

# In-Context-Learning

## Experiments on Manual Template Engineering

Ann Weitz, Xiao Yang, Annabelle Runge, Lea Krumbach,  
Emanuel Nierstenhöfer & Sandy Rodrigues

June 12, 2024

# Outline

1. **WHAT IS MANUAL TEMPLATE ENGINEERING?**
2. **PROMPTING TECHNIQUES**
3. **LANGUAGE MODELS AS KNOWLEDGE BASES**
4. **SMALL LANGUAGE MODELS ARE ALSO FEW-SHOT LEARNERS**
5. **SUMMARY**

# WHAT IS MANUAL TEMPLATE ENGINEERING?

# What is Manual Template Engineering?

## ▶ **Definition:**

- ▶ Creating specific input formats (templates) for AI models.
- ▶ Guides the responses of AI.

## ▶ **Purpose:**

- ▶ Improves the accuracy and quality of AI responses.

# Advantages

- ▶ **Improved Accuracy:** Enhances the precision of responses by reducing ambiguity.
- ▶ **Consistency:** Ensures uniformity in responses across different instances.
- ▶ **Efficiency:** Reduces the need for extensive post-processing or corrections.
- ▶ **User Satisfaction:** Leads to more relevant and satisfactory interactions for end-users.

## 2. PROMPTING TECHNIQUES

# What prompting techniques do you know?

(or: How would you structure your prompts?)

# Tips for structuring your prompts

- ▶ Be specific but avoid unnecessary details
- ▶ Use Keywords
  - ▶ "Write"
  - ▶ "Classify"
  - ▶ "Summarize"
  - ▶ "Translate"
  - ▶ "Order"
- ▶ Experiment with different prompts
- ▶ Context Setting
- ▶ Separate instruction and context (e.g., “, ””””‘)
- ▶ Articulate the desired output format



# Prompting techniques

1. Zero-Shot Prompting
2. Few-Shot Prompting
3. Chain-of-Thought Prompting
4. Generate Knowledge Prompting
5. Tree of Thoughts

# Zero-Shot Prompting: Definition and Advantages

- ▶ **Definition:** AI models can perform tasks without specific training.
- ▶ **Advantages:**
  - ▶ **Versatility:** Handles various tasks without task-specific training.
  - ▶ **Efficiency:** Saves time and resources by not needing task-specific data.
  - ▶ **Adaptability:** Quickly adjusts to new tasks with minimal modifications.

# Zero-Shot Prompting: Example and Applications

- ▶ **Example:**

- ▶ **Prompt:**

- “Classify the text into neutral, negative or positive.

- Text: I think the vacation is okay.

- Sentiment: “

- ▶ **Response:** Neutral

- ▶ **Applications:**

- ▶ Language translation
  - ▶ Text summarization
  - ▶ Question answering
  - ▶ Content generation

# Few-Shot Prompting: Definition and Advantages

- ▶ **Definition:** AI models learn from a few examples to perform tasks.
- ▶ **Advantages:**
  - ▶ **Flexibility:** Adapts to various tasks with minimal examples.
  - ▶ **Scalability:** Scales efficiently with a small dataset.
  - ▶ **Accuracy:** Maintains high performance with limited data.

# Few-Shot Prompting: Example and Applications

- ▶ **Example:**

- ▶ **Prompt:**

- “This is awesome! // Negative

- This is bad! // Positive

- Wow that movie was rad! // Positive

- What a horrible show! //

- ▶ **Response:** Negative

- ▶ **Applications:**

- ▶ Content summarization

- ▶ Sentiment analysis

- ▶ Text classification

- ▶ Document categorization

# Arithmetic Tasks

## Standard Prompting

### Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The answer is 27. ❌

Wei et al. 2022

# Chain-of-Thought Prompting

**Aim:** enable complex reasoning capabilities through intermediate reasoning steps; generate a chain of thought

## **Why?**

- insight into reasoning path of LM (facilitates debugging)
- useful for math word problems, commonsense reasoning, and symbolic manipulation

# Chain-of-Thought Prompting

## Standard Prompting

### Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The answer is 27. ❌

## Chain-of-Thought Prompting

### Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is  $5 + 6 = 11$ . The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had  $23 - 20 = 3$ . They bought 6 more apples, so they have  $3 + 6 = 9$ . The answer is 9. ✓

Wei et al. 2022



# Limitations

- ▶ High scale models → high performance
- ▶ Effectiveness of LM reliant on complexity of problem
- ▶ Uncertainty whether LM is actually "reasoning"
- ▶ Costly to serve in real-world applications

# Zero-shot CoT Prompting

Add "Let's think step-by-step" to the original prompt

# Zero-shot CoT Prompting

Add "Let's think step-by-step" to the original prompt

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

*(Output) The answer is 8. X*

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

*(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are  $16 / 2 = 8$  golf balls. Half of the golf balls are blue. So there are  $8 / 2 = 4$  blue golf balls. The answer is 4. ✓*

