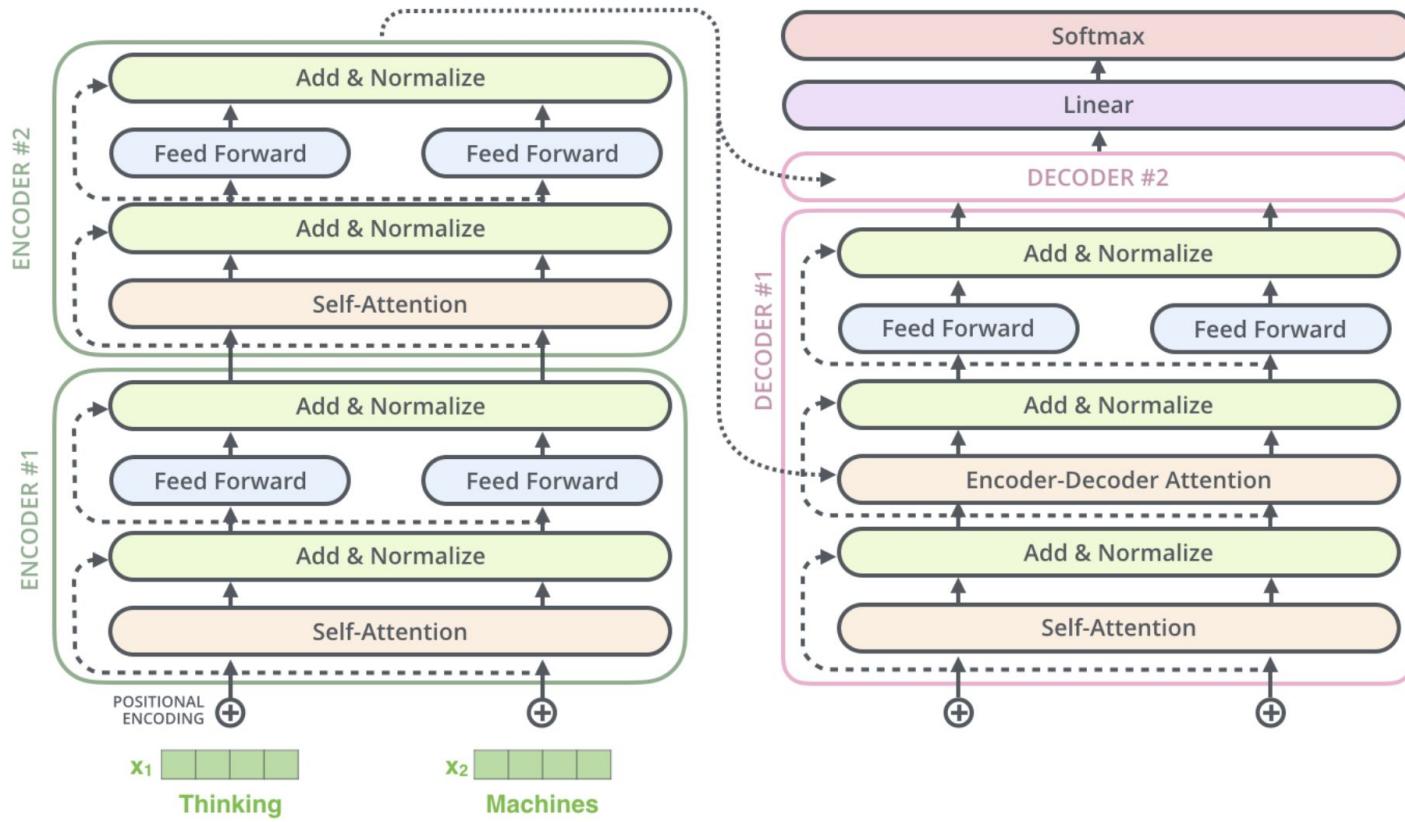


Instruction Fine-Tuning

Pawan Goyal, IIT Kharagpur

Quick Recap

Transformer with 2 stacked encoders and decoders



Masked Self-attention for decoder (to avoid seeing the future tokens)

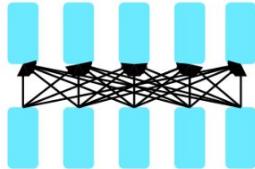
	q1·k1	−∞	−∞	−∞	−∞
N	q2·k1	q2·k2	−∞	−∞	−∞
	q3·k1	q3·k2	q3·k3	−∞	−∞
	q4·k1	q4·k2	q4·k3	q4·k4	−∞
	q5·k1	q5·k2	q5·k3	q5·k4	q5·k5

Figure 10.3 The $N \times N$ $\mathbf{Q}\mathbf{K}^\top$ matrix showing the $q_i \cdot k_j$ values, with the upper-triangle portion of the comparisons matrix zeroed out (set to $-\infty$, which the softmax will turn to zero).

