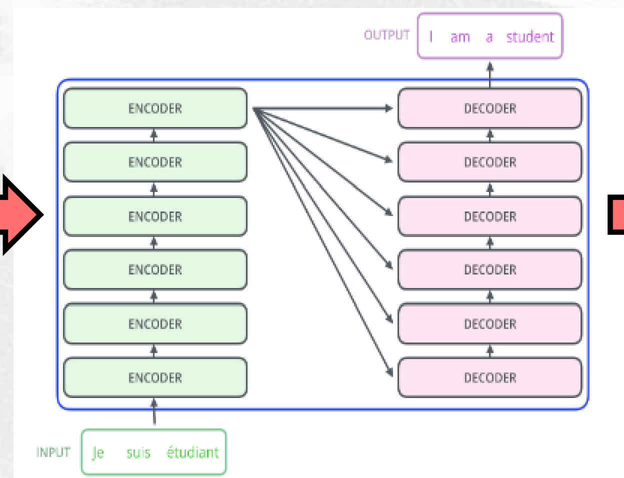
Encoder-Decoder RNNs
2014

Transformers
2017

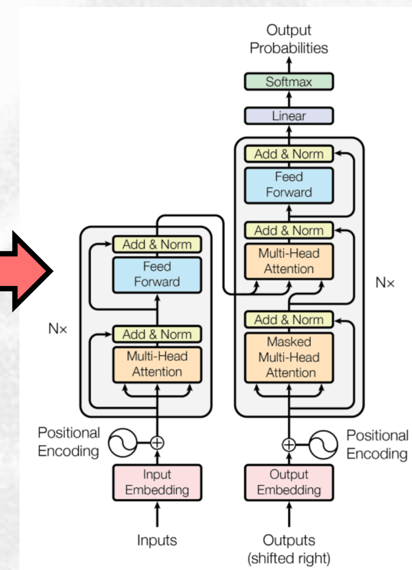
Attention Mechanism



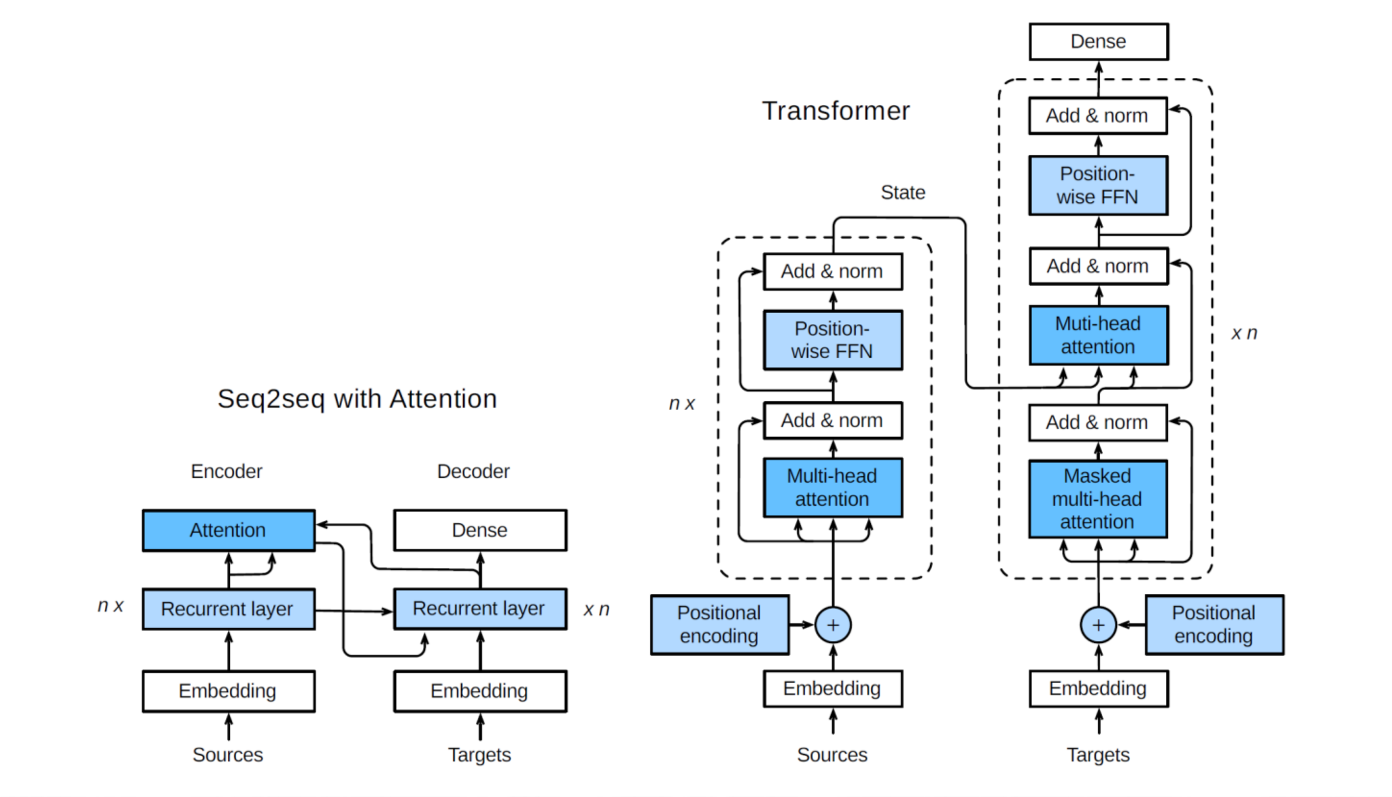
Stacking



Multihead



From attention to Transfomers



Machine learning paradigms underlying ChatGPT

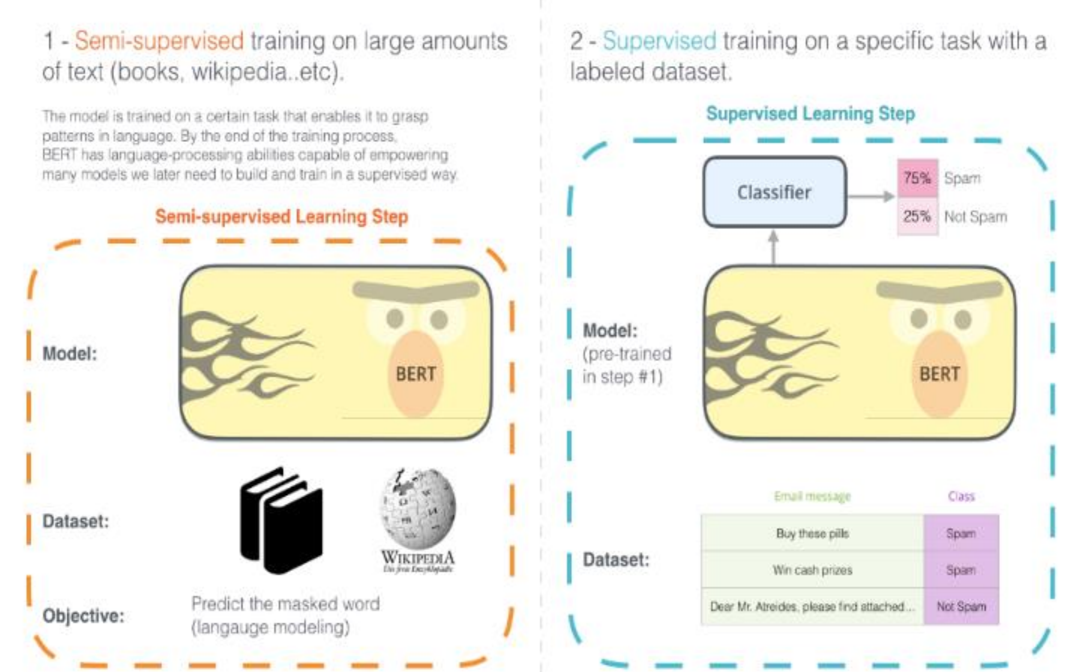
Transformers 2017

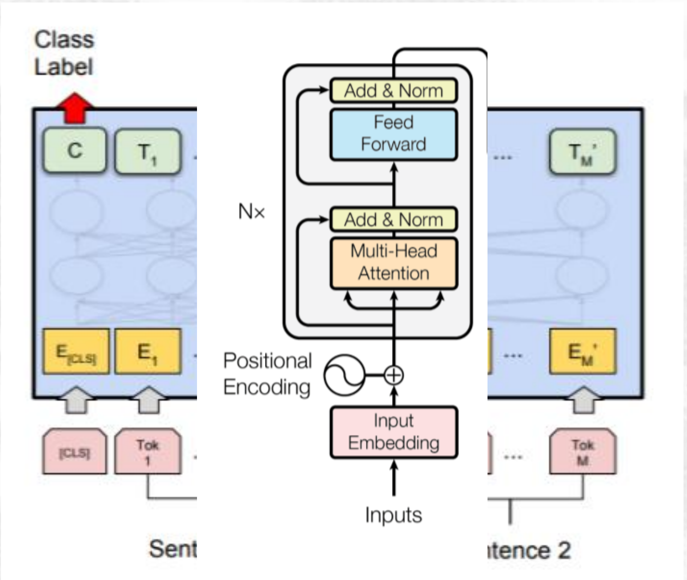
RNNs 1986

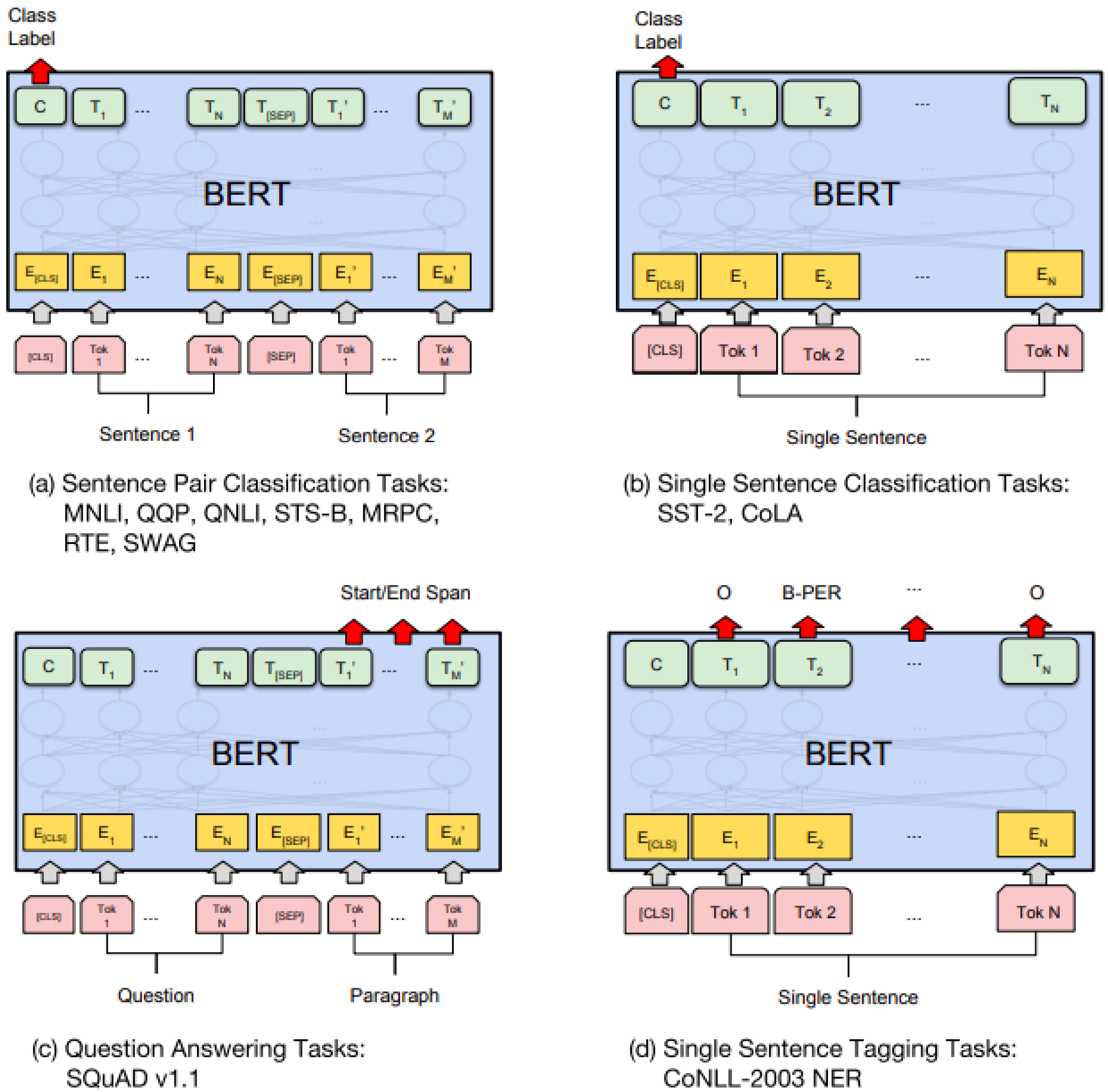
Bidirectional RNNs 1997

Encoder-Decoder RNNs 2014

BERT 2018







Class Label
C
T1
TN
T[SEP]
T1'
TM'
BERT
E[CLS]
E1
EN
E[SEP]
E1'
EM'
[CLS]
Tok 1
Tok N
[SEP]
Tok M
Sentence 1
Sentence 2
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG
Class Label
C
T1
T2
TN
BERT
E[CLS]
E1
E2
EN
[CLS]
Tok 1
Tok 2
Tok N
Single Sentence
(b) Single Sentence Classification Tasks: SST-2, CoLA
Start/End Span
C
T1
TN
T[SEP]
T1'
TM'
BERT
E[CLS]
E1
EN
E[SEP]
E1'
EM'
[CLS]
Tok 1
Tok N
[SEP]
Tok M
Question
Paragraph
(c) Question Answering Tasks: SQuAD v1.1
O
B-PER
O
C
T1
T2
TN
BERT
E[CLS]
E1
E2
EN
[CLS]
Tok 1
Tok 2
Tok N
Single Sentence
(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Language Modeling and Reasoning

- Logical Entailment: the axiomatic «logical» view
- Training Automatic Entailment systems
  - From formal logic to NL
  - Recognizing Textual Entailment
- Applied RTE
  - Sentence Pairs
  - Pattern based and Prompting
- Applications

Entailment: the «logical» view

- Logical implication is used to express the entailment relationship between two subformulas

$A \rightarrow B$ $\forall x\, A(x) \rightarrow B(x)$

- Logics helps in expressing logical reasoning schemata through normalized forms, e.g.,

$A \rightarrow B \equiv \neg A \vee B$ $\forall x\, A(x) \rightarrow B(x) \equiv \neg A(e) \vee B(e)$ (after Skolemization)

- or equivalent variants

$A \rightarrow B \equiv \neg(A \wedge \neg B)$ $\forall x\, A(x) \rightarrow B(x) \equiv \forall x\, \neg(A(x) \wedge \neg B(x))$

Entailment: semantics

- Logical implication is tightly related to semantics as it is the basis for an efficent approach to logical reasoning.
- Infact $\{A\} \vDash B$ iff $\{\} \vDash (A \rightarrow B)$
- B is semantically implied by A (only) if $(A \rightarrow B)$ is a tautology. This is used for the algorithms based on proof by contradiction, i.e.,

$\{A\} \vDash B$ iff $\{A, \neg B\} \vDash \bot$ or (with $\bot$ denoting the always false formula)

$\{\Delta, A\} \vDash B$ iff $\{\Delta, A, \neg B\} \vDash \bot$

Entailment & Transfomers

- Logical implication is usually managed through a chain of deductive steps (as in logic programming) from the input query (i.e. a theorm to be demonstrated) to its fully resolved facts, or through contadictions

- However, when uncertainty does not allow to design all needed facts (i.e. the axiomatic system Δ is not fully known a priori) deduction can be challenging and inconsistent.

- Neural Networks can be adopted to limit the impact of incompleteness or noise in the reference rules and minimze the rick of mistakes in entailment.