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

*(Output) 8 X*

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

*(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓*

Kojima et al. 2022

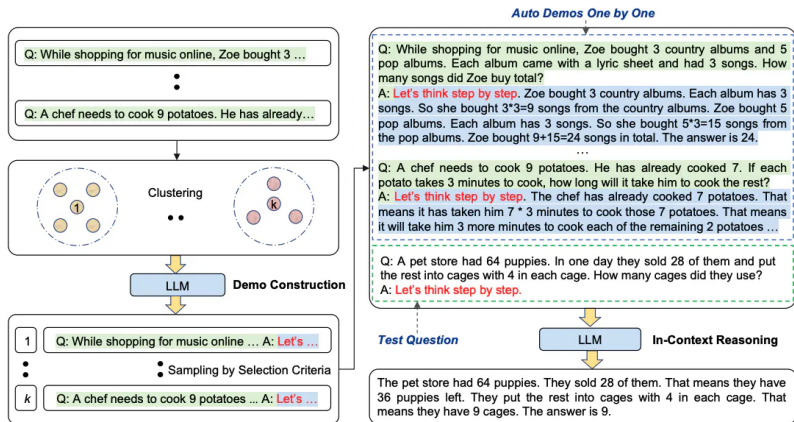
# Automatic CoT

Eliminate manual efforts by leveraging LLMs with "Let's think step by step" prompt to generate reasoning chains for demonstrations one by one

Two stages:

- (1) **Question clustering**: partition questions of a given dataset into clusters
- (2) **Demonstration sampling**: select a representative question from each cluster and generate its reasoning chain using zero-shot-CoT

# Automatic CoT



Zhang et al. 2022

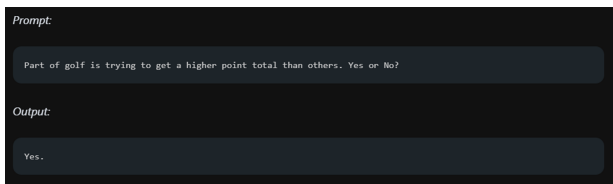
# Generate Knowledge Prompting

**Popular idea:** incorporate knowledge to help the model make more accurate ideas

# Generate Knowledge Prompting

**Popular idea:** incorporate knowledge to help the model make more accurate ideas

→ Can the model also be used to generate knowledge before making a prediction? (Liu et al. 2021)



# Generate Knowledge Prompting: Knowledge

## Prompt

Input: A rock is the same size as a pebble.

Knowledge: A pebble is a clast of rock with a particle size of 4 to 64 millimetres based on the Udden-Wentworth scale of sedimentology. Pebbles are generally considered larger than granules (2 to 4 millimetres diameter) and smaller than cobbles (64 to 256 millimetres diameter).

Input: Part of golf is trying to get a higher point total than others.

Knowledge:



# Generate Knowledge Prompting: Knowledge

## Prompt

Input: A rock is the same size as a pebble.

Knowledge: A pebble is a clast of rock with a particle size of 4 to 64 millimetres based on the Udden-Wentworth scale of sedimentology. Pebbles are generally considered larger than granules (2 to 4 millimetres diameter) and smaller than cobbles (64 to 256 millimetres diameter).

Input: Part of golf is trying to get a higher point total than others.

Knowledge:

## Knowledge 1

The objective of golf is to play a set of holes in the least number of strokes. A round of golf typically consists of 18 holes. Each hole is played once in the round on a standard golf course. Each stroke is counted as one point, and the total number of strokes is used to determine the winner of the game.

# Generate Knowledge Prompting: Knowledge

## Prompt

Input: A rock is the same size as a pebble.

Knowledge: A pebble is a clast of rock with a particle size of 4 to 64 millimetres based on the Udden-Wentworth scale of sedimentology. Pebbles are generally considered larger than granules (2 to 4 millimetres diameter) and smaller than cobbles (64 to 256 millimetres diameter).

Input: Part of golf is trying to get a higher point total than others.

Knowledge:

## Knowledge 1

The objective of golf is to play a set of holes in the least number of strokes. A round of golf typically consists of 18 holes. Each hole is played once in the round on a standard golf course. Each stroke is counted as one point, and the total number of strokes is used to determine the winner of the game.

## Knowledge 2

Golf is a precision club-and-ball sport in which competing players (or golfers) use many types of clubs to hit balls into a series of holes on a course using the fewest number of strokes. The goal is to complete the course with the lowest score, which is calculated by adding up the total number of strokes taken on each hole. The player with the lowest score wins the game.

Liu et al. 2021

# Generate Knowledge Prompting: Prediction

→ Integrate knowledge and get a prediction

Example based on Knowledge 1:

Question: Part of golf is trying to get a higher point total than others. Yes or No?

Knowledge: The objective of golf is to play a set of holes in the least number of strokes. A round of golf typically consists of 18 holes. Each hole is played once in the round on a standard golf course. Each stroke is counted as one point, and the total number of strokes is used to determine the winner of the game.