<https://web.stanford.edu/~jurafsky/slp3/>

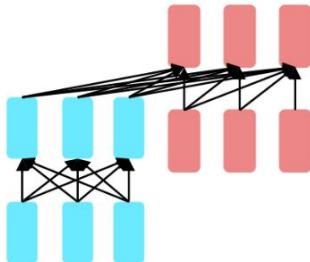
Pretraining for three types of architectures

The neural architecture influences the type of pretraining, and natural use cases.



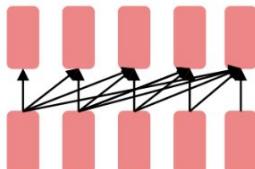
Encoders

- Gets bidirectional context – can condition on future!
- How do we train them to build strong representations?



Encoder-Decoders

- Good parts of decoders and encoders?
- What's the best way to pretrain them?



Decoders

- Language models! What we've seen so far.
- Nice to generate from; can't condition on future words

Practice Problem: *Counting the parameters*

Consider the T5-Large pretrained model, which has a model dimensionality of 1024, and a vocabulary size of 32k. It has 24 transformer layers, and has 16 attention heads (for all different attentions). The feed-forward network has 4096 nodes. How many parameters does T5-Large has? *Ignore bias terms.*

Solution

T5-Large

Voc. (Embedding Params): 1024×32000
 $\approx 32.7 \text{ m}$

Encoder

$$\left[\frac{\frac{d_{\text{model}}}{16} \times d_{\text{model}} \times 3) \times 16 + \frac{d_{\text{model}}^2}{w_0} + \frac{8 \times d_{\text{model}}^2}{ff}}{16 w_e w_k w_o} \right] \times 24$$

$\approx 302 \text{ m}$

$$= 12 \times 24 \times d_{\text{model}}^2 = 288 d_{\text{model}}^2$$

Decoder: Same number of Parameters +

Total $\approx 737 \text{ m}$

Same Encoder-Decoder (along with w_0)

$$= 4 d_{\text{model}}^2$$

$$24 \times 4 = 96 d_{\text{model}}^2 + 288 d_{\text{model}}^2 < 403 \text{ m}$$

Practice Problem: Self-attention

Suppose, you give the following input to your transformer encoder: {thinking, machines}.

The input embeddings for these two words are **[0,1,1,1,1,0]** and **[1,1,0,-1,-1,1]**, respectively. For the first attention head in your first encoder, the query, key and value matrices just take the 2 dimensions from the input each. Thus, the first 2 dimensions define the query vector, next 2 the key vector, and the final 2 the value vector. (a) What are the weights of the query, key and value transformation matrices? (b) What will be the self-attention output for the word ‘machines’ for this attention head. You are using the scaled dot vector for self-attention.

Solution

Solution 2:

(a) query \rightarrow
$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}_{6 \times 2}$$

key \rightarrow
$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}_{6 \times 2}$$

value \rightarrow
$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}_{6 \times 2}$$

(b) query for machine $[1 \ 1]$

think

machines

key $[1 \ 1]$

value $[1 \ 0]$

$[0 \ -1]$

$[-1 \ 1]$

scaled dot-product $\frac{q \cdot k}{\sqrt{2}} = \sqrt{2} \quad \frac{-1}{\sqrt{2}}$

$\rightarrow \text{att}^n \quad 0.89 \text{ (x)} \quad 0.11 \text{ (y)}$

final output $[0.78 \quad 0.11]$

$$\begin{array}{r} x-y \\ \hline y \\ \hline 13 \end{array}$$

Decoding: Sampling Strategies

Random Sampling with temperature

More probable words would have more chance of being generated

$$P(x_i|x_1, \dots, x_{i-1}) = \frac{\exp(u_i/t)}{\sum_j \exp(u_j/t)}, 0 < t \leq 1$$

Top-k Sampling

Only Top k tokens are considered for generation, so the less probable words would not have any chance (rescale).

Nucleus Sampling

Focus on the smallest set of words such that the sum of their probability is $\geq p$. Helps when the model is certain on some words.

Practice Problem

Suppose you have a vocabulary of size 5 and during decoding, the output vector is [3, -1, 2, 1, -2]. Write down the effective probability distribution when you use the following sampling strategies.