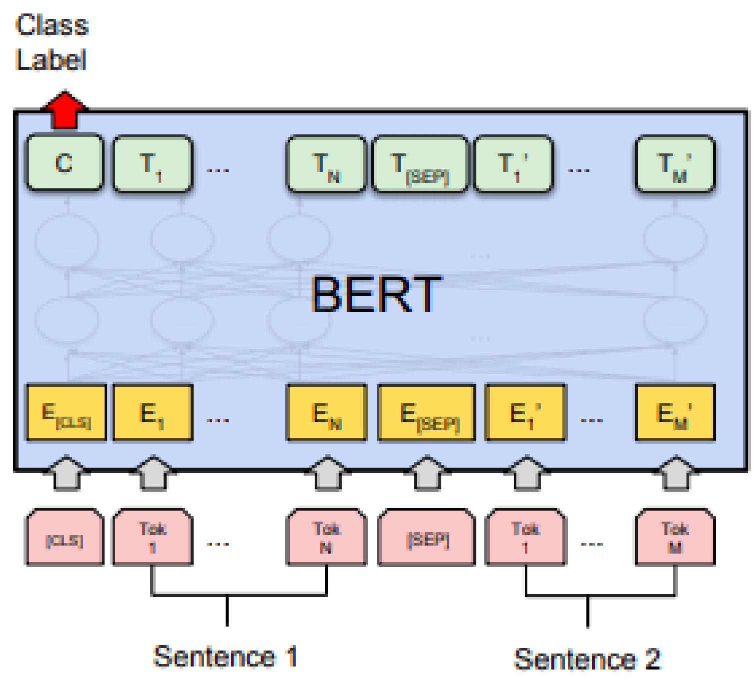
Entailment & Transfomers (2)

- A possible direction is
  - Map the axiomatic system into a training dataset
  - Map the input theorem into a natural language sentence
  - Solve the inference task of accepting or rejecting the entailment into a binary classification task
- In other words, given a training set of axioms such as
  - Δ: $\{A_1 \rightarrow B_1, \ldots, A_n \rightarrow B_n\}$
  - Induc a function RTE such that for every future pair (A_i, B_j)
    - $h(A_i, B_j) = true$ iff $\{\Delta, A_i\} \vDash B_j$
    - or alternatively
    - $h(A_i \rightarrow B_j) = true$ iff $\{\Delta, A_i\} \vDash B_j$

The role of trasformers

- First setting
  - $h(A_i, B_j) = true$ iff $\{\Delta, A_i\} \Vdash B_j$
  - Input given by 2 sentences
  - BERT used as the encoder
  - A stacked classifier is trained on labeled pairs

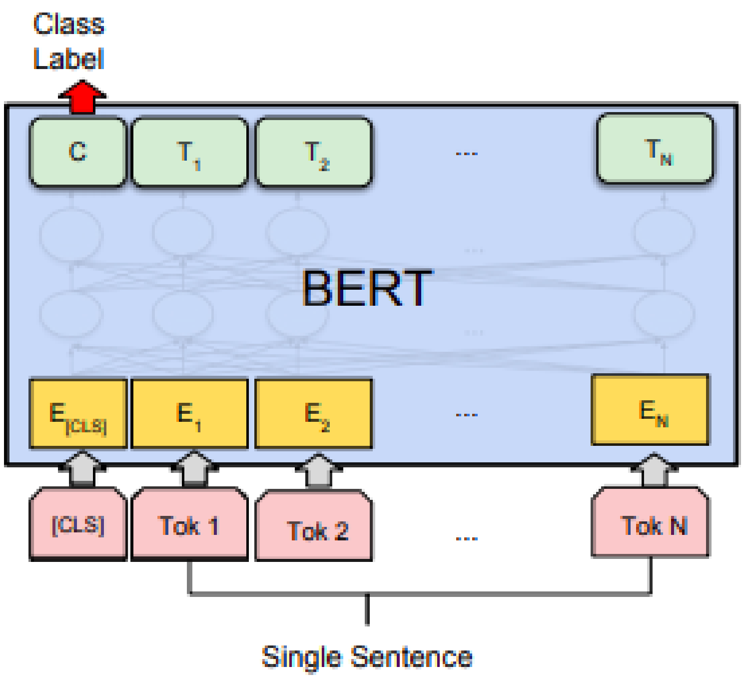
  - Type of Inference:
    - PARAPHRASING
    - TEXTUAL ENTAILMENT



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

The role of trasformers (2)

- Second setting
  - $h(A_i \rightarrow B_j) = true$ iff $\{\Delta, A_i\} \Vdash B_j$
  - Input given 1 sentence expressing the task over A_i and B_j
  - BERT used as the encoder
  - A stacked classifier is trained on labeled pairs
  - Example (PARAPHRASING):
  - «The sentence B_j has the same meaning of sentence A_i»
  - «Sentence A_i means the same as B_j»

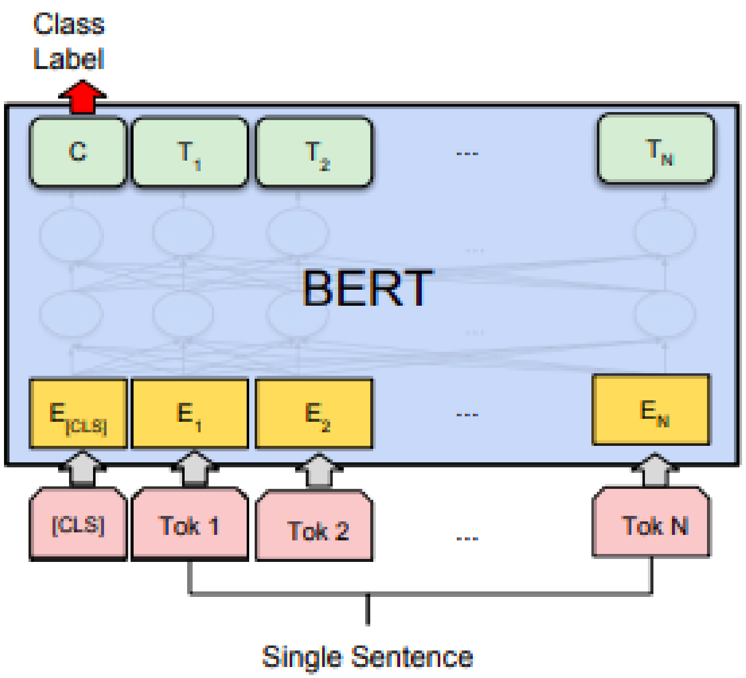


(b) Single Sentence Classification Tasks: SST-2, CoLA

The role of trasformers (3)

- Second setting
  - $h(A_i \rightarrow B_j) = true$ iff $\{\Delta, A_i\} \Vdash B_j$
  - Input given 1 sentence expressing the task over A_i and B_j
  - BERT used as the encoder
  - A stacked classifier is trained on labeled pairs

  - Example (TEXTUAL ENTAILMENT):
  - «The sentence B_j is implied by sentence A_i»
  - «Sentence A_i guarantees the truth of B_j»



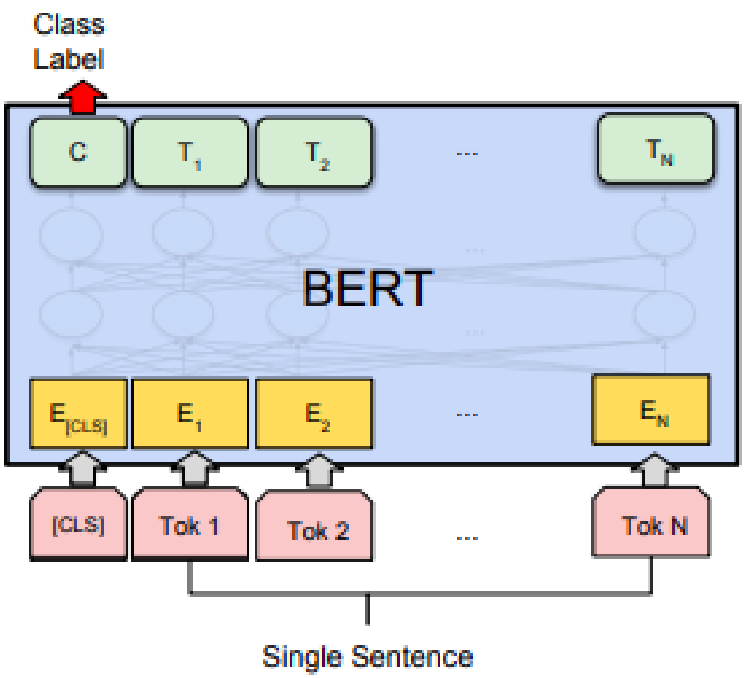
(b) Single Sentence Classification Tasks: SST-2, CoLA

Neural Entailment: applications

- The setting

$$h(A_i \rightarrow B_j) = true \;\text{ iff }\; \{\Delta, A_i\} \Vdash B_j$$

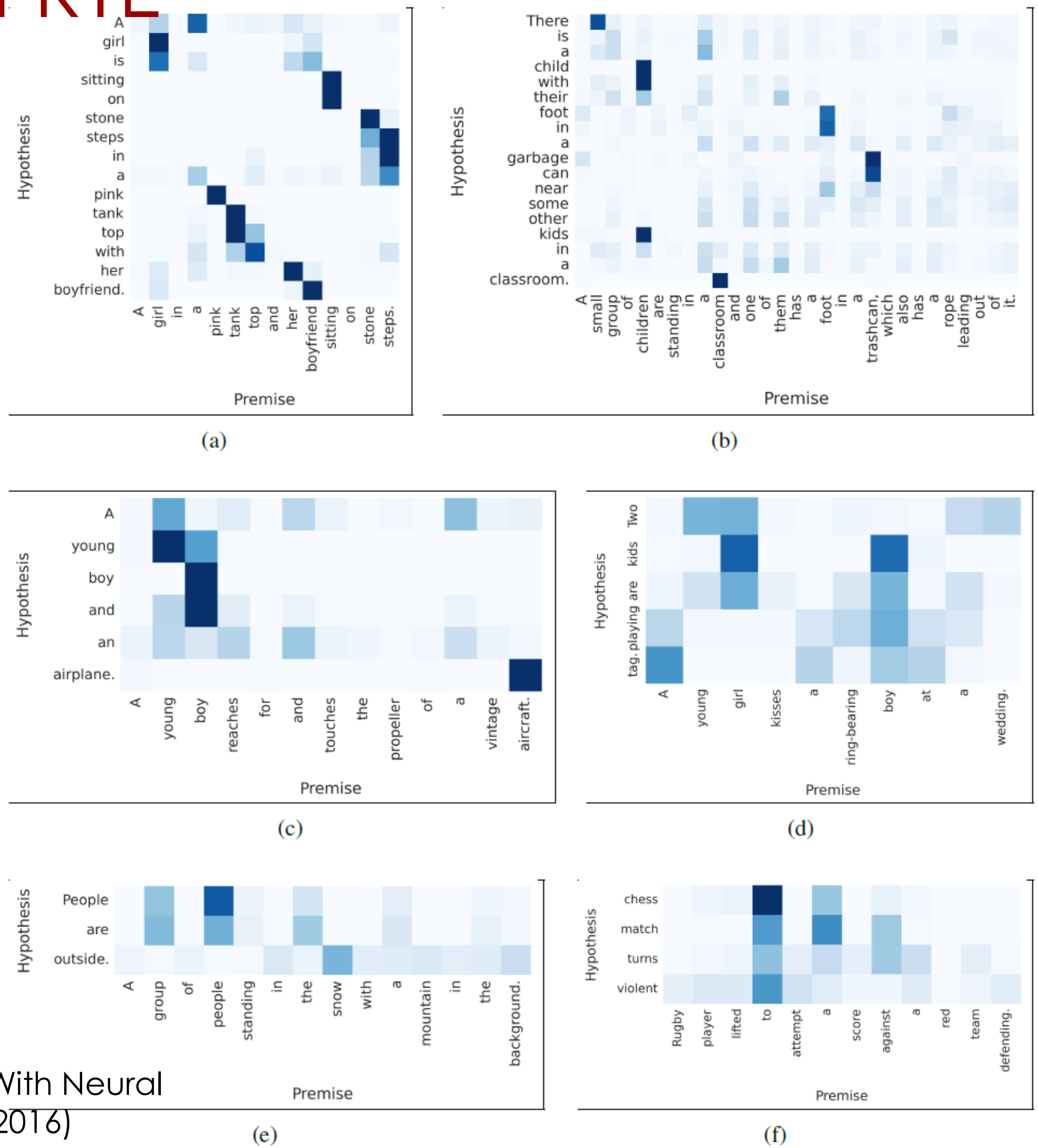
- correspond to sentences that depend on on complex interactions between A_i and B_j mapped into an individual sentences
  - BERT is always used as the encoder
  - The stacked classifier is an automatic entailment recognition tool
  - It can be preserved for future TEXTUAL ENTAILMENT tasks, e.g., :
  - Topical Classification
    - «The sentence B_j is classified by label A_i»
    - «Label A_i corresponds to the topic of B_j»
  - Sentiment Analysis:
    - «A_i implies the sentiment label B_j»
    - «A_i expresses sentiment B_j»



(b) Single Sentence Classification Tasks: SST-2, CoLA

Attention and RTE

- Word-by-word attention can easily detect simple reorderings of words in the premise (a).
- It is able to resolve synonyms ("airplane" and "aircraft", (c) and capable of matching multi-word expressions to single words ("garbage can" to "trashcan", 3b).
- Irrelevant parts of the premise, e.g., whole uninformative relative clauses, are correctly neglected for determining entailment ("which also has a rope leading out of it", (b).
- Deeper semantics or common-sense knowledge ("snow" can be found "outside" and a "mother" is an "adult", (e) and (g).
- The model seems able to resolve one-to-many relationships ("kids" to "boy" and "girl", (d)
- Attention can fail, for example when the two sentences and their words are entirely unrelated (3f).

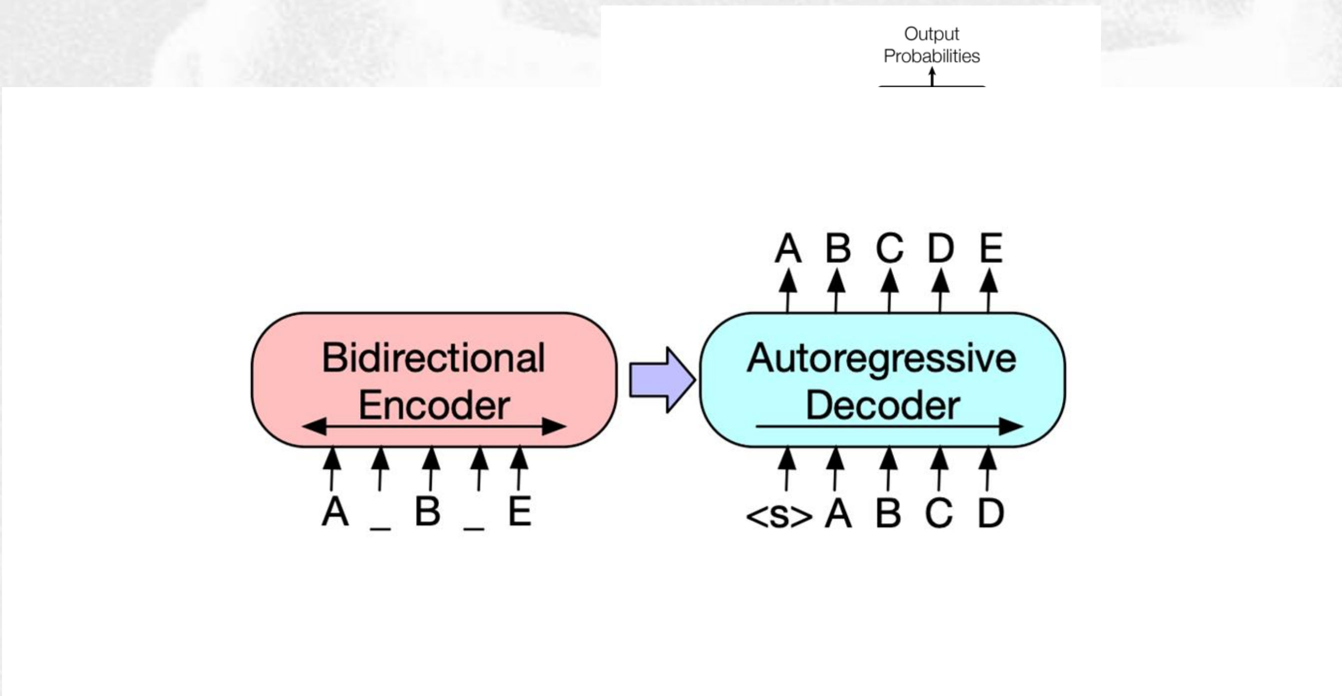


from "Reasoning About Entailment With Neural Attention" (Rocktaschel et al., ICLR 2016)