Explain and Answer:

No, the objective of golf is not to get a higher point total than others. Rather, the objective is to play a set of holes in the least number of strokes. The total number of strokes is used to determine the winner of the game, not the total number of points.

# Generate Knowledge Prompting: Prediction

→ Integrate knowledge and get a prediction

Example based on Knowledge 2:

Question: Part of golf is trying to get a higher point total than others. Yes or No?

Knowledge: Golf is a precision club-and-ball sport in which competing players (or golfers) use many types of clubs to hit balls into a series of holes on a course using the fewest number of strokes. The goal is to complete the course with the lowest score, which is calculated by adding up the total number of strokes taken on each hole. The player with the lowest score wins the game.

Explain and Answer:

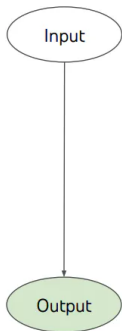
Yes, part of golf is trying to get a higher point total than others. Each player tries to complete the course with the lowest score, which is calculated by adding up the total number of strokes taken on each hole. The player with the lowest score wins the game.

# Tree of Thoughts

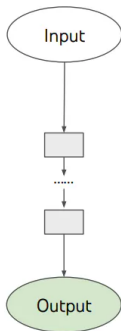
Main idea:

1. Generate multiple reasoning paths for one question/problem
2. Evaluate reasoning paths in accordance with criteria
3. Expand and prune
4. Choose final path based on the highest score

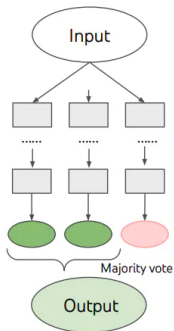
# Tree of Thoughts



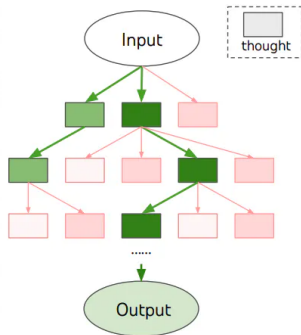
(a) Input-Output Prompting (IO)



(c) Chain of Thought Prompting (CoT)



(c) Self Consistency with CoT (CoT-SC)



(d) **Tree of Thoughts (ToT)**

Yao et al. 2024

### 3. LANGUAGE MODELS AS KNOWLEDGE BASES

## 3.1. The LAMA probe

**L**anguage **M**odel **A**nalysis probe (Petroni et al., 2019)

- How much knowledge is present in pretrained Language Models?
- Can pretrained LLMs outperform state-of-the-art NLP methods in receiving knowledge?
- How does the performance of LLMs differ for different kinds of knowledge (relational, common sense, factual)?

**”Knowledge:”**

- ▶ (subject, relation, object)
- ▶ (question, answer)



## 3.1. The LAMA probe

### Procedure

- ▶ manually convert "knowledge" (from existing knowledge sources) into cloze-statements
- ▶ Example: (Einstein, born\_in, Ulm) → "Einstein was born in [MASK]"
- ▶ ask models to predict the masked token/missing object ([MASK])

Assumption: LLM "knows" a fact, if it can predict a single object [MASK] or answer [MASK] token.

## 3.1. The LAMA probe: Considerations

### 1. Manually defined templates

Relation	Query	Answer	Generation
P19	Francesco Bartolomeo Conti was born in ____.	Florence	Rome [-1.8], <b>Florence</b> [-1.8], Naples [-1.9], Milan [-2.4], Bologna [-2.5]
P20	Adolphe Adam died in ____.	Paris	<b>Paris</b> [-0.5], London [-3.5], Vienna [-3.6], Berlin [-3.8], Brussels [-4.0]
P279	English bulldog is a subclass of ____.	dog	dogs [-0.3], breeds [-2.2], <b>dog</b> [-2.4], cattle [-4.3], sheep [-4.5]
P37	The official language of Mauritius is ____.	English	<b>English</b> [-0.6], French [-0.9], Arabic [-6.2], Tamil [-6.7], Malayalam [-7.0]

### 2. Single token prediction

→ only single token objects as prediction targets

## 3.1. The LAMA probe: Considerations

### 3. Object slot predictions

- only query object slots *not* subject or relation slots
- relations can be expressed with many different wordings: what would be the correct pattern for a relation?