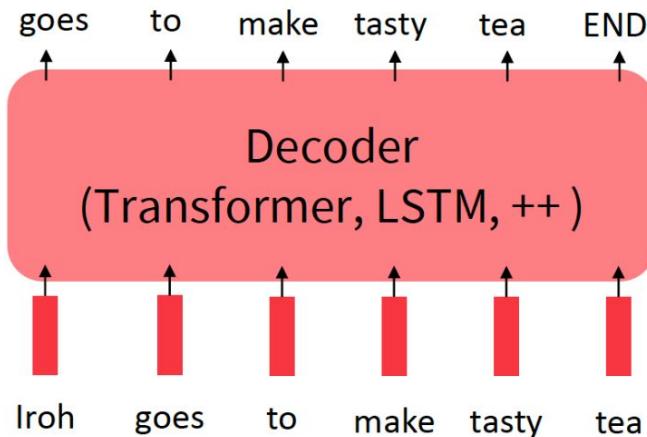
- Random sampling with temperature 0.5
- Top-2 sampling
- Nucleus sampling with $p = 0.5$

The Pretraining / Finetuning paradigm

Pretraining can improve NLP applications by serving as parameter initialization.

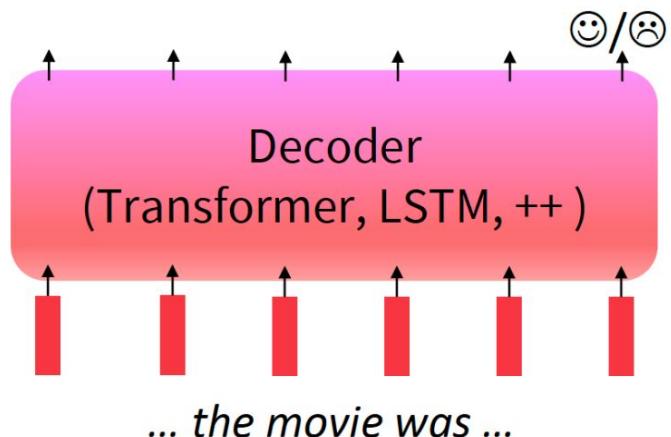
Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



Step 2: Finetune (on your task)

Not many labels; adapt to the task!



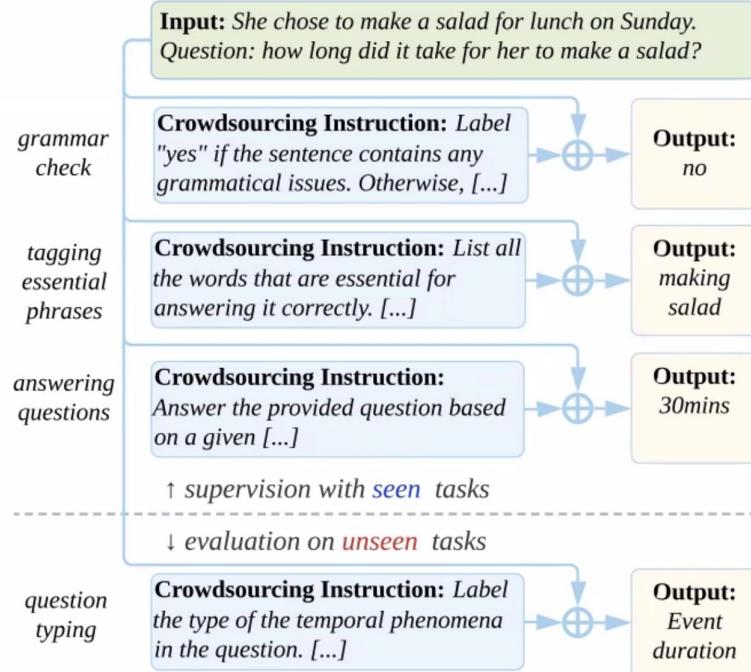
How to Learn a New Task? via Instructions?



A kid learns to play the new game just by following instructions.



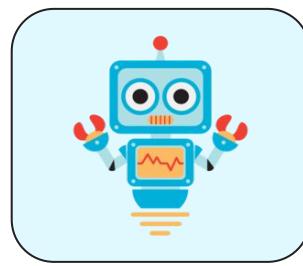
+ ...