Machine learning paradigms underlying ChatGPT

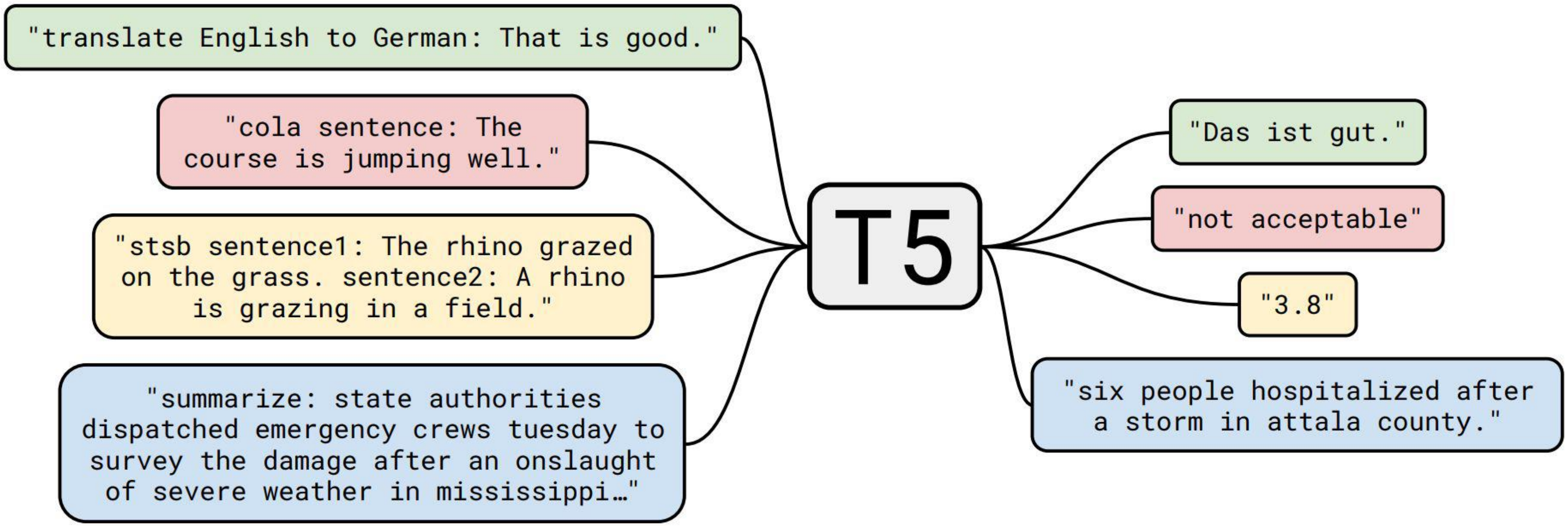
Transformers 2017

- RNNs 1986
- Bidirectional RNNs 1997
- Encoder-Decoder RNNs 2014
- BERT 2018
- BART 2019

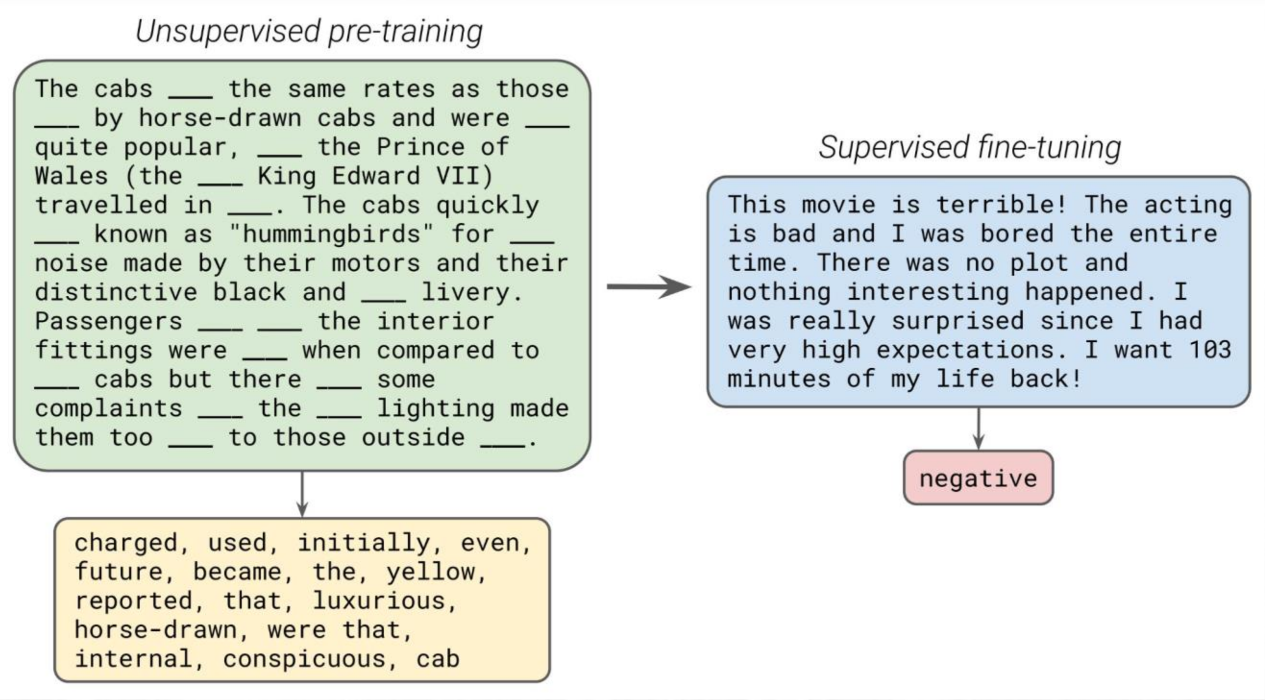


ENCODER-DECODERS for NLP

EncDec Architectures for NLP



Traditional use of LMs

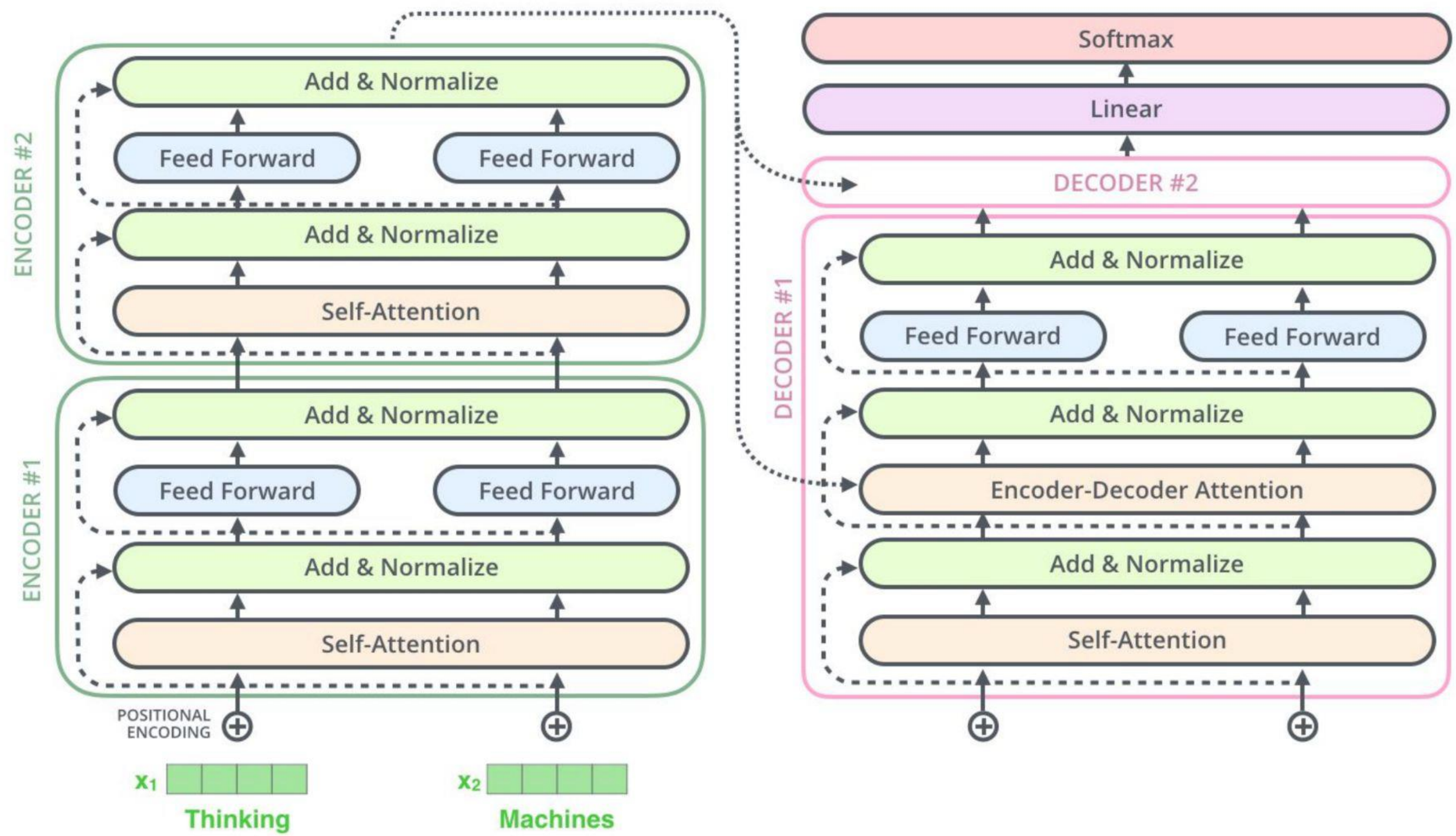


NLP Tasks: Input and Output

[Task-specific prefix]: [Input text]

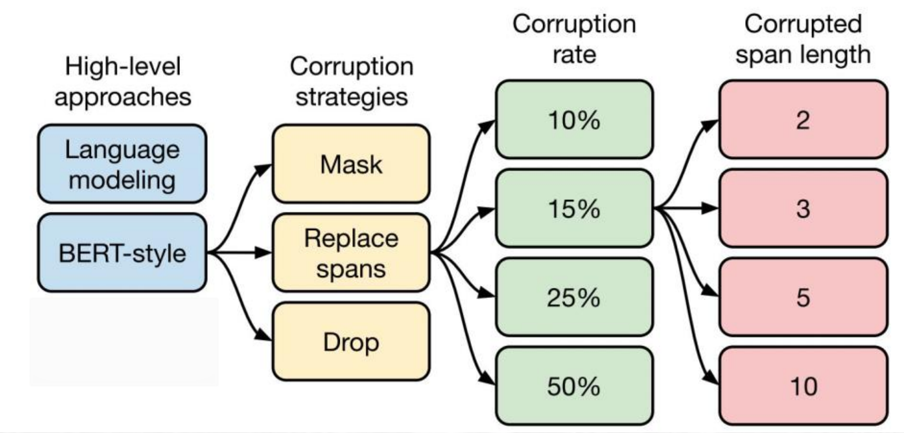
- CoLA (GLUE; Classification):
  - **Input**: sentence, **output**: labels "acceptable" or "not acceptable"
  - "cola sentence: The course is jumping well." -> "not acceptable"
  - "cola sentence: The course is jumping well." -> "hamburger" (Fail!)
- STS-B (GLUE; Regression):
  - **Input**: pair of sentences, **output**: similarity score [1,5]
  - "stsb sentence1: The rhino grazed. sentence2: A rhino is grazing." -> "3.8"
- EnDe (Translation):
  - "translate English to German: That is good" -> "Das ist gut"
- CNNDM (Summarization):
  - "summarize: state authorities dispatched…" -> "six people hospitalized after storm"

EncDec: the T5 model

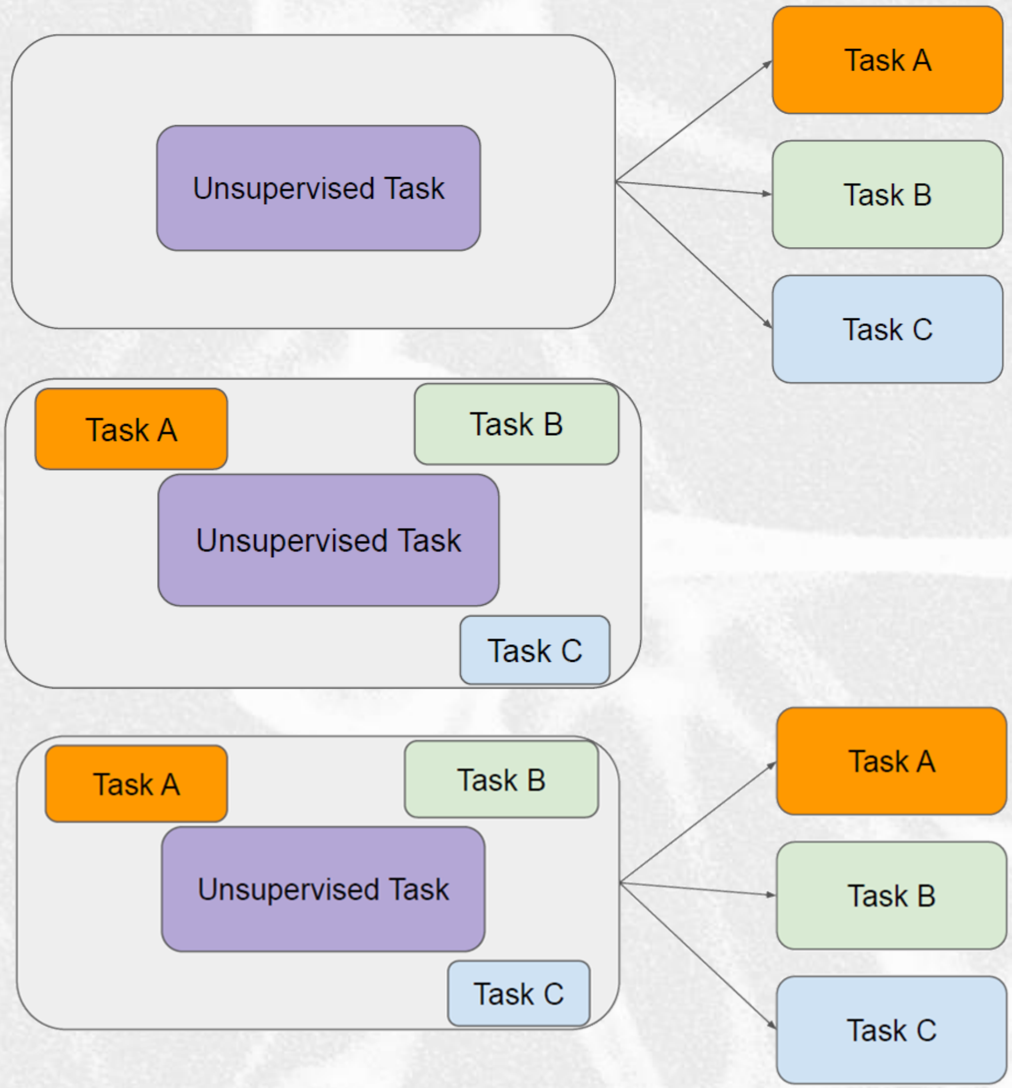


Pretraining Objectives

- Prefix Language Modeling
  - INPUT: Thank you for inviting
  - TARGETS: me to your party last week.
- BERT-style:
  - INPUT: Thank you <M> <M> me to your party apple week
  - TARGETS: Thank you for inviting e to your party last week.
- Strategies, Rates and Corrupted Span lengths suggests variants



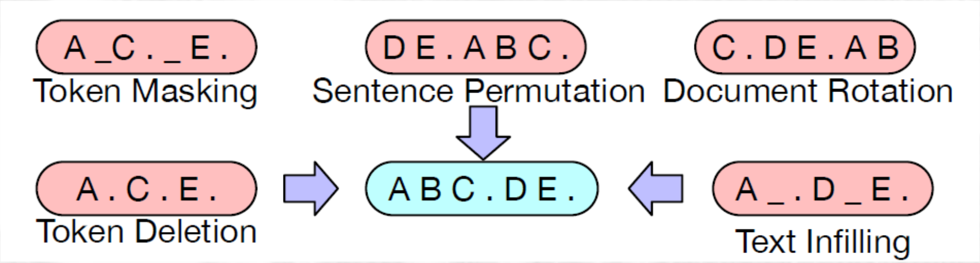
Multitask pretraining



Training strategy	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Unsupervised pre-training + fine-tuning	**83.28**	**19.24**	**80.88**	**71.36**	**26.98**	39.82	27.65
Multi-task training	81.42	**19.24**	79.78	67.30	25.21	36.30	27.76
Multi-task pre-training + fine-tuning	**83.11**	**19.12**	**80.26**	**71.03**	**27.08**	39.80	**28.07**

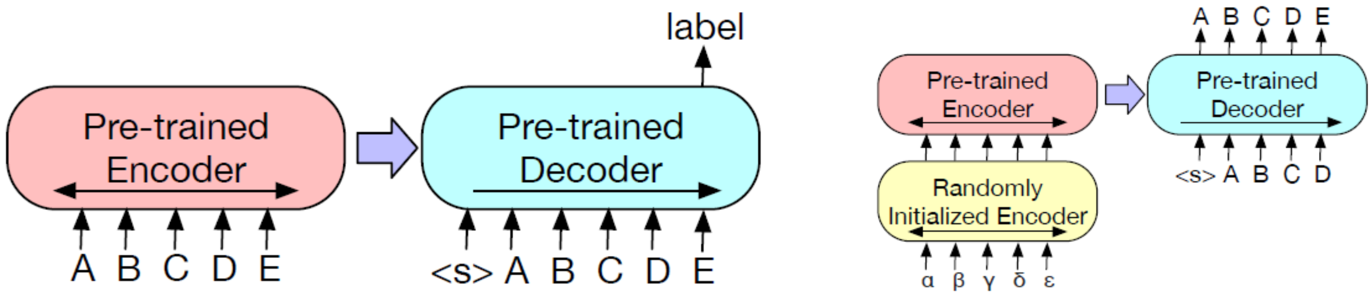
BART (Lewis et al., 2019) - Facebook

- Enconding decoding architecture based on Pretraining and fine tuned towards different tasks such as: RTE, SA, …
- Two stages of PRETRAINING
  - Text is first corrupted with an arbitrary noising function,
  - A sequence-to-sequence model is learned to reconstruct the original text.