### 4. Intersection of Vocabularies

- intersection of the vocabulary all models were trained on - about 21k tokens
- every model can only rank tokens of that vocabulary
- the larger the vocabulary, the harder to rank correct token

## 3.2. Knowledge sources

- ▶ Google-RE
- ▶ T-REx
- ▶ ConceptNet
- ▶ SQuAD

## 3.2. Knowledge sources

### 1. Google-RE

```
{
  "pred": "/people/person/date_of_birth",
  "sub": "/m/0j240kx",
  "obj": "1944",
  "evidences": [
    {
      "url": "http://en.wikipedia.org/wiki/Gao_Yu_(journalist)",
      "snippet": "Gao Yu (born 1944) is a Chinese journalist.",
      "considered_sentences": [
        "Gao Yu (born 1944) is a Chinese journalist ."
      ]
    }
  ],
  "sub_label": "Gao Yu",
  "obj_label": "1944",
  "masked_sentences": [
    "Gao Yu (born [MASK]) is a Chinese journalist ."
  ]
}
```

→ about 60k facts from Wikipedia

→ 3 relations (place\_of\_birth, date\_of\_birth, place\_of\_death)

→ manually defined templates

### 2. T-REx

→ 41 relations with about 1000 facts per relation from Wikidata

→ manually defined templates

## 3.3. Language Models vs. Baselines

### Language Models:

- ▶ fairseq-fconv (Fs)
- ▶ Transformer-XL (Txl)
- ▶ ELMo base (Eb)
- ▶ ELMo 5.5 (E5B)
- ▶ BERT base (Bb)
- ▶ BERT large (Bl)

### 3.3. Language Models vs. Baselines

Exercise 2 (see colab)

### 3.3. Language Models vs. Baselines

#### **Metrics for LLMs: Ranking and mean precision at k (P@k)**

- ▶ model generates output-prediction layer (logits) for possible objects which are just unnormalized numbers (eg. [2.3, -0.5, 4.6])
- ▶ softmax function is applied to those logits which converts the raw scores into probabilities that sum up to 1 (eg. [0.878,0.045,0.077])
- ▶ those probabilities are ranked in descending order (first position = highest probability)
- ▶ k is the number of predictions we consider
- ▶ if ground truth object is among these top k predictions, it's counted as a correct prediction
- ▶ calculate mean precision by dividing the correct predicted objects by all predicted objects



### 3.3. Language Models vs. Baselines

#### **Baselines:**

= existing methods/systems commonly used for *relation knowledge extraction*

- ▶ Freq (Freq)
- ▶ Relation Extraction with naive entity linking ( $RE_n$ )
- ▶ Relation Extraction with oracle entity linking ( $RE_o$ )
- ▶ DrQA

## 3.5. Results

### Results with $p@1$ :

Corpus	Relation	Statistics		Baselines		KB		LM					
		#Facts	#Rel	Freq	DrQA	RE <sub>n</sub>	RE <sub>o</sub>	Fs	Txl	Eb	E5B	Bb	Bl
Google-RE	birth-place	2937	1	4.6	-	3.5	13.8	4.4	2.7	5.5	7.5	14.9	<b>16.1</b>
	birth-date	1825	1	1.9	-	0.0	<b>1.9</b>	0.3	1.1	0.1	0.1	1.5	1.4
	death-place	765	1	6.8	-	0.1	7.2	3.0	0.9	0.3	1.3	13.1	<b>14.0</b>
	Total	5527	3	4.4	-	1.2	7.6	2.6	1.6	2.0	3.0	9.8	<b>10.5</b>
T-REx	1-1	937	2	1.78	-	0.6	10.0	17.0	36.5	10.1	13.1	68.0	<b>74.5</b>
	<i>N</i> -1	20006	23	23.85	-	5.4	<b>33.8</b>	6.1	18.0	3.6	6.5	32.4	34.2
	<i>N</i> - <i>M</i>	13096	16	21.95	-	7.7	<b>36.7</b>	12.0	16.5	5.7	7.4	24.7	24.3
	Total	34039	41	22.03	-	6.1	<b>33.8</b>	8.9	18.3	4.7	7.1	31.1	32.3
ConceptNet	Total	11458	16	4.8	-	-	-	3.6	5.7	6.1	6.2	15.6	<b>19.2</b>
SQuAD	Total	305	-	-	<b>37.5</b>	-	-	3.6	3.9	1.6	4.3	14.1	17.4

Table 2: Mean precision at one (P@1) for a frequency baseline (Freq), DrQA, a relation extraction with naïve entity linking (RE<sub>n</sub>), oracle entity linking (RE<sub>o</sub>), fairseq-fconv (Fs), Transformer-XL large (TxL), ELMo original (Eb), ELMo 5.5B (E5B), BERT-base (Bb) and BERT-large (Bl) across the set of evaluation corpora.

Petroni et al. (2019)

## 3.5. Results

### Results with p@1:

Corpus	Relation	Statistics		Baselines		KB		LM					
		#Facts	#Rel	Freq	DrQA	RE <sub>n</sub>	RE <sub>o</sub>	Fs	Tx1	Eb	E5B	Bb	Bl
Google-RE	birth-place	2937	1	4.6	-	3.5	13.8	4.4	2.7	5.5	7.5	14.9	<b>16.1</b>
	birth-date	1825	1	1.9	-	0.0	<b>1.9</b>	0.3	1.1	0.1	0.1	1.5	1.4
	death-place	765	1	6.8	-	0.1	7.2	3.0	0.9	0.3	1.3	13.1	<b>14.0</b>
	Total	5527	3	4.4	-	1.2	7.6	2.6	1.6	2.0	3.0	9.8	<b>10.5</b>