Instruction-Tuning

Slide source: Swaroop Mishra

Evolution of Models and Our Framework



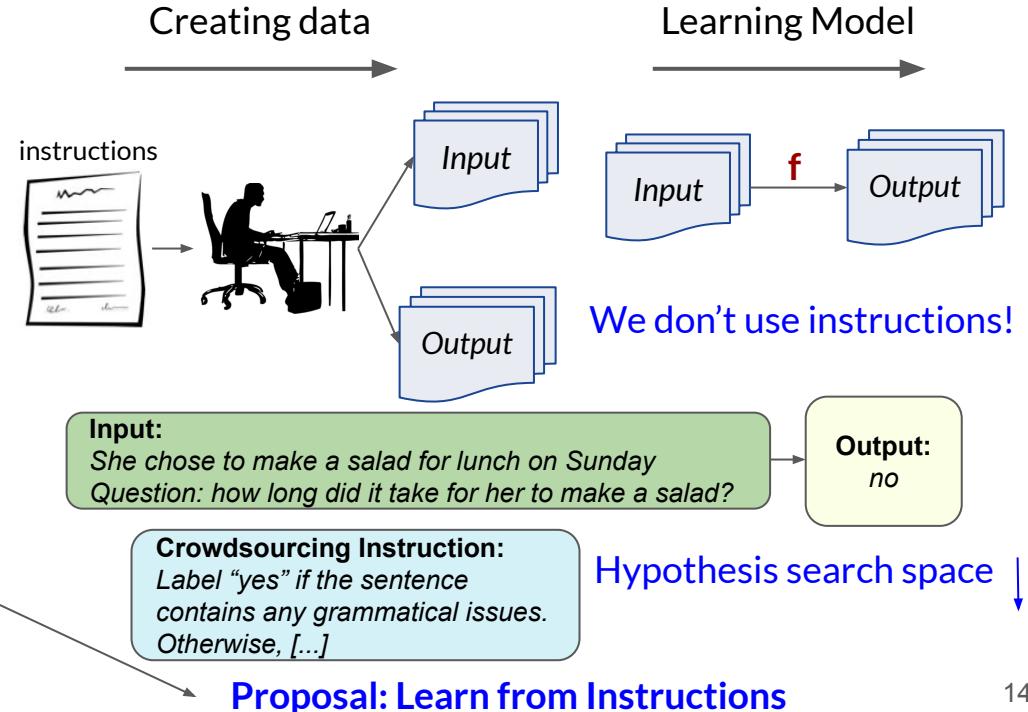
$$\text{Output} = \mathbf{f}(\text{Input})$$

\mathbf{f} is known. We can operate machines by programming
machines by programming

Model Based Design

Learn \mathbf{f} from data
Machine Learning

A layperson can use NLP resources
in instruction paradigm



Basic Premise

NLP tasks can be described via natural language instructions, such as

“Is the sentiment of this movie review positive or negative?”

or

“Translate ‘how are you’ into Chinese.”

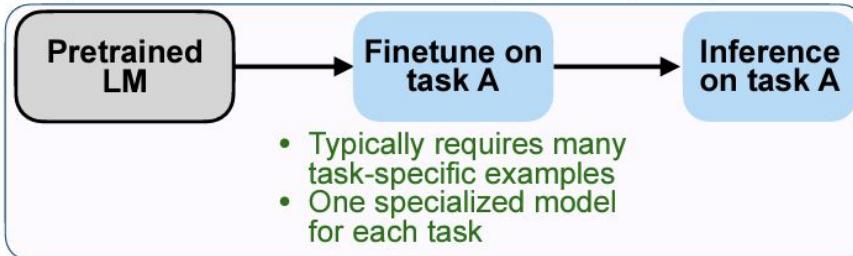
Why define tasks in natural language?

While the current dominant paradigm (supervised learning with task-specific labeled examples) has been successful in building task-specific models, ***such models can't generalize to unseen tasks***; for example, a model that is supervised to **solve questions** cannot solve a **classification task**.

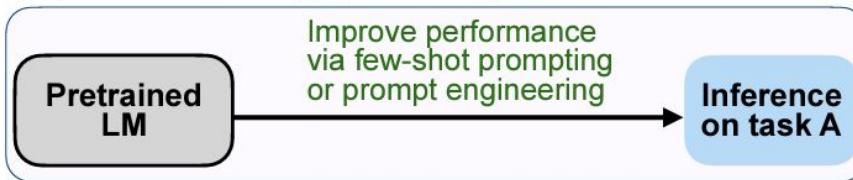
We hypothesize that **a model equipped with understanding and reasoning with natural language instructions should be able to generalize** to any task that can be defined in terms of natural language.

How is instruction tuning a different paradigm?

(A) Pretrain–finetune (BERT, T5)



(B) Prompting (GPT-3)



(C) Instruction tuning (FLAN)

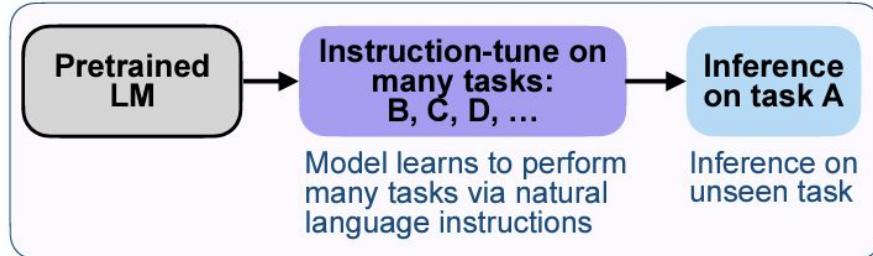


Figure 2: Comparing instruction tuning with pretrain–finetune and prompting.