- FINE TUNING:
  - **MNLI** (Williams et al., 2017), a bitext classification task to predict whether one sentence **entails** another. The fine-tuned model concatenates the two sentences with appended an EOS token, and passes them to both the BART encoder and decoder. In contrast to BERT, the representation of the EOS token is used to classify the sentences relations.
  - ELI5 (Fan et al., 2019), a long-form abstractive question answering dataset. Models generate answers conditioned on the concatenation of a question and supporting documents.

Applying BART



(a) To use BART for classification problems, the same input is fed into the encoder and decoder, and the representation from the final output is used.

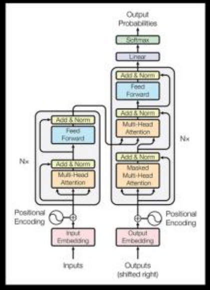
(b) For machine translation, we learn a small additional encoder that replaces the word embeddings in BART. The new encoder can use a disjoint vocabulary.

Figure 3: Fine tuning BART for classification and translation.

GrUT: The Overall Flow

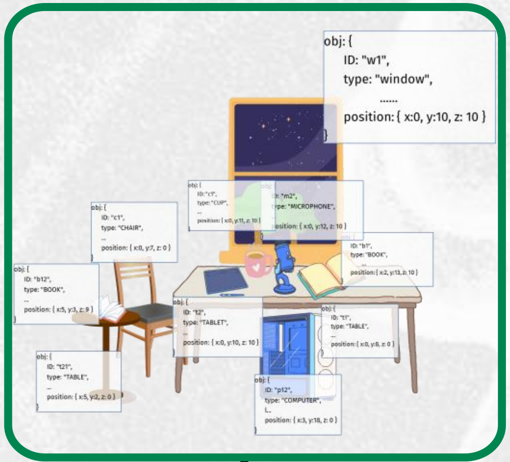
Output:

`TAKING(Theme(b1))`



GrUT-IT

Command: “*Prendi il volume sul tavolo vicino la finestra*"



Entities Retrieval

Linguistic Extraction

Input: *Command + MD*

MD: *b1, conosciuto anche come libro o volume, è un’istanza della classe BOOK, t1, conosciuto anche come tavolo o scrivania, è un’istanza della classe TABLE # b1 è vicino t1*

Hromei et al, 2022, "Embedding Contextual Information in Seq2seq Models for Grounded Semantic Role Labeling"

Experimental Evaluation

FP = Frame Prediction
AIC = Argument Identification and Classification
EM = Exact Match
HM = Head Match

Model	Learning Rate	FP	AIC-Exact Match	AIC-Head Match
LU4R	-	*95.32%*	*77.67%*	*86.35%*
GrUT-IT	$5 \cdot 10^{-5}$	**96.86%**	**82.30%**	**85.19%**

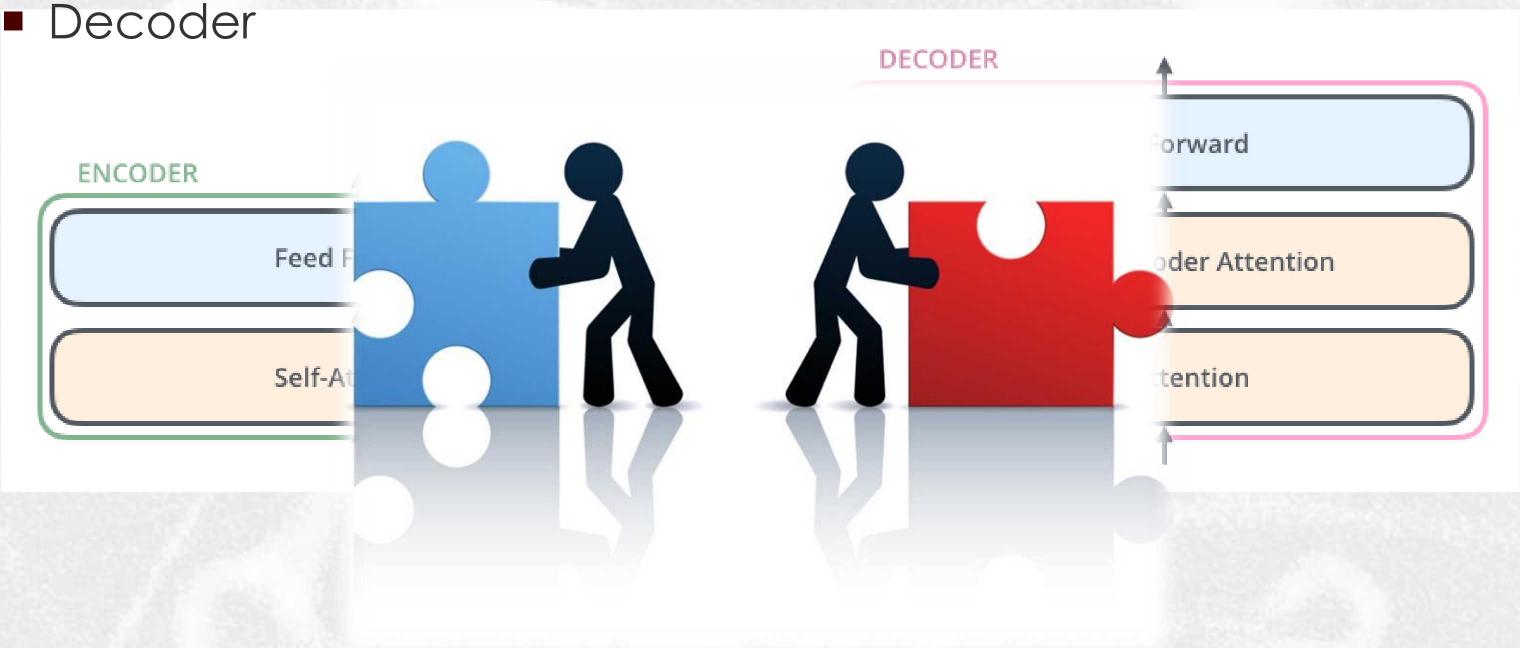
LU4R: `TAKING(Theme("libro"))`
GrUT-IT: `TAKING(Theme(b1))`

Results here are reported as F1 values on 10-fold cross-validation schema with 80/10/10 data split.
Performance for LU4R is reported in *italic* as it is not entirely comparable with.

The Transformer was only the beginning

A transformer is made of two components

- Encoder
- Decoder



GPT-2: decoder only architectures (Radford et al., 2019)

- “We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText”
- GPT-2 is a large transformer-based language model with 1.5 billion parameters, trained on a dataset of 8 million web pages.
- GPT-2 is trained with a simple objective: predict the next word, given all of the previous words within some text.
- The diversity of the dataset causes this simple goal to contain naturally occurring demonstrations of many tasks across diverse domains.
- GPT-2 is a direct scale-up of GPT, with more than 10X the parameters and trained on more than 10X the amount of data

GPT-2: sources of insipiration

- Multitask QA Networks (MQAN) (McCann et al, 2018)

Examples

Question	Context	Answer
What is a major importance of Southern California in relation to California and the US?	...Southern California is a major economic center for the state of California and the US....	major economic center
What is the translation from English to German?	Most of the planet is ocean water.	Der Großteil der Erde ist Meerwasser
What is the summary?	Harry Potter star Daniel Radcliffe gains access to a reported £320 million fortune...	Harry Potter star Daniel Radcliffe gets £320M fortune...
Hypothesis: Product and geography are what make cream skimming work. Entailment, neutral, or contradiction?	Premise: Conceptually cream skimming has two basic dimensions – product and geography.	Entailment
Is this sentence positive or negative?	A stirring, funny and finally transporting re-imagining of Beauty and the Beast and 1930s horror film.	positive

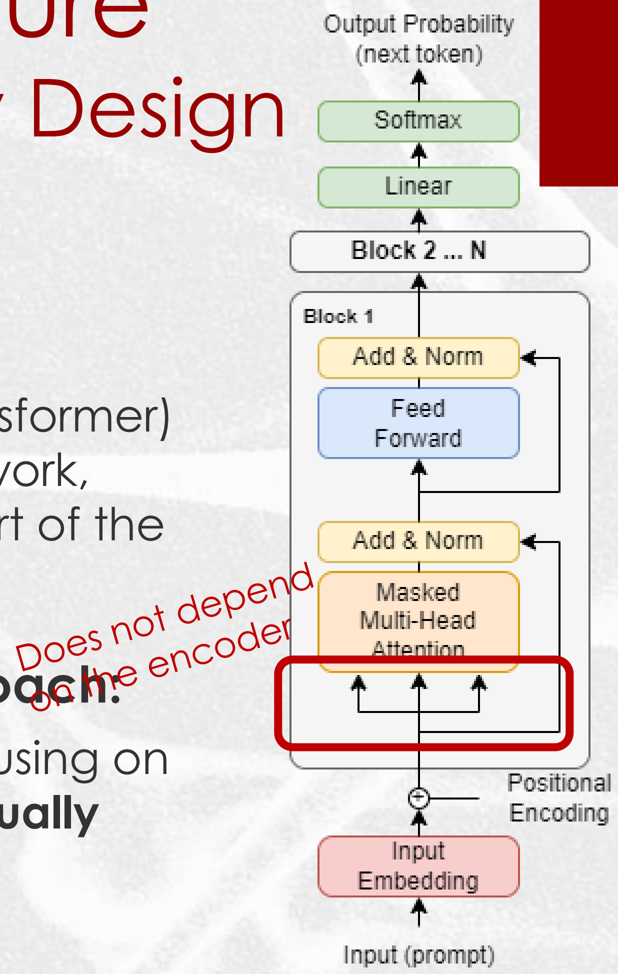
Question	Context	Answer
What has something experienced?	Areas of the Baltic that have experienced eutrophication.	eutrophication
Who is the illustrator of Cycle of the Werewolf?	Cycle of the Werewolf is a short novel by Stephen King, featuring illustrations by comic book artist Bernie Wrightson.	Bernie Wrightson
What is the change in dialogue state?	Are there any Eritrean restaurants in town?	food: Eritrean
What is the translation from English to SQL?	The table has column names... Tell me what the notes are for South Australia	SELECT notes from table WHERE 'Current Slogan' = 'South Australia'
Who had given help? Susan or Joan?	Joan made sure to thank Susan for all the help she had given.	Susan

Figure 1: Overview of the decaNLP dataset with one example from each decaNLP task in the order presented in Section 2. They show how the datasets were pre-processed to become question answering problems. Answer words in red are generated by pointing to the context, in green from the question, and in blue if they are generated from a classifier over the output vocabulary.

- Our speculation is that a language model with sufficient capacity will begin to learn to infer and perform the tasks demonstrated in natural language sequences in order to better predict them, regardless of their method of procurement. If a language model is able to do this it will be, in effect, performing unsupervised multitask learning.

The GPT Architecture and Its Decoder-Only Design (Radford et al., 2018)

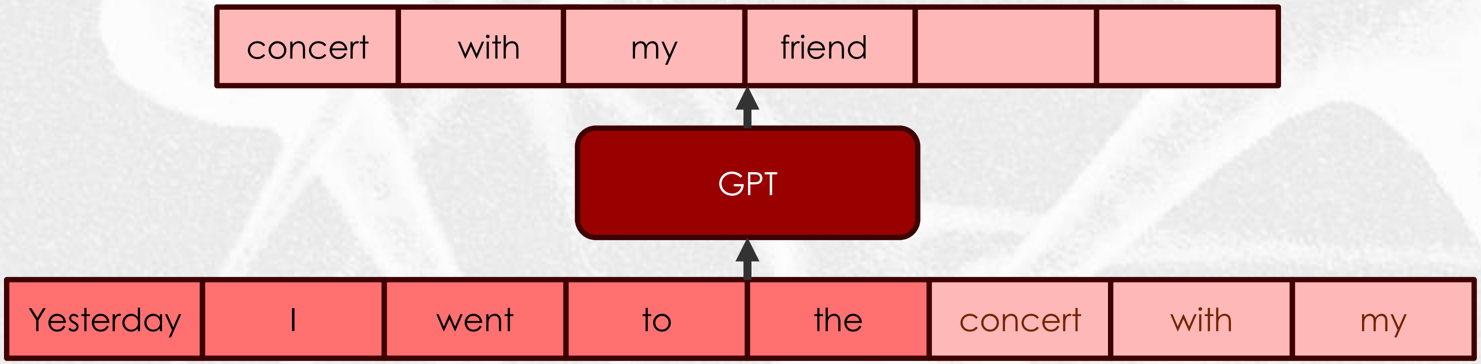
- **Decoder-Focused Architecture:**
  - GPT (Generative Pre-trained Transformer) is built on a decoder-only framework, exclusively using the decoder part of the original Transformer model.
- **Purpose of Decoder-Only Approach:**
  - to **generate meaningful text**, focusing on producing **coherent and contextually relevant** output sequences.



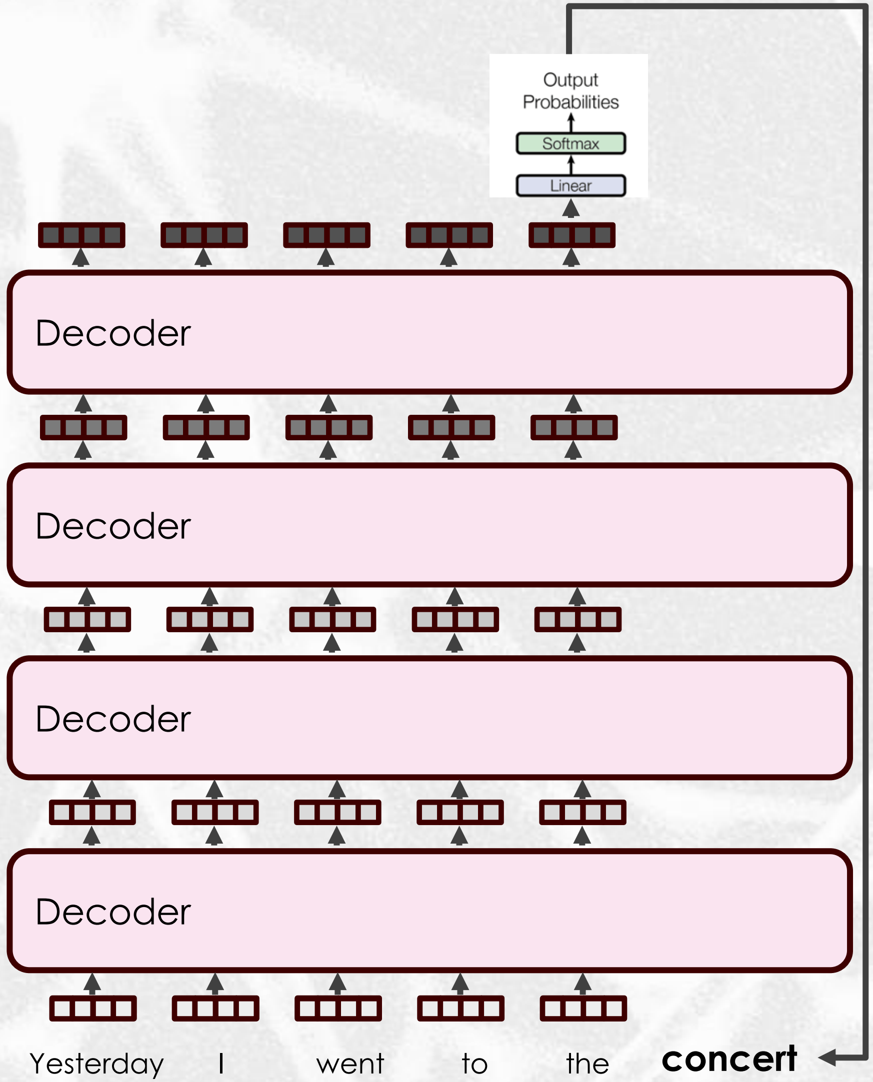
Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by Generative Pre-Training. Technical report, OpenAI.