- ▶ used "standard" template for each relation
- ▶ Surprising: RE<sub>o</sub> baseline has seen at least one sentence per fact
- ▶ But: BERT prob. has sentence in training data (trained on Wikipedia)

## 3.5. Results

### Results with p@1:

Corpus	Relation	Statistics		Baselines		KB		LM					
		#Facts	#Rel	Freq	DrQA	RE <sub>n</sub>	RE <sub>o</sub>	Fs	Txl	Eb	E5B	Bb	Bl
T-REx	1-1	937	2	1.78	-	0.6	10.0	17.0	36.5	10.1	13.1	68.0	<b>74.5</b>
	N-1	20006	23	23.85	-	5.4	<b>33.8</b>	6.1	18.0	3.6	6.5	32.4	34.2
	N-M	13096	16	21.95	-	7.7	<b>36.7</b>	12.0	16.5	5.7	7.4	24.7	24.3
	Total	34039	41	22.03	-	6.1	<b>33.8</b>	8.9	18.3	4.7	7.1	31.1	32.3

- ▶ BERT way better than RE<sub>o</sub> for 1-1 relations (eg. *capital\_of*)
- ▶ results N-1 BERT\_large  $\approx$  results RE<sub>o</sub>
- ▶ RE<sub>o</sub> unrivaled for N-M relations
- ▶ general results BERT\_large  $\approx$  general results RE<sub>o</sub>

## 3.5. Results

### **Conclusion:**

- ▶ could be complicated to improve the performance of RE by providing additional data
- ▶ RE performs similar to BERT<sub>large</sub> in general and doesn't need complicated pipelines
- ▶ LMs could become a useful alternative for traditionally extracted knowledge bases
- ▶ in the future: with LLMs that are trained on even more data, they might be able to replace knowledge bases

# 4. SMALL LANGUAGE MODELS ARE ALSO FEW-SHOT LEARNERS

## 4.1 General idea

- ▶ Paper by Schick and Schütze (LMU) published in June 2021
- ▶ GPT-3 achieves great results on SuperGLUE tasks by priming
- ▶ Two problems:
  - ▶ GPT-3 is a LLM and has a large carbon footprint
  - ▶ Examples are limited to a few due to size of the context window
- ▶ Solution: Use Pattern-Exploiting Training (PET)

Schick and Schütze 2020

## 4.2 Pattern-Exploiting Training (PET)

- ▶ PET combines the idea of reformulating tasks as cloze questions with regular gradient-based finetuning
- ▶ PET additionally requires unlabeled data, unlabeled data is much easier to obtain than labeled examples for many real-world applications.
- ▶ Crucially, PET only works when the answers to be predicted by the LM correspond to a single token in its vocabulary; this is a severe limitation as many tasks cannot easily be worded that way.

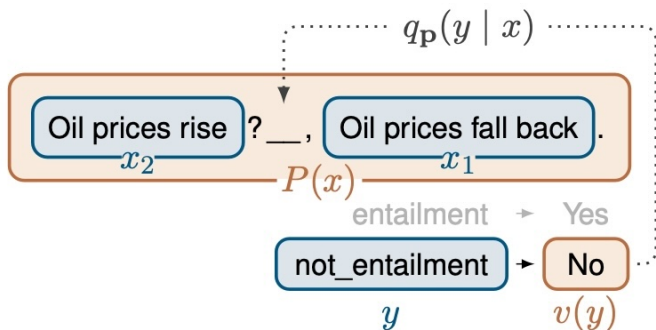


## 4.3 Pattern-Verbalizer Pairs

Each PVP  $p = (P, v)$  consists of:

- ▶ A pattern  $P : X \rightarrow T^*$  maps inputs to cloze questions containing a single mask. ( $T^*$ : set of all token sequences)
- ▶ A verbalizer  $v : Y \rightarrow T$  maps each output to a single token representing its task-specific meaning in the pattern. ( $T$ : vocabulary)

## 4.3 Pattern-Verbalizer Pairs



Application of a PVP  $p = (P, v)$  for recognizing textual entailment:

- ▶ An input  $x = (x_1, x_2)$  is converted into a cloze question  $P(x)$ .
- ▶  $q_p(y|x)$  for each  $y$  is derived from the probability of  $v(y)$  being a plausible choice for the masked position.