Finetune on many tasks (“instruction-tuning”)

Input (Commonsense Reasoning)

Here is a goal: Get a cool sleep on summer days.

How would you accomplish this goal?

OPTIONS:

- Keep stack of pillow cases in fridge.
- Keep stack of pillow cases in oven.

Target

keep stack of pillow cases in fridge

Sentiment analysis tasks

Coreference resolution tasks

...

Input (Translation)

Translate this sentence to Spanish:

The new office building was built in less than three months.

Target

El nuevo edificio de oficinas se construyó en tres meses.

Inference on unseen task type

Input (Natural Language Inference)

Premise: At my age you will probably have learnt one lesson.

Hypothesis: It's not certain how many lessons you'll learn by your thirties.

Does the premise entail the hypothesis?

OPTIONS:

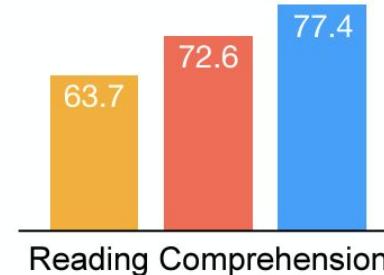
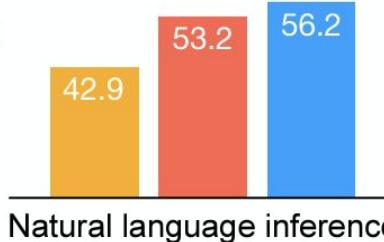
- yes
- it is not possible to tell
- no

FLAN Response

It is not possible to tell

Performance
on unseen
task types

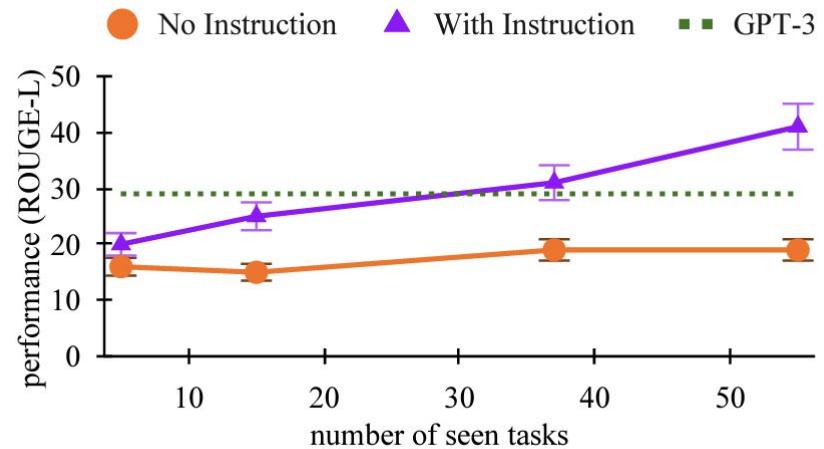
■ GPT-3 175B zero shot ■ GPT-3 175B few-shot ■ FLAN 137B zero-shot



Generalization to unseen tasks

Task	Instance-Level Generalization	Task-Level Generalization
Training data	$X^{\text{train}}, Y^{\text{train}}$	$(I_t, X_t^{\text{train}}, Y_t^{\text{train}})_{t \in \mathcal{T}_{\text{seen}}}$
Evaluation	$x \rightarrow y$ where: $(x, y) \in (X^{\text{test}}, Y^{\text{test}})$	$(x, I_t) \rightarrow y$ where: $(x, y) \in (X_t^{\text{test}}, Y_t^{\text{test}})_{t \in \mathcal{T}_{\text{unseen}}}$

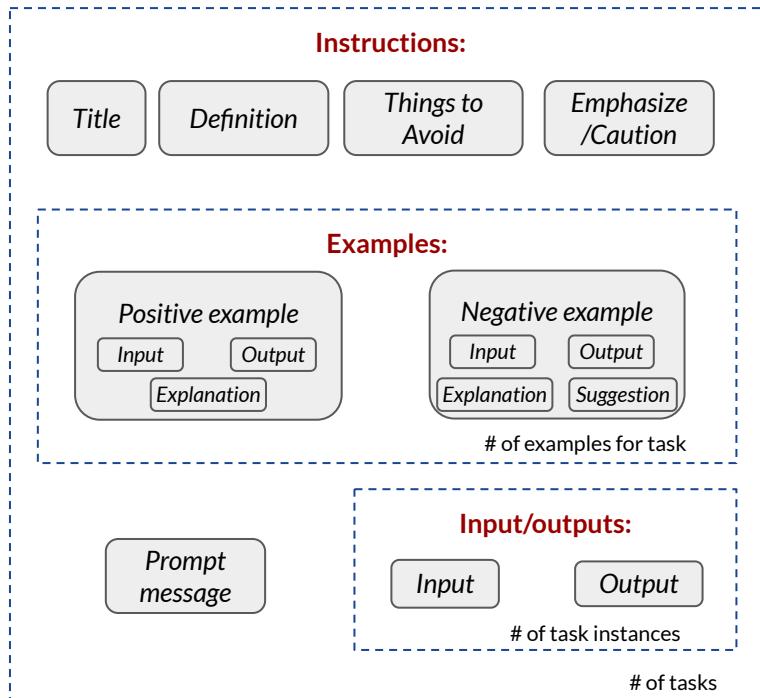
(a) A comparison of *task* vs *instance-level* generalization I_t , X_t and Y_t indicate natural language instructions, input, and output sets respectively for task t . In the conventional setup, training and evaluation are done on the instances of the same task. However, in task-level generalization, a model is expected to generalize to **unseen** tasks, where $\mathcal{T}_{\text{unseen}} \cap \mathcal{T}_{\text{seen}} = \emptyset$.