The task:
Next Token Prediction

GPT is trained to **predict the next token in a sequence**, learning to generate text based on the preceding context.



the «Pure» Decoder in Action



- It works similarly as in the Transformer
  - But query, value and key only depends on the input sequence
- Auto-regressive
  - Masked attention is crucial

GPT-2: architecture

- Modifications:
  - Local attention: Sequence tokens are divided into blocks of similar length and attention is performed in each block independently. In our experiments, we choose to have blocks of 256 tokens.
  - Memory-compressed attention: After projecting the tokens into the query, key, and value embeddings, we reduce the number of keys and values by using a strided convolution. The number of queries remains unchanged.
- "They allow us in practice to process sequences 3x in length over the T-D model (Vaswani et al., 2017)."

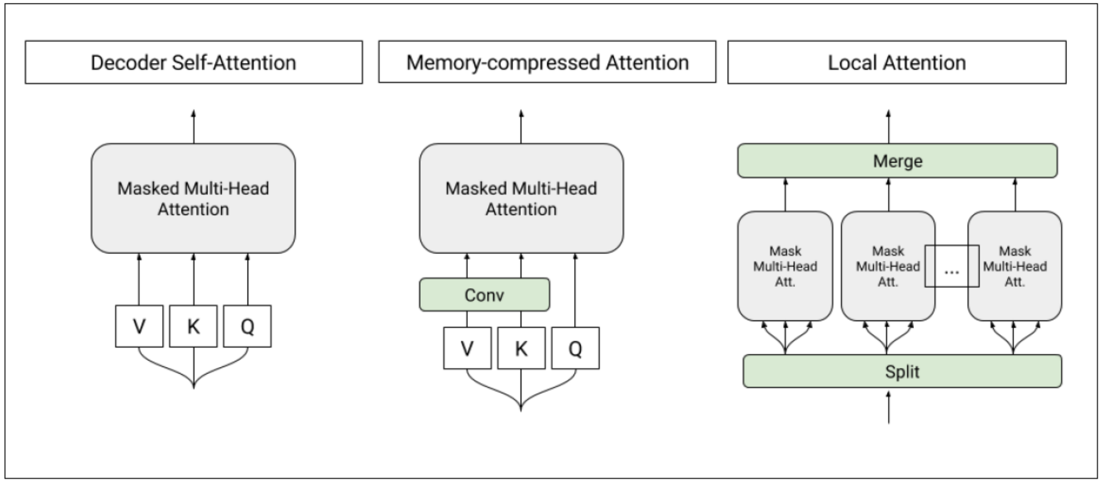


Figure 1: The architecture of the self-attention layers used in the T-DMCA model. Every attention layer takes a sequence of tokens as input and produces a sequence of similar length as the output. **Left:** Original self-attention as used in the transformer-decoder. **Middle:** Memory-compressed attention which reduce the number of keys/values. **Right:** Local attention which splits the sequence into individual smaller sub-sequences. The sub-sequences are then merged together to get the final output sequence.

GPT-2: architecture (2)

- From (Radford et al., 2017, GPT paper)

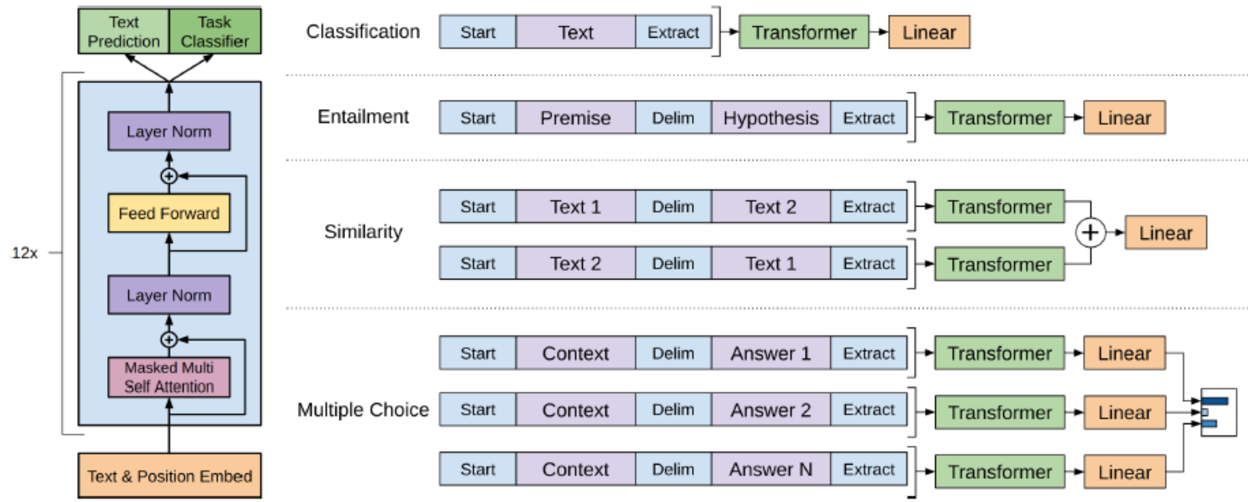


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

GPT Demonstrations

”I’m not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I’m not a fool].**
In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: ”**Mentez mentez, il en restera toujours quelque chose**,” which translates as, ”**Lie lie and something will always remain.**”
“I hate the word ‘**perfume**,”’ Burr says. ‘It’s somewhat better in French: ‘**parfum**.’
If listened carefully at 29:55, a conversation can be heard between two guys in French: “**-Comment on fait pour aller de l’autre coté? -Quel autre coté?**”, which means “**- How do you get to the other side? - What side?**”.
If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?
“**Brevet Sans Garantie Du Gouvernement**”, translated to English: “**Patented without government warranty**”.

Table 1. Examples of naturally occurring demonstrations of English to French and French to English translation found throughout the WebText training set.

GPT-2: results

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	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

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).

- The LAMBADA dataset (Paperno et al., 2016)
  - It tests the ability of systems to model long-range dependencies in text.
  - The task is to predict the final word of sentences which require at least 50 tokens of context for a human to successfully predict.

GPT-2: results on Lambada

- The LAMBADA dataset (Paperno et al., 2016)
  - It tests the ability of systems to model long-range dependencies in text.
  - The task is to predict the final word of sentences which require at least 50 tokens of context for a human to successfully predict.

(1) *Context:* "Yes, I thought I was going to lose the baby." "I was scared too," he stated, sincerity flooding his eyes. "You were ?" "Yes, of course. Why do you even ask?" "This baby wasn't exactly planned for."
Target sentence: "Do you honestly think that I would want you to have a _____ ?"
Target word: miscarriage

(2) *Context:* "Why?" "I would have thought you'd find him rather dry," she said. "I don't know about that," said Gabriel. "He was a great craftsman," said Heather. "That he was," said Flannery.
Target sentence: "And Polish, to boot," said _____.
Target word: Gabriel

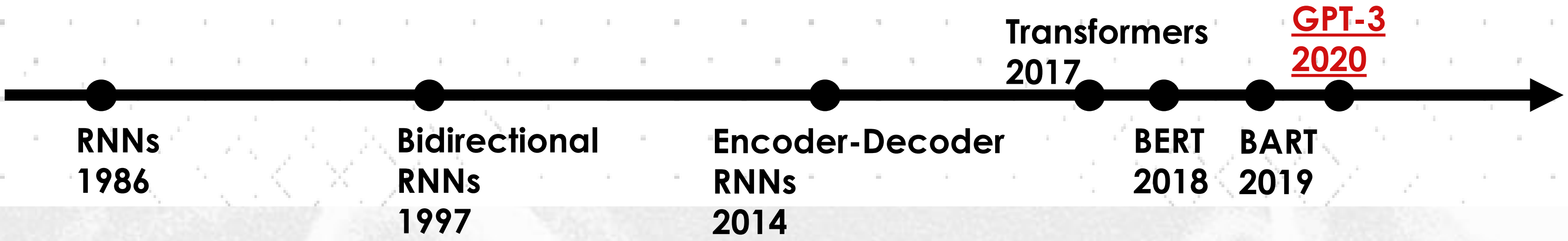
(3) *Context:* Preston had been the last person to wear those chains, and I knew what I'd see and feel if they were slipped onto my skin-the Reaper's unending hatred of me. I'd felt enough of that emotion already in the amphitheater. I didn't want to feel anymore. "Don't put those on me," I whispered. "Please."
Target sentence: Sergei looked at me, surprised by my low, raspy please, but he put down the _____.
Target word: chains

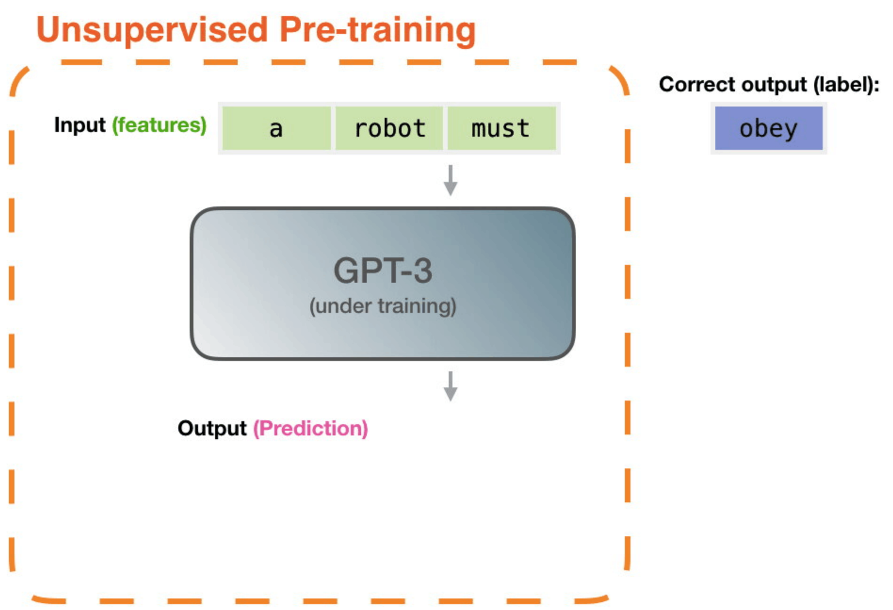
(4) *Context:* They tuned, discussed for a moment, then struck up a lively jig. Everyone joined in, turning the courtyard into an even more chaotic scene, people now dancing in circles, swinging and spinning in circles, everyone making up their own dance steps. I felt my feet tapping, my body wanting to move.
Target sentence: Aside from writing, I 've always loved _____.
Target word: dancing

- GPT-2 improves the state of the art from 99.8 (Grave et al., 2016) to 8.6 perplexity and increases the accuracy of LMs on this test from 19% (Dehghani et al., 2018) to 52.66%. Adding a stop-word filter as an approximation to this further increases accuracy to 63.24%.

- Investigating GPT-2's errors showed most predictions are valid continuations of the sentence, but are not valid final words

Machine learning paradigms underlying ChatGPT





GPT3: novelty

- «Language Models are Few-Shot Learners" (Brown et al., 2020)

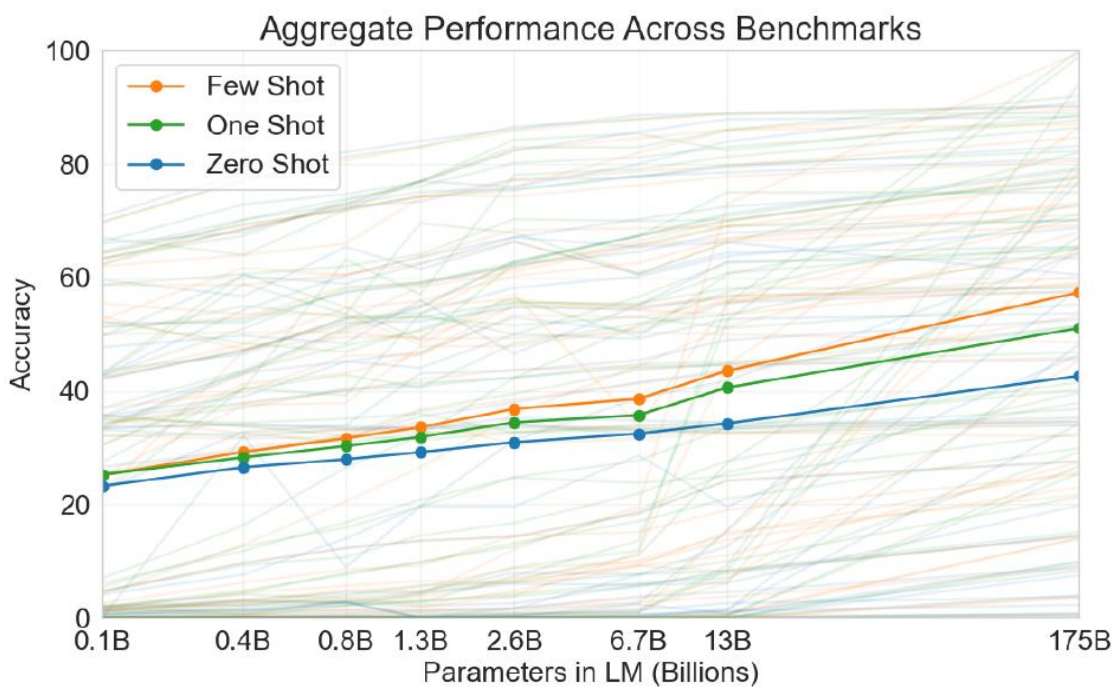


Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis on SuperGLUE, a standard NLP benchmark suite.