**iPET**: Iterative variant of PET for improved learning through iterations

## Process:

- ▶ **Initial Training:** Train an ensemble of MLMs using PET
- ▶ **Generate New Training Set:** For each model  $M_i$ :
  - ▶ Select a random subset of other models
  - ▶ Generate a new training set  $T_i$
  - ▶ Assign labels to unlabeled examples based on the subset's most confident predictions
- ▶ **Retrain Models:** Retrain each  $M_i$  on  $T_i$
- ▶ **Iterate:** Repeat the process, increasing the size of  $T_i$  by a constant factor in each iteration

## Benefits:

- ▶ **Enhanced Learning:** Models learn from different patterns and data points
- ▶ **Progressive Improvement:** Gradual increase in training data size leads to better model performance

## 4.4 GLUE and SuperGLUE

### GLUE

- ▶ Multi-task benchmark platform for Natural Language Understanding (NLU) tasks
- ▶ Consists of 9 tasks
  - ▶ CoLa: Corpus of Linguistic Acceptability
  - ▶ QQP: Quora Question Pairs
- ▶ Performance of LM's surpassed level of non-expert humans quickly

Wang et al. (2019b)

# 4.4 GLUE and SuperGLUE

## GLUE Leaderboard

GLUE SuperGLUE Paper <> Code Tasks Leaderboard FAQ Diagnostics Submit

Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AS
1	Microsoft Alexander v-team	Turing ULR v5	<a href="#">🔗</a>	91.3	73.3	97.5	94.2/92.3	93.5/93.1	76.4/90.9	92.5	92.1	96.7	93.6	97.9	55.1
2	JDExplore d-team	Vega v1		91.3	73.8	97.9	94.5/92.6	93.5/93.1	76.7/91.1	92.1	91.9	96.7	92.4	97.9	51.1
3	Microsoft Alexander v-team	Turing NLR v5	<a href="#">🔗</a>	91.2	72.6	97.6	93.8/91.7	93.7/93.3	76.4/91.1	92.6	92.4	97.9	94.1	96.9	57.1
21	Facebook AI	RoBERTa	<a href="#">🔗</a>	88.1	67.8	96.7	92.3/89.6	92.2/91.9	74.3/90.2	90.8	90.2	90.4	88.2	89.0	48.1
22	Microsoft D365 AI & MSR AI	MT-DNN-ensemble	<a href="#">🔗</a>	87.6	68.4	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9	87.4	96.0	86.3	89.0	42.1
23	GLUE Human Baselines	GLUE Human Baselines	<a href="#">🔗</a>	87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	96.9	
90	GLUE Baselines	BiLSTM+ELMo+Attn	<a href="#">🔗</a>	70.0	33.6	90.4	84.4/78.0	74.2/72.3	63.1/64.3	74.1	74.5	79.8	58.9	65.1	21.1
		BiLSTM+ELMo	<a href="#">🔗</a>	67.7	32.1	89.3	84.7/78.0	70.3/67.8	61.1/62.6	67.2	67.9	75.5	57.4	65.1	21.1
		Single Task BiLSTM+ELMo+Attn	<a href="#">🔗</a>	66.5	35.0	90.2	80.2/68.8	64.0/60.2	66.1/66.5	76.9	76.7	76.7	50.3	65.1	27.1
		Single Task BiLSTM+ELMo	<a href="#">🔗</a>	66.4	35.0	90.2	80.8/69.0	64.0/60.2	65.6/65.7	72.9	73.4	71.7	50.1	65.1	19.1
		GenSen	<a href="#">🔗</a>	66.1	7.7	83.1	83.0/76.6	79.3/79.2	59.8/82.9	71.4	71.3	78.6	59.2	65.1	20.1
		BiLSTM+Attn	<a href="#">🔗</a>	65.6	18.6	83.0	83.5/76.2	72.8/70.5	60.1/62.4	67.6	68.3	74.3	58.4	65.1	17.1
		BiLSTM	<a href="#">🔗</a>	64.2	11.6	82.8	81.8/74.3	70.3/67.8	62.5/64.2	65.6	66.1	74.6	57.4	65.1	20.1
		InferSent	<a href="#">🔗</a>	63.9	4.5	85.1	81.2/74.1	75.9/75.3	59.1/61.7	66.1	65.7	72.7	58.0	65.1	18.1
		Single Task BiLSTM	<a href="#">🔗</a>	63.7	15.7	85.9	79.4/69.3	66.0/62.8	61.4/61.7	70.3	70.8	75.7	52.8	62.3	21.1

## 4.4 GLUE and SuperGLUE

### SuperGLUE

- ▶ New and improved benchmark with more difficult and more diverse tasks, total of 8
- ▶ Retained the two hardest tasks of GLUE: Winograd Schema Challenge and Recognizing Textual Entailment
- ▶ New tasks include CommitmentBank, Words in Context and Reading Comprehension with Commonsense Reasoning

Wang et al. (2019a)