(b) BART evaluation on *unseen* tasks (*y*-axis is perf. on $\mathcal{T}_{\text{unseen}}$) when supervised with *seen* tasks (*x*-axis is $|\mathcal{T}_{\text{seen}}|$). A model using **instructions** (I_t) consistently improves with more observed tasks. In contrast, models with **no access to the instructions** show no sign of improved generalization. Details in §6.3.

Figure 2: The formal definition of generalization to unseen tasks (a) and a summary of its empirical outcome (b).

Representing Instructions: Instruction Schema



Natural Instructions: Annotation

Sentence:

There will be a sentence will be shown here ...

Ask a question regarding event duration

Question: You will type a proper English question here

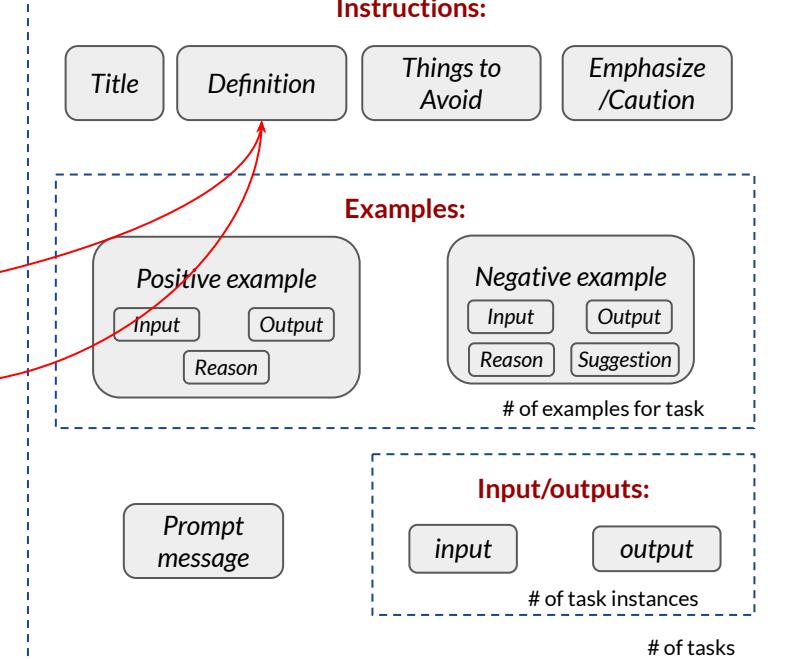
Answer: You will provide the answer to your question here

The extracted question-answer pairs have to meet two goals, which we will elaborate in the next two sections:

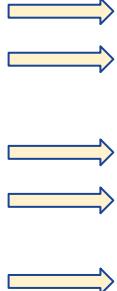
1. Temporal Reasoning

The question has to require "temporal" understanding regarding some events in the sentence. There are five categories of temporal properties that we are interested in, and in this task we ask you to create one question answer pairs per category.

- **Event Duration for events:** is the understanding of how long events last. For example, "brushing teeth", usually takes few minutes.
- **Transient v. Stationary events:** the sentence "he was born in the U.S." is a stationary event since it will last forever; however, "he is hungry" is transient since it continues for short period of time. This category means the understanding of whether an event can be changed in the future.
- **Event Ordering** is the understanding of how events are usually ordered in nature. For example, "earning money" usually comes before "spending money"
- **Absolute Timepoint:** is our understanding of when events usually happen. For example, "going to school" usually happens during the day, not at 2 A.M.
- **Frequency:** refers to how many times an event is likely to be repeated during a period. For example, I take showers 5 times a week, a lottery draws 3 times a week, etc.