GPT-3

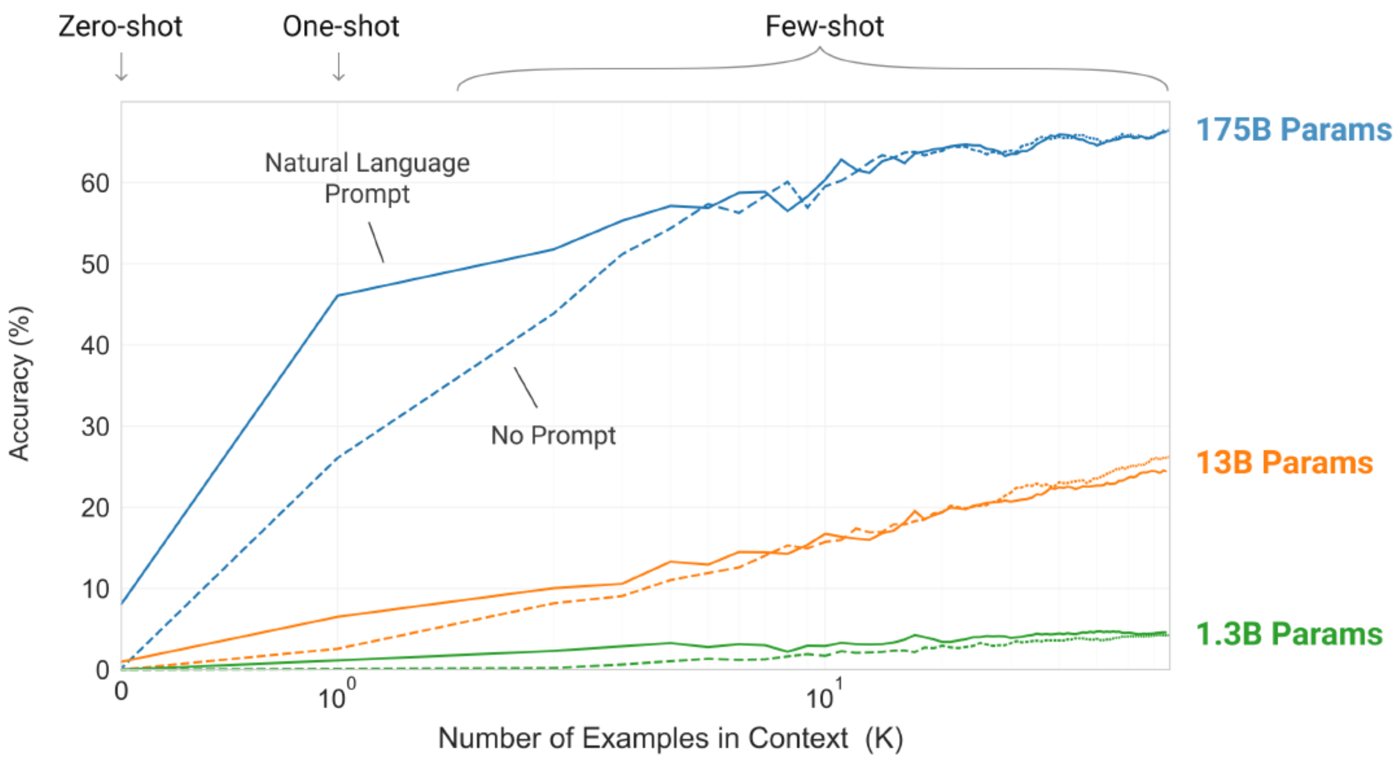
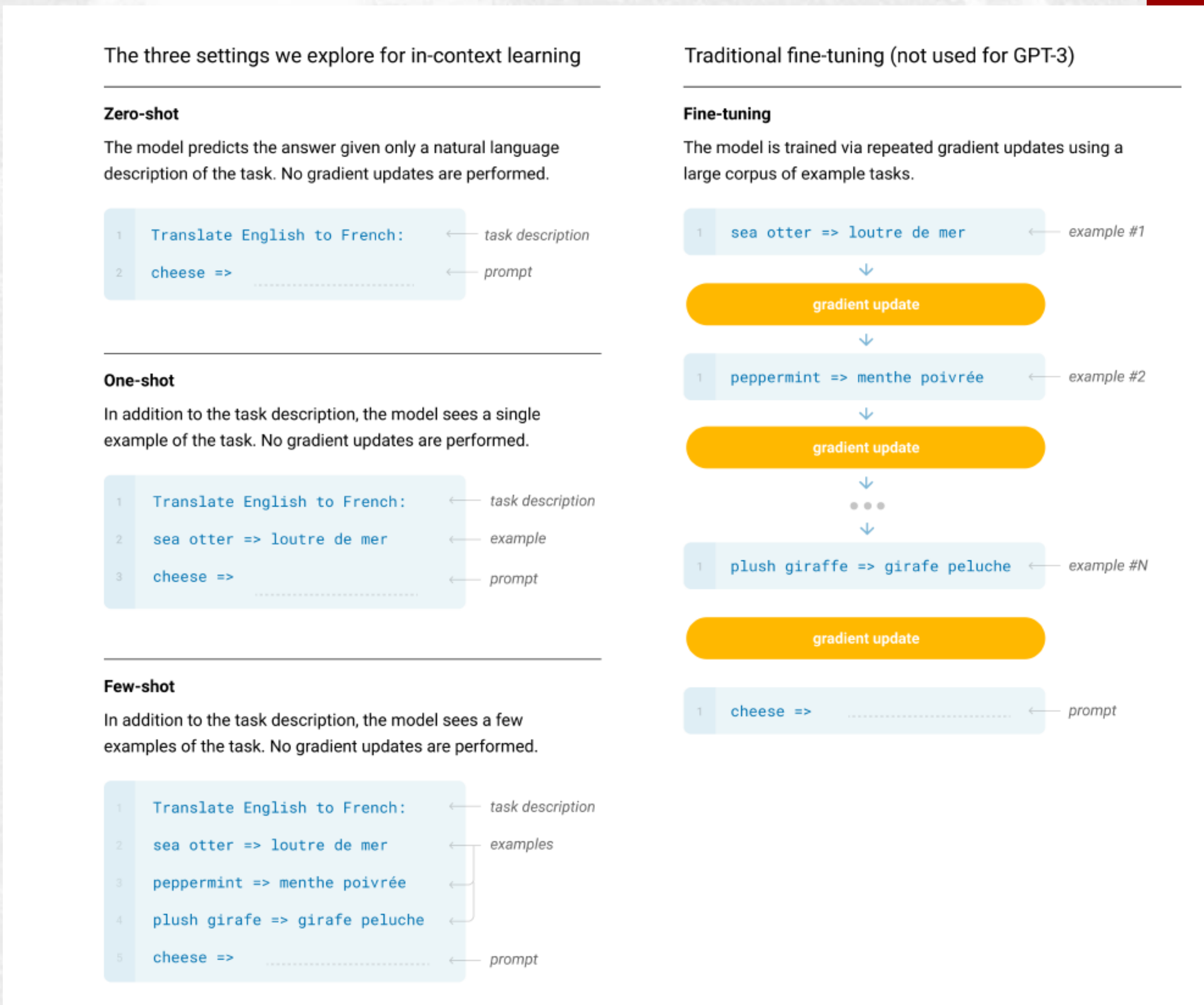


Figure 1.2: Larger models make increasingly efficient use of in-context information. We show in-context learning performance on a simple task requiring the model to remove random symbols from a word, both with and without a natural language task description (see Sec. 3.9.2). The steeper "in-context learning curves" for large models demonstrate improved ability to learn a task from contextual information. We see qualitatively similar behavior across a wide range of tasks.


The three settings we explore for in-context learning
Zero-shot
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.
Translate English to French:
task description
cheese =>
prompt
One-shot
In addition to the task description, the model sees a single example of the task. No gradient updates are performed.
Translate English to French:
task description
sea otter => loutre de mer
example
cheese =>
prompt
Few-shot
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.
Translate English to French:
task description
sea otter => loutre de mer
examples
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>
prompt
Traditional fine-tuning (not used for GPT-3)
Fine-tuning
The model is trained via repeated gradient updates using a large corpus of example tasks.
sea otter => loutre de mer
example #1
gradient update
peppermint => menthe poivrée
example #2
gradient update
plush giraffe => girafe peluche
example #N
gradient update
cheese =>
prompt

GPT-3: size

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

- Here n_{params} is the total number of trainable parameters, n_{layers} is the total number of layers, d_{model} is the number of units in each bottleneck layer (we always have the feedforward layer four times the size of the bottleneck layer, $d_{ff}=4xd_{model}$), and d_{head} is the dimension of each attention head.
- All models use a context window of n_{ctx} = 2048 tokens

But does GPT 'only' know how to predict the next word in a sentence?

- If we are smart enough, we can use the generation capability of GPT to solve a task, but…
  - We can ask GPT to do something, e.g. write an article:

Title: United Methodists Agree to Historic Split

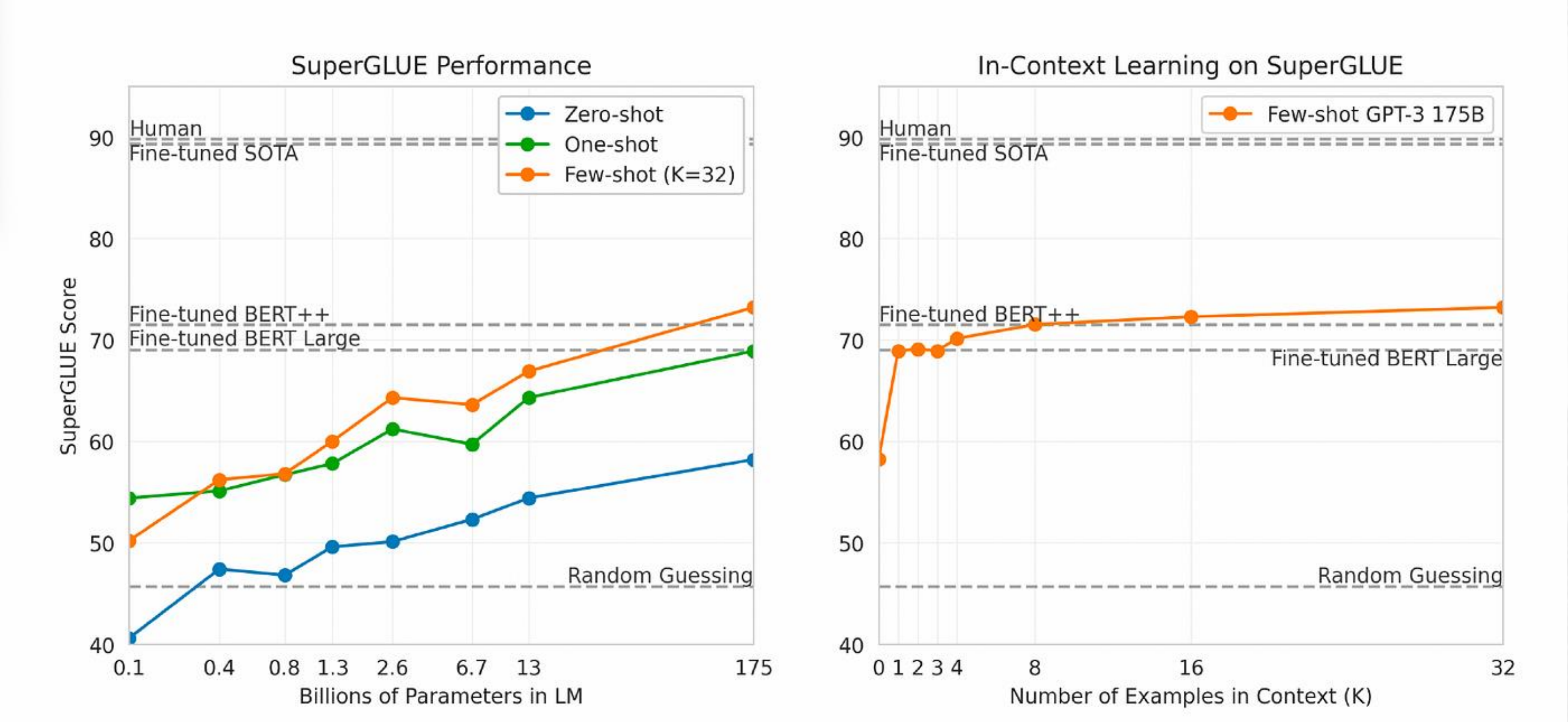
Subtitle: Those who oppose gay marriage will form their own denomination

Article: After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination. The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split "comes at a critical time for the church, which has been losing members for years," which has been "pushed toward the brink of a schism over the role of LGBTQ people in the church." Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with