# 4.4 GLUE and SuperGLUE

## SuperGLUE Leaderboard

Leaderboard Version: 2.0

Rank	Team	Model	URL	Score	BoolQ	CB	CoQA	MuSiRC	ReCoRD	RTE	WiC	WiS	AX-b	AX-g	
1	JDExplore d-team	Vega v2	<a href="#">🔗</a>	91.3	90.5	98.699.2	99.4	88.262.4	94.493.9	96.0	77.4	96.6	-0.4	100.0/90.0	
+	2	Liam Fedus	ST-MoE-32B	<a href="#">🔗</a>	91.2	92.4	96.998.0	99.2	89.665.8	95.194.4	93.5	77.7	96.6	72.3	96.1/94.1
	3	Microsoft/Alexander v-team	Turing NLR v5	<a href="#">🔗</a>	90.9	92.0	95.997.6	98.2	88.463.0	96.495.9	94.1	77.1	97.3	67.8	93.3/95.5
	4	ERNIE Team - Baidu	ERNIE 3.0	<a href="#">🔗</a>	90.6	91.0	98.699.2	97.4	88.663.2	94.794.2	92.6	77.4	97.3	68.6	92.7/94.7
	5	Yi Tay	PLM 540B	<a href="#">🔗</a>	90.4	91.9	94.496.0	99.0	88.763.6	94.293.3	94.1	77.4	98.9	72.9	95.9/90.4
+	6	Ziwei Wang	T5 + UDQ, Single Model (Google Brain)	<a href="#">🔗</a>	90.4	91.4	95.897.6	98.0	88.363.0	94.293.5	93.0	77.9	96.6	69.1	92.7/91.9
+	7	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4	<a href="#">🔗</a>	90.3	90.4	95.797.6	98.4	88.263.7	94.594.1	93.2	77.5	96.9	66.7	93.9/93.6
	8	SuperGLUE Human Baselines	SuperGLUE Human Baselines	<a href="#">🔗</a>	89.8	89.0	95.098.9	100.0	81.851.3	91.791.3	93.6	80.0	100.0	76.6	99.3/97.7
+	9	T5 Team - Google	T5	<a href="#">🔗</a>	89.3	91.2	93.996.8	94.8	88.163.3	94.193.4	92.5	76.9	90.8	65.6	92.7/91.9
	10	SPoT Team - Google	Frozen T5 1.1 + SPoT	<a href="#">🔗</a>	89.2	91.1	95.897.6	95.6	87.961.9	93.992.4	92.9	75.8	90.6	66.9	83.1/82.6
+	11	Huawei Noah's Ark Lab	MEZHA-Plus	<a href="#">🔗</a>	86.7	87.8	94.496.0	93.6	84.605.1	90.189.6	89.1	74.6	90.2	58.0	87.1/74.1
+	12	Alibaba PAU/CBU	PAU Albert	<a href="#">🔗</a>	86.1	88.1	92.496.4	91.8	84.654.7	89.088.3	88.8	74.1	90.2	79.6	98.9/99.2
+	13	Infisy - DAWN - AI Research	RoBERTa-ICETS	<a href="#">🔗</a>	85.9	88.2	92.593.6	90.8	84.493.4	91.591.0	87.9	74.1	91.8	57.6	89.3/75.6
+	14	Tencent Jarvis Lab	RoBERTa (ensemble)	<a href="#">🔗</a>	85.9	88.2	92.593.6	90.8	84.493.4	91.591.0	87.9	74.1	91.8	57.6	89.3/75.6
	15	Zhiyi Technology	RoBERTa-emb-adv	<a href="#">🔗</a>	85.7	87.1	92.495.6	91.2	85.164.3	91.791.3	88.1	72.1	91.8	58.5	91.0/78.1
	16	Facebook AI	RoBERTa	<a href="#">🔗</a>	84.6	87.1	90.595.2	90.6	84.452.5	90.690.0	88.2	69.9	89.0	57.9	91.0/78.1
+	17	Anuar Sharafutdinov	ALLaS Team, Transformers	<a href="#">🔗</a>	82.6	88.1	91.694.8	86.8	85.154.7	82.879.8	88.9	74.1	78.8	100.0	100.0/100.0
	18	Ying Luo	FoL++(ALBERT)-Few-Shot(32 Examples)	<a href="#">🔗</a>	77.7	81.1	87.892.0	87.0	77.398.4	81.981.1	75.1	60.5	88.4	35.9	94.463.5
	19	Ralfon Bector	Text to Text PETL	<a href="#">🔗</a>	77.0	82.0	86.992.4	80.2	80.444.8	82.281.3	78.1	67.6	74.0	38.1	97.2/83.7
+	20	CASIA	INSTALL(ALBERT)-few-shot	<a href="#">🔗</a>	76.6	78.4	85.992.0	85.6	79.935.1	84.383.5	74.9	60.9	84.9	-0.4	100.0/90.0
	21	Rakesh Radhakrishnan Nelson	ADAPT (ALBERT) - few-shot	<a href="#">🔗</a>	76.0	80.0	82.392.0	85.4	76.235.7	86.185.5	75.0	55.5	86.6	-0.4	100.0/90.0
+	22	Timo Schick	IPET (ALBERT) - Few-Shot (32 Examples)	<a href="#">🔗</a>	75.4	81.2	79.988.8	90.8	74.131.7	85.985.4	70.8	49.3	88.4	36.2	97.8/87.9
	23	Adrian de Wiynter	Bort (Alexa AI)	<a href="#">🔗</a>	74.1	83.7	81.986.4	89.6	83.764.1	49.849.0	81.2	70.1	66.8	48.0	96.1/61.5
	24	IBM Research AI	BERT-ml	<a href="#">🔗</a>	73.5	84.8	89.694.0	73.8	73.290.5	74.674.0	84.1	66.2	61.0	29.6	97.8/87.3
	25	Ben Mann	GPT-3 few-shot - OpenAI	<a href="#">🔗</a>	71.8	76.4	82.075.6	92.0	75.490.5	91.190.2	69.0	49.4	80.1	21.1	90.4/55.3
	26	SuperGLUE Baselines	BERT++	<a href="#">🔗</a>	71.5	79.0	84.890.4	73.8	70.024.1	72.071.3	79.0	69.6	64.4	38.0	99.4/81.4