Natural Instructions: Examples



question generation (from MC-TACO)

- **Title:** Writing questions that involve commonsense understanding of "event duration".
- **Definition:** In this task, we ask you to write a question that involves "event duration", based on a given sentence. Here, event duration is defined as the understanding of how long events typically last. For example, "brushing teeth", usually takes few minutes.
- **Emphasis & Caution:** The written questions are not required to have a single correct answer.
- **Things to avoid:** Don't create questions which have explicit mentions of answers in text. Instead, it has to be implied from what is given. In other words, we want you to use "instinct" or "common sense".

Positive Example

- **Input:** Sentence: Jack played basketball after school, after which he was very tired.
- **Output:** How long did Jack play basketball?
- **Reason:** the question asks about the duration of an event; therefore it's a temporal event duration question.

Negative Example

- **Input:** Sentence: He spent two hours on his homework.
 - **Output:** How long did he do his homework?
 - **Reason:** We DO NOT want this question as the answer is directly mentioned in the text.
 - **Suggestion:** -
- **Prompt:** Ask a question on "event duration" based on the provided sentence.

Task Instance

- **Input:** Sentence: Still , Preetam vows to marry Nandini if she meets him again .
• **Expected Output:** How long had they known each other?

Task Instance

- **Input:** Sentence: It's hail crackled across the comm, and Tara spun to retake her seat at the helm.
• **Expected Output:** How long was the storm?

Task Instance

- **Input:** Sentence: Max and Joey would often run through fields in a game of chase.
• **Expected Output:** How long would they run through the fields?

Instructions remain common across instances

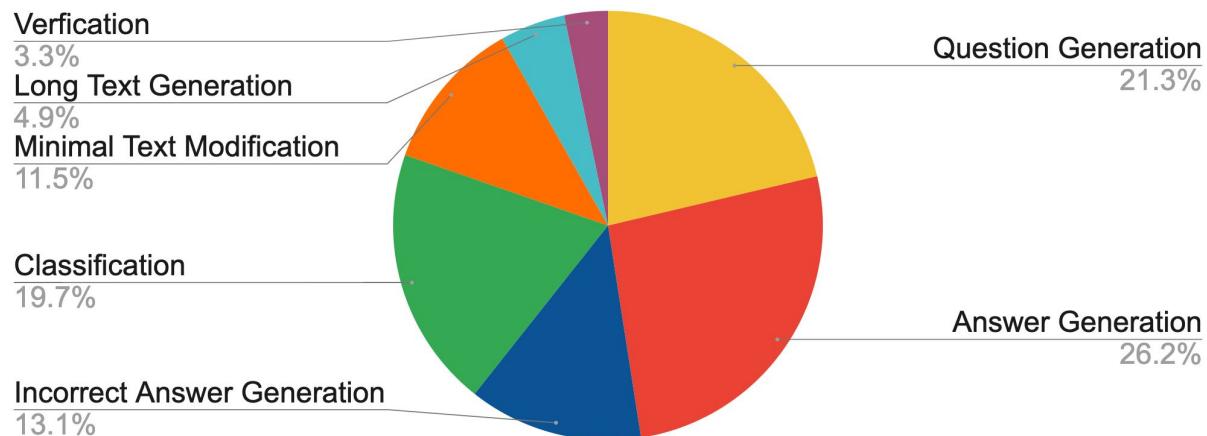
Natural Instructions: Statistics

- Tasks: ~61
- Instances: ~193k
- Categories: ~7
- Diverse Reasoning Skills:

*E.g. Numerical Reasoning,
Coreference Resolution,
Commonsense Reasoning,
Multi-hop Reasoning.*

- Diverse Domains:

*E.g. Sports, History, News,
Conversations, Geography,
NFL games, Captions, Maths.*



Instruction Encoding

Prompt : I_t^{prompt}

Definition : $I_t^{\text{Definition}}$

Things to Avoid : $I_t^{\text{avoid.}}$

Emphasis&Caution : $I_t^{\text{emph.}}$

NegativeExample1—

input : $I_t^{\text{pos. ex.}}$, **output** : $I_t^{\text{pos. ex.}}$, **reason** : $I_t^{\text{pos. ex.}}$

PositiveExample1—

input : $I_t^{\text{pos. ex.}}$, **output** : $I_t^{\text{pos. ex.}}$, **reason** : $I_t^{\text{pos. ex.}}$

input : x , **output** : ”

I refers to the text of a component c in the instruction schema.