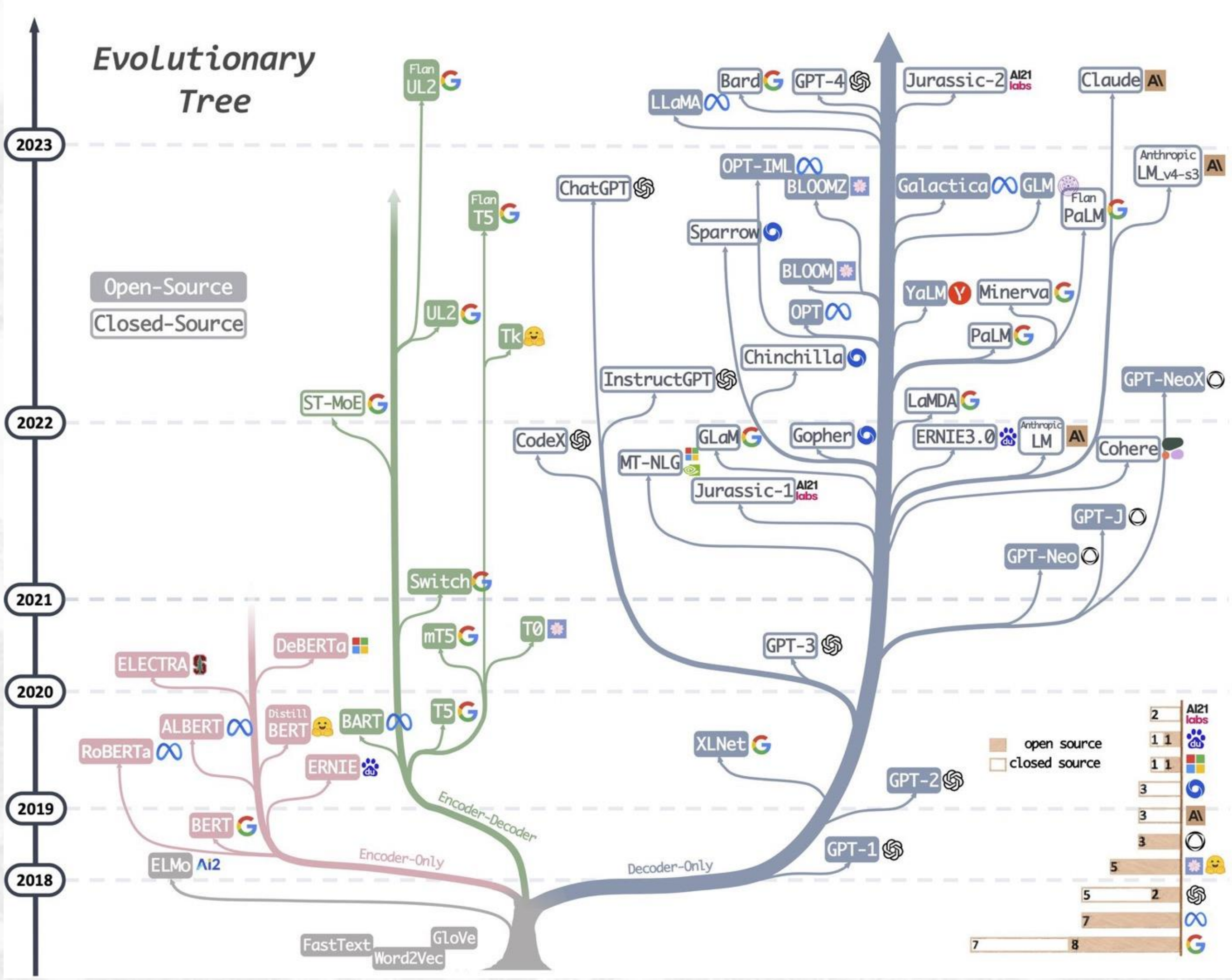
The «powers» of GPT3

Diverse Task Performance Without Fine-Tuning

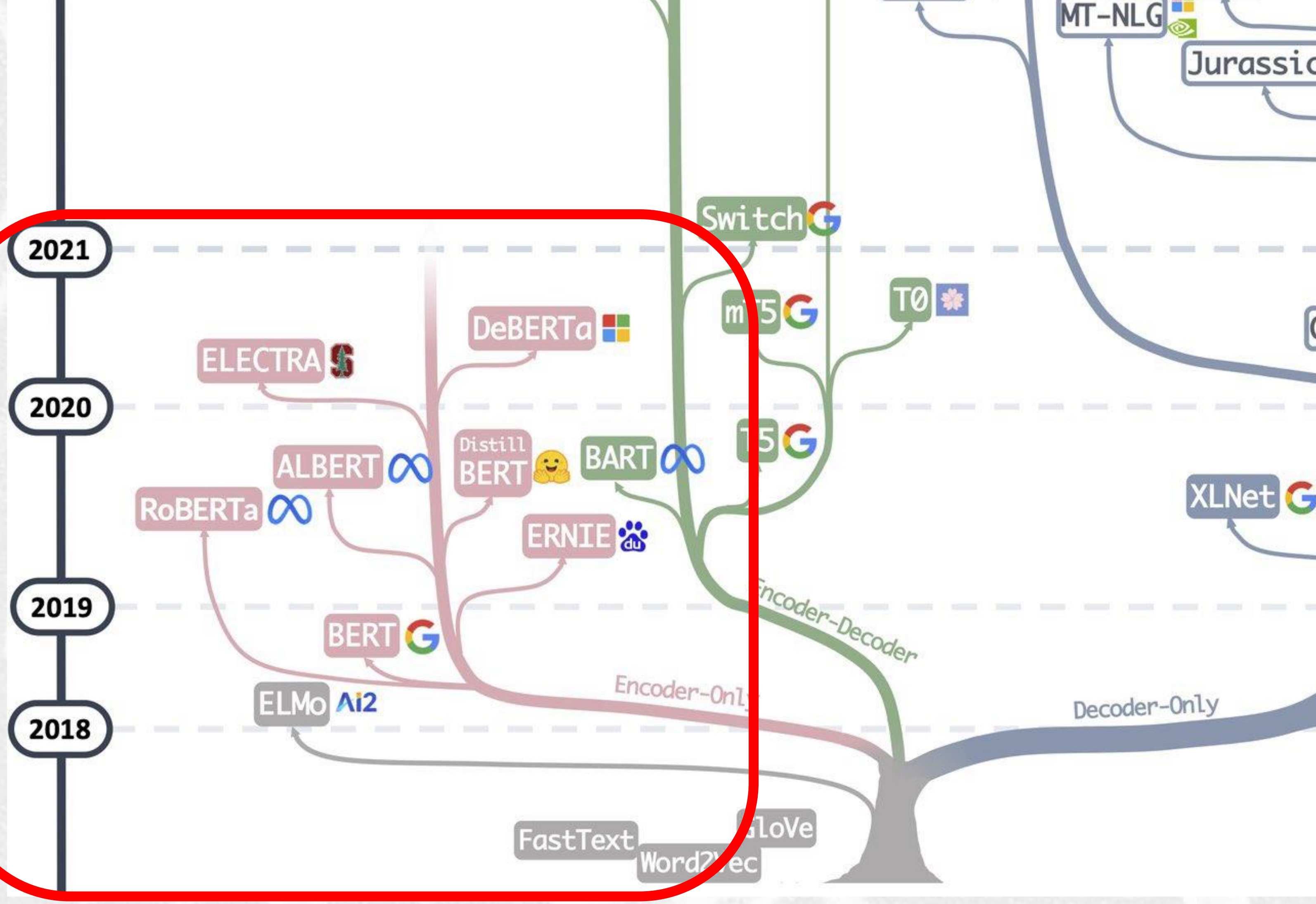
- Exhibits strong performance across various NLP tasks through text interactions alone, including translation, question-answering, and reasoning tasks.



The rest is a family tree

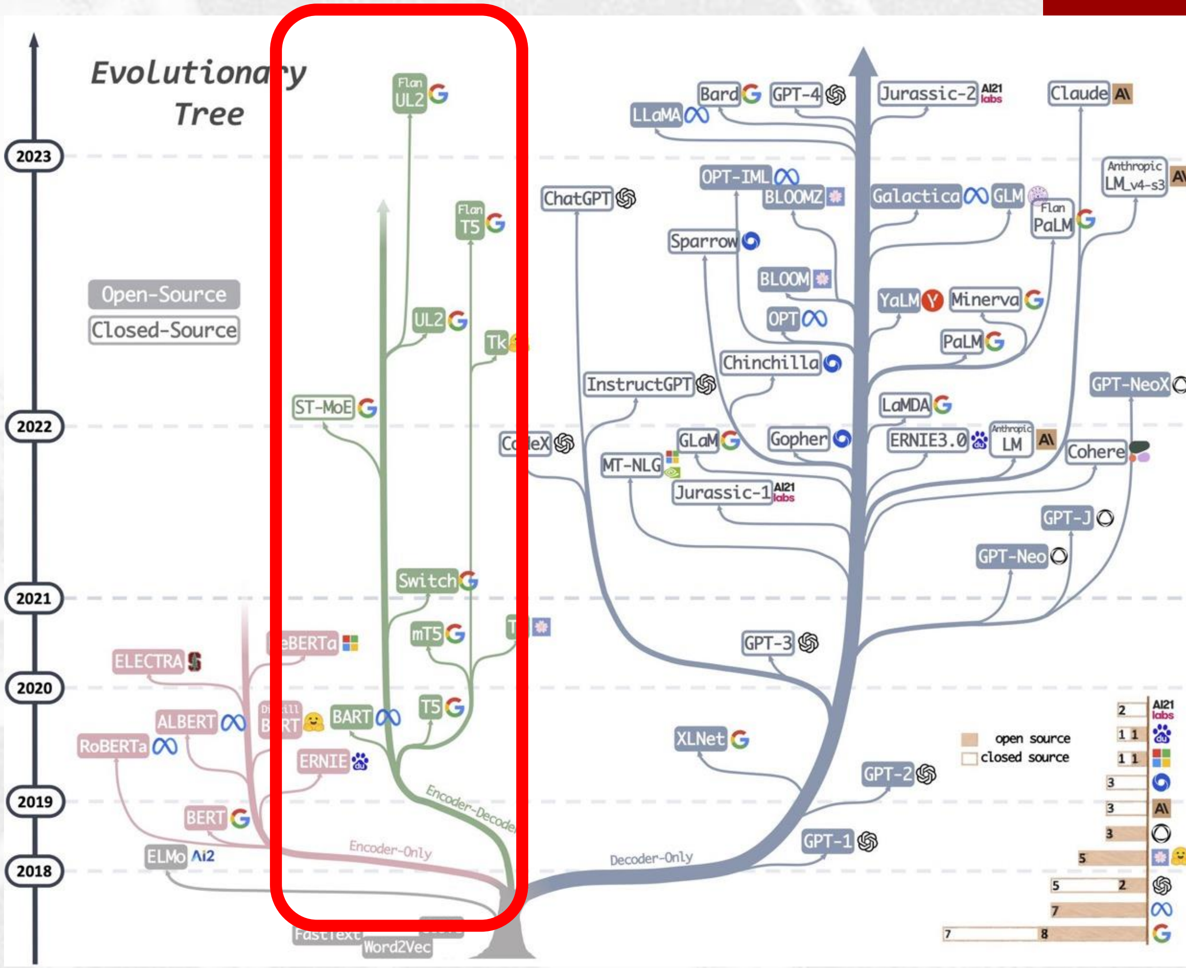


The «Encoder Only» family



Encoder-based architectures experienced **rapid initial growth** and enormous success until 2021, **after which interest shifted**.

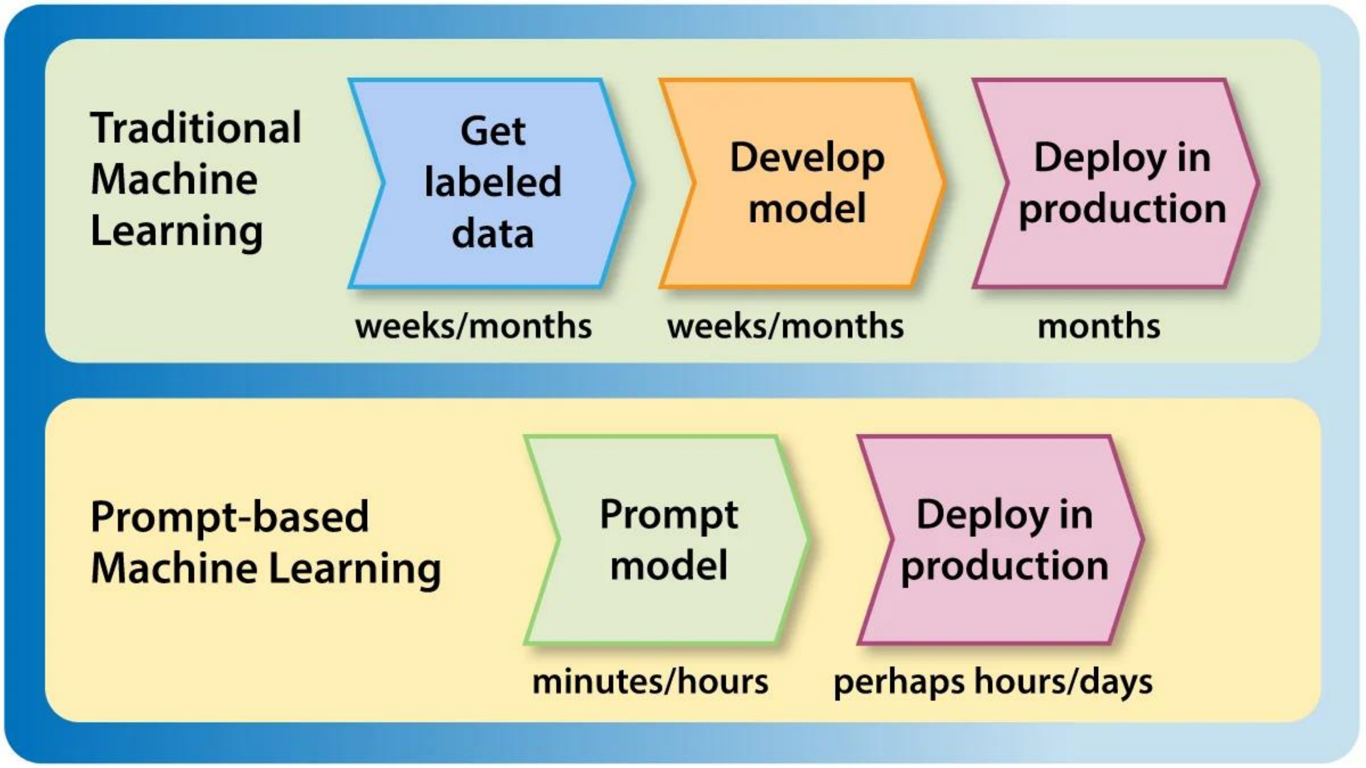
The «Encoder/Decoder» family



Encoder-Decoder based architectures experienced **a more limited success but largerly used**, especially tasks requiring generation

More on Prompting

Trends …

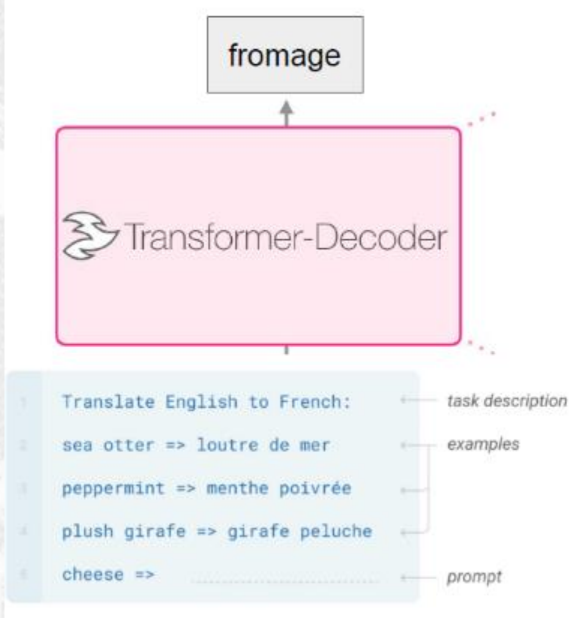

Traditional Machine Learning
Get labeled data
weeks/months
Develop model
weeks/months
Deploy in production
months
Prompt-based Machine Learning
Prompt model
minutes/hours
Deploy in production
perhaps hours/days

Learning Modalities

- Fine Tuning (as BERT/BART)
- In-context learning
- Prompting

IN-context Learning

- Pretrain a large language model on a task
- Manually design a «prompt» that shows how to define a novel taks as a generation task
- There is no need to train further the model, i.e. update model weights



Brown et al. 2020

PROMPTING

- "A good prompt is one that is specific and provides enough context for the model to be able to generate a response that is relevant to the task." (GPT-3)

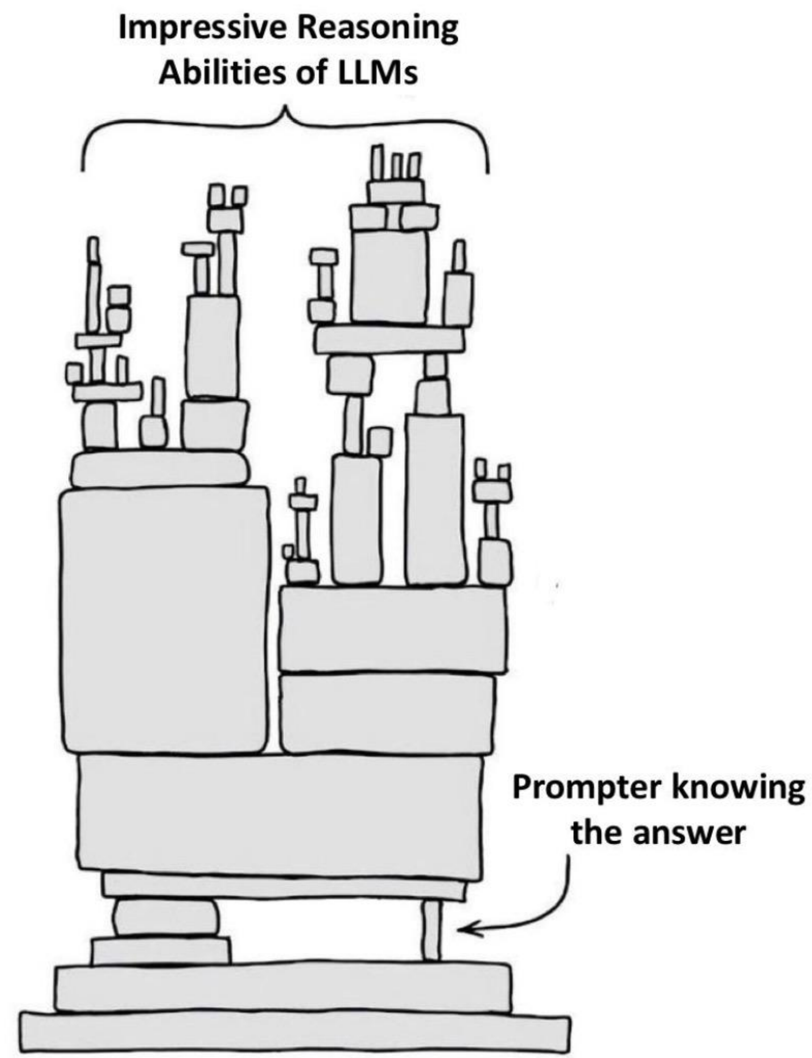
- Earliest work in prompts traces back to GPT-1/2 (Radford et al., 2018,2019)

- If LMs are given good prompts they can achieve significant zero-shot performance on NLP tasks ranging from sentiment classification to reading comprehension

Prompting LLMs

Subbarao Kambhampati (కంభంపాటి సుబ్బారావు)
@rao2z

The tldr I use: "**LLMs always hallucinate. Sometimes their hallicinations align with your reality"**. Whether or not the prompt makes them hallucinate in a way that aligns with reality depends very much on the prompter's ability to check, and thus.. x.com/rao2z/status/1...)



PROMPT based fine tuning

Fine tuning: more paremeters for the stacked classifier, more examples (even in few-shot scenarios)

Prompt-based fine tuning: need for good prompts, no further parameters to tune

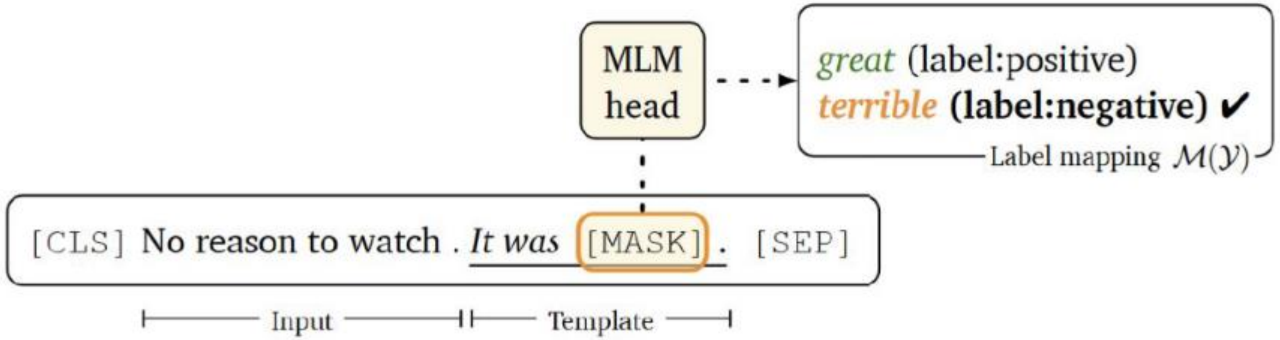
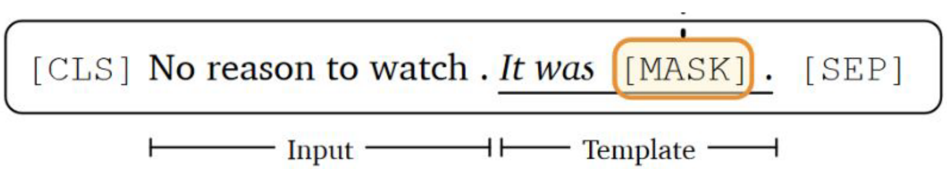


Image Source: Making Pre-trained Language Models Better Few-shot Learners, Gao, et al. 2021

Prompt-based fine tuning: the process

Input: x_1 = No reason to watch.