Click on a submission to see more information

## 4.4 GLUE and SuperGLUE

Exercise 3, see Colab



## 4.5 Results

	Model	Params (M)	BoolQ Acc.	CB Acc. / F1	COPA Acc.	RTE Acc.	WiC Acc.	WSC Acc.	MultiRC EM / F1a	ReCoRD Acc. / F1	Avg -
dev	GPT-3 Small	125	43.1	42.9 / 26.1	67.0	52.3	49.8	58.7	6.1 / 45.0	69.8 / 70.7	50.1
	GPT-3 Med	350	60.6	58.9 / 40.4	64.0	48.4	55.0	60.6	11.8 / 55.9	77.2 / 77.9	56.2
	GPT-3 Large	760	62.0	53.6 / 32.6	72.0	46.9	53.0	54.8	16.8 / 64.2	81.3 / 82.1	56.8
	GPT-3 XL	1,300	64.1	69.6 / 48.3	77.0	50.9	53.0	49.0	20.8 / 65.4	83.1 / 84.0	60.0
	GPT-3 2.7B	2,700	70.3	67.9 / 45.7	83.0	56.3	51.6	62.5	24.7 / 69.5	86.6 / 87.5	64.3
	GPT-3 6.7B	6,700	70.0	60.7 / 44.6	83.0	49.5	53.1	67.3	23.8 / 66.4	87.9 / 88.8	63.6
	GPT-3 13B	13,000	70.2	66.1 / 46.0	86.0	60.6	51.1	75.0	25.0 / 69.3	88.9 / 89.8	66.9
	GPT-3	175,000	77.5	82.1 / 57.2	92.0	72.9	<b>55.3</b>	75.0	32.5 / 74.8	<b>89.0 / 90.1</b>	73.2
	PET	223	79.4	85.1 / 59.4	<b>95.0</b>	69.8	52.4	<b>80.1</b>	<b>37.9 / 77.3</b>	86.0 / 86.5	74.1
iPET	223	<b>80.6</b>	<b>92.9 / 92.4</b>	<b>95.0</b>	<b>74.0</b>	52.2	<b>80.1</b>	33.0 / 74.0	86.0 / 86.5	<b>76.8</b>	
test	GPT-3	175,000	76.4	75.6 / 52.0	<b>92.0</b>	69.0	49.4	80.1	30.5 / 75.4	<b>90.2 / 91.1</b>	71.8
	PET	223	79.1	87.2 / 60.2	90.8	67.2	<b>50.7</b>	<b>88.4</b>	<b>36.4 / 76.6</b>	85.4 / 85.9	74.0
	iPET	223	<b>81.2</b>	<b>88.8 / 79.9</b>	90.8	<b>70.8</b>	49.3	<b>88.4</b>	31.7 / 74.1	85.4 / 85.9	<b>75.4</b>
	SotA	11,000	91.2	93.9 / 96.8	94.8	92.5	76.9	93.8	88.1 / 63.3	94.1 / 93.4	89.3

► Better than Chat GPT-3 on most of the tasks, but not SOTA

## 4.6 Analysis of the results

**What can influence the performance?**

## 4.6 Analysis of the Results

### **What can influence the performance?**

- ▶ Patterns/Templates
- ▶ Labeled and unlabeled data usage
- ▶ Model type
- ▶ Training examples

# 5. SUMMARY

# Summary

- ▶ Very intuitively and easy to understand
- ▶ Performance can vary greatly depending on multiple factors
- ▶ **But:** It can be very time and cost intensive.
- ▶ **Solution:** Automated Template Learning (next week :))

# Questions and Discussion

**Thank You for Your  
attention!**

# Literature I

- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.
- Liu, J., Liu, A., Lu, X., Welleck, S., West, P., Bras, R. L., Choi, Y., and Hajishirzi, H. (2021). Generated knowledge prompting for commonsense reasoning. *arXiv preprint arXiv:2110.08387*.
- Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., and Riedel, S. (2019). Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.
- Schick, T. and Schütze, H. (2020). It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.



## Literature II

- Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019a). SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint 1905.00537*.
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019b). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., and Narasimhan, K. (2024). Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

# Literature III

Zhang, Z., Zhang, A., Li, M., and Smola, A. (2022). Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.