Step 1. Formulate the downstream task into a (Masked) LM problem using a *template*:



Step 2. Choose a *label word mapping* $\mathcal{M}$, which maps task labels to individual words.

Image Source: Making Pre-trained Language Models Better Few-shot Learners, Gao, et al. 2021

Prompt-based fine tuning: the process

Step 3. Fine-tune the LM to fill in the correct label word.

$$p(y \mid x_{\text{in}}) = p\left(\texttt{[MASK]} = \mathcal{M}(y) \mid x_{\text{prompt}}\right) = \frac{\exp\left(\mathbf{w}_{\mathcal{M}(y)} \cdot \mathbf{h}_{\texttt{[MASK]}}\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\mathbf{w}_{\mathcal{M}(y')} \cdot \mathbf{h}_{\texttt{[MASK]}}\right)},$$

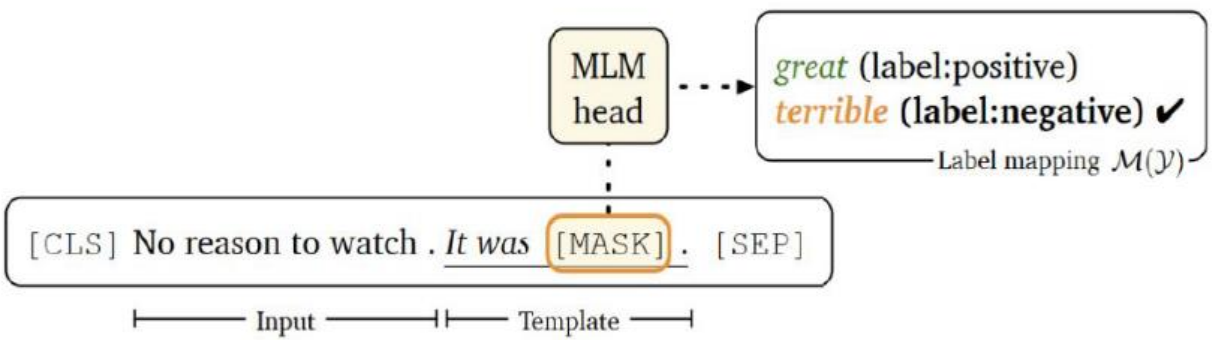


Image Source: Making Pre-trained Language Models Better Few-shot Learners, Gao, et al. 2021

Prompt based fine tuning: tasks

SST-2: sentiment analysis.

- E.g. **S1** = "The movie is ridiculous". **Label**: negative.
- Manual prompt:

Template	**Label words**
$<S_1>$ It was [MASK] .	great/terrible

SNLI: Natural Language Inference

- **S1** = "A soccer game with multiple males playing". **S2** = "Some men are playing sport". **Label**: Entailment.
- Manual prompt:

Template	**Label words**
$<S_1>$? [MASK] , $<S_2>$	Yes/Maybe/No

Prompting

- GPT-3 🤖
- PET 🐱
- LM-BFF 👯

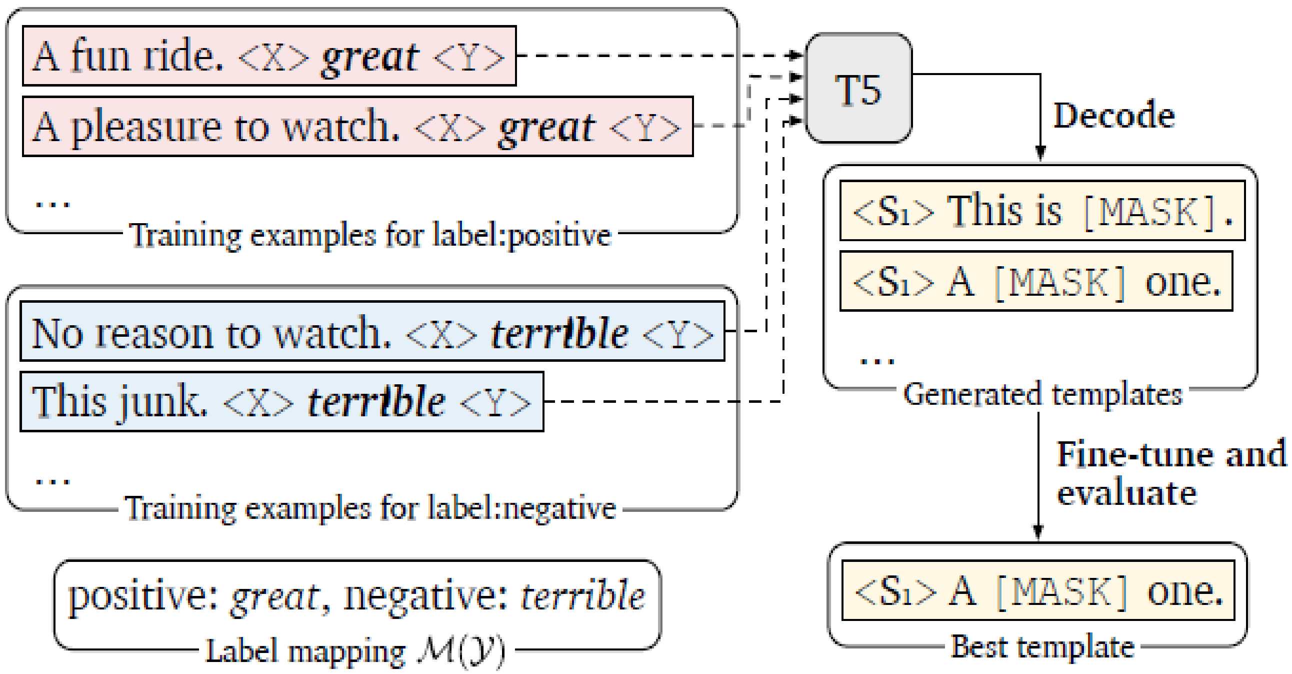


Figure 2: Our approach for template generation.

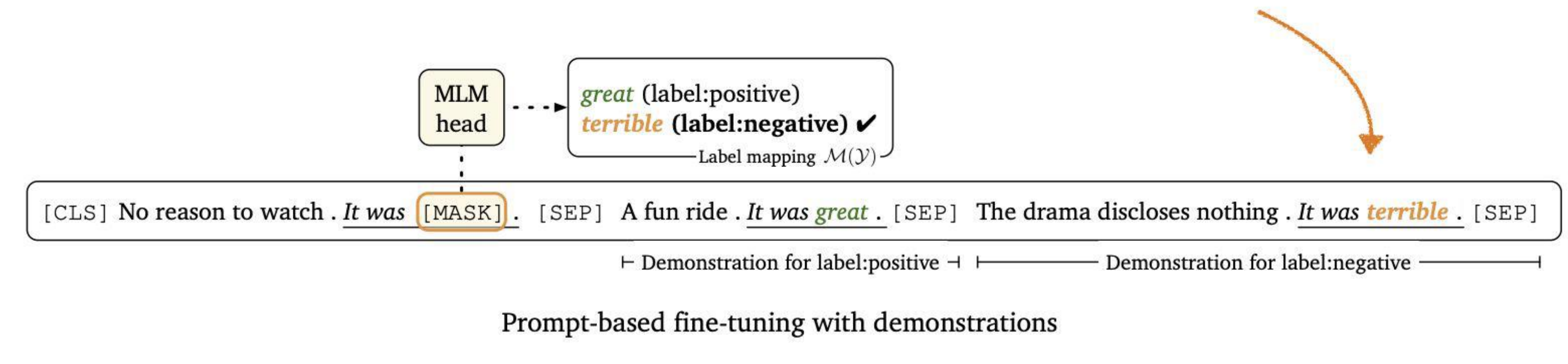
Datasets

Category	Dataset	$\|\mathcal{Y}\|$	Type	Labels (classification tasks)
single-sentence	SST-2	2	sentiment	positive, negative
	SST-5	5	sentiment	v. pos., positive, neutral, negative, v. neg.
	MR	2	sentiment	positive, negative
	CR	2	sentiment	positive, negative
	MPQA	2	opinion polarity	positive, negative
	Subj	2	subjectivity	subjective, objective
	TREC	6	question cls.	abbr., entity, description, human, loc., num.
	CoLA	2	acceptability	grammatical, not_grammatical
sentence-pair	MNLI	3	NLI	entailment, neutral, contradiction
	SNLI	3	NLI	entailment, neutral, contradiction
	QNLI	2	NLI	entailment, not_entailment
	RTE	2	NLI	entailment, not_entailment
	MRPC	2	paraphrase	equivalent, not_equivalent
	QQP	2	paraphrase	equivalent, not_equivalent
→	STS-B	$\mathcal{R}$	sent. similarity	-

Source: Making Pre-trained Language Models Better Few-shot Learners, Gao, et al. 2021

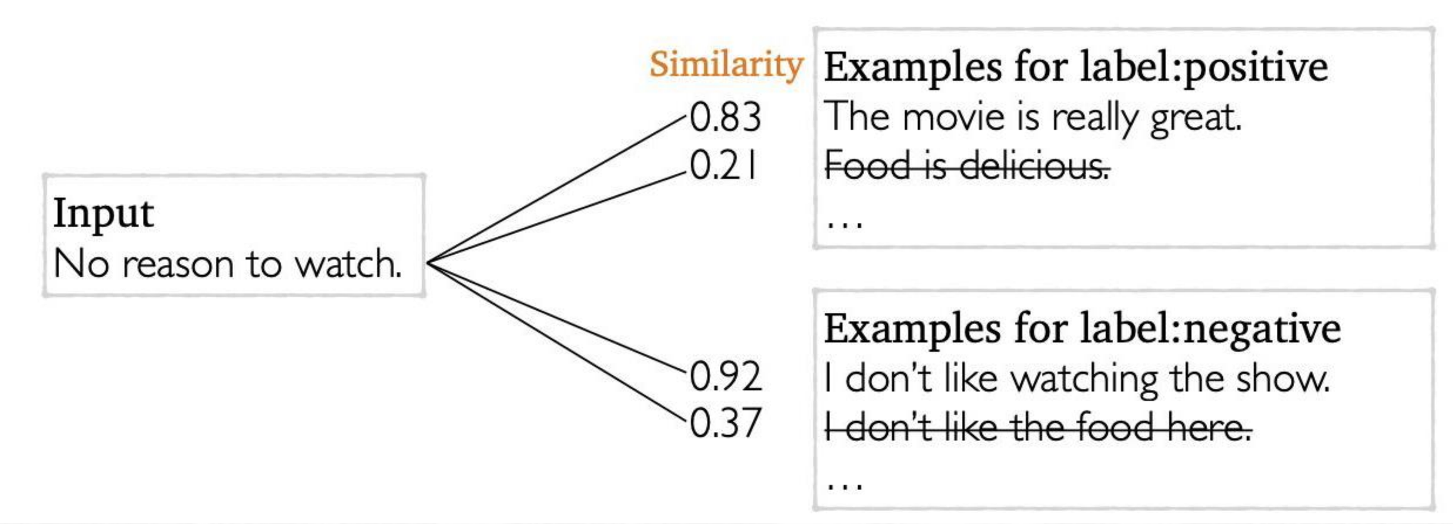
Prompt based on demonstration

- Demonstration is based on the idea that in few-shot learning you can exemplify a task by using instances from the training set that demonstrate how to solve a task

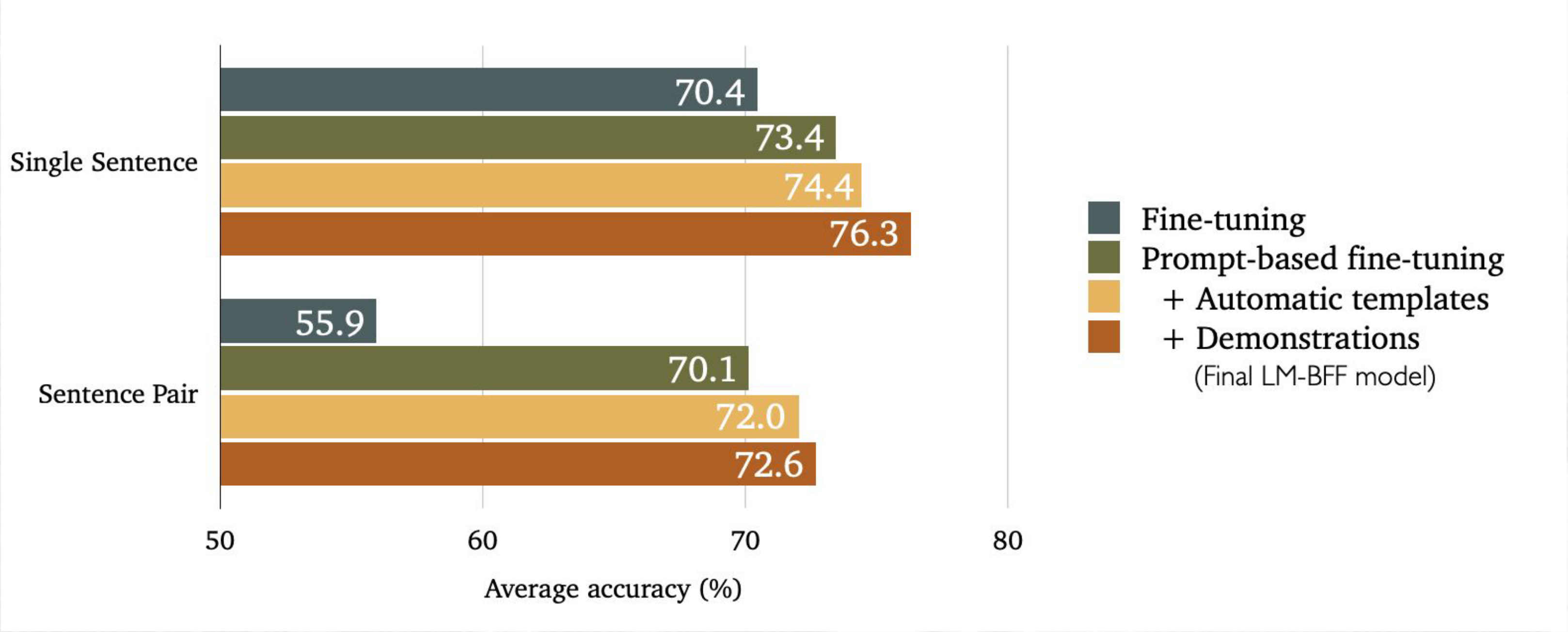


- Selective demonstration (INTUITION): Apply **demonstrations** that are **semantically close** to the input for optimal results

Examples of demonstrations



Prompting with demostrations



- From '**Making Pre-trained Language Models Better Few-shot Learners**', Gao et al, ACL 2021 paper
  - Paper
  - VIDEO

Beyond Transformer
bibliography

- (Vaswani 2017), Attention is all you need, https://arxiv.org/abs/1706.03762
- (Devlin et al 2018), BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, https://arxiv.org/abs/1810.04805
- Rocktaschel et al., "Reasoning About Entailment With Neural Attention" (ICLR 2016)
- T5: (Wolf et al, 2019) Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. ArXiv, abs/1910.03771.
- BART Encoding-Decoding: Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.https://arxiv.org/abs/1910.13461
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, "Improving Language Understanding by Generative Pre-Training", 2019
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei: Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901. https://arxiv.org/abs/2005.14165, NeurIPS 2020.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, Paul F. Christiano: Learning to summarize with human feedback. NeurIPS 2022