Learning from “Instructions” vs “examples only”

model ↓	evaluation set $\mathcal{T}_{\text{unseen}} \rightarrow$	random split	leave-one-	leave-one-	leave-one-
		of tasks	category (QG)	dataset (QASC)	task (QASC QG)
BART (fine-Tuned)	MULTITASK LEARNING	13	6	37	20
	INSTRUCTION TUNING	32	17	51	56
GPT3 (not fine-tuned)	FULL INSTRUCTIONS	24	33	22	33
:					
BART (fine-Tuned)	SUPERVISED UPPERBOUND	67	59	60	66

Contribution of Instruction Encodings

model ↓	task category →	QG	AG	CF	IAG	MM	VF	avg
BART (fine-tuned)	NO INSTRUCTION	26	6	0	21	33	7	13
	PROMPT	27	22	7	22	34	9	20
	+DEFINITION	35	24	50	25	36	7	30↑ (+50)
	+THINGS TO AVOID	33	24	4	24	58	9	25↑ (+25)
	+EMPHASIS	38	23	16	26	49	3	26↑ (+30)
	+POS. EXAMPLES	53	22	14	25	17	7	23↑ (+15)
	+DEFINITION+POS. EXAMPLES	51	23	56	25	37	6	33↑ (+65)
	POS. EXAMP.	55	6	18	25	8	6	20
GPT3 (not fine-tuned)	FULL INSTRUCTION	46	25	52	25	35	7	32↑ (+60)
	FULL INSTRUCTION	33	18	8	12	60	11	24 (+11)

QG: Question Generation, AG: Answer Generation, CF: Classification, IAG: Incorrect Answer Generation, MM: Minimal Text Modification, VF: Verification.

Importance of Instructions for Humans

Category	Helpful Fields	Explanation
Question Generation (QG)	1. DEFINITION 2. EMPHASIS & CAUTION 3. POSITIVE EXAMPLES 4. NEGATIVE EXAMPLES	- Provides a holistic picture of the task. - Provides key information for solving the task. - This gives an idea of what is expected in the output. - Good to know the common mistakes people do.
Answer Generation (AG)	1. PROMPT 2. DEFINITION 3. POSITIVE EXAMPLES	- It limits the exploration space to question spans. - Provides a general understanding of the task. - Reason field is very helpful.
Classification (CF)	1. DEFINITION	- The task is unclear without this field.
Incorrect Answer Generation (IAG)	1. DEFINITION 2. EMPHASIS & CAUTION 3. POSITIVE EXAMPLES	- Helps understand the utility of such a task. - Source of some useful shortcuts. - Helps in understanding the type of questions asked.
Minimal Text Modification (MM)	1. THINGS TO AVOID	- Provides critical information.
Verification (VF)	1. DEFINITION 2. THINGS TO AVOID 3. POSITIVE EXAMPLES 4. NEGATIVE EXAMPLES	- Makes the task easy to understand. - Contains useful tips required for this task. - Exemplifies task understanding. - Helps avoid potential mistakes.

Excluding Negative Examples Help!

Model ↓	Split ↓	w/ neg. examples	w/o neg. examples
BART	random	32	35
	leave-one- x		
	↳ x = category (AG)	19	21
	↳ x = dataset (Quoref)	37	37
GPT3	↳ x = task (QASC QG)	56	57
	-	24	44

Negative instructions are surprisingly difficult for the models to learn from.

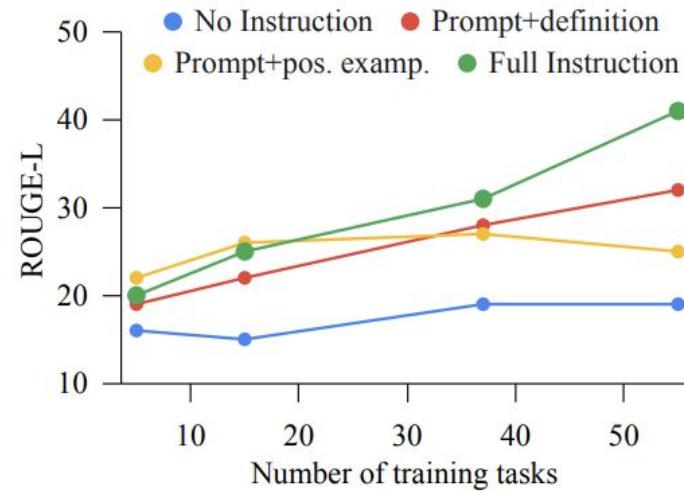
Prompts vs Instructions

task	Natural Instructions	PromptSource (Sanh et al. 2021)
CosmosQA (question answering)	<ul style="list-style-type: none">* Definition: Craft one correct answer to the question given in input. To make it more interesting, try to use non-stereotypical language if possible. Make sure your correct answer is reasonably long, consistent with the context, and requires common sense (instead of explicit extraction from the context.)* Emphasis: 1. In your answer, use as few words as possible from the given context. 2. Use a response that is uncommon/non-stereotypical, so that it is less predictable. 3. To be less repetitive, please vary your language for each question.* Prompt: Craft one correct answer to the question given in input. Context: {{ context }} Question: {{ question }}	<p>According to the above context, choose the best option to answer the following question.</p> <p>Question: {{ question }} Options: {{answer_choices}}</p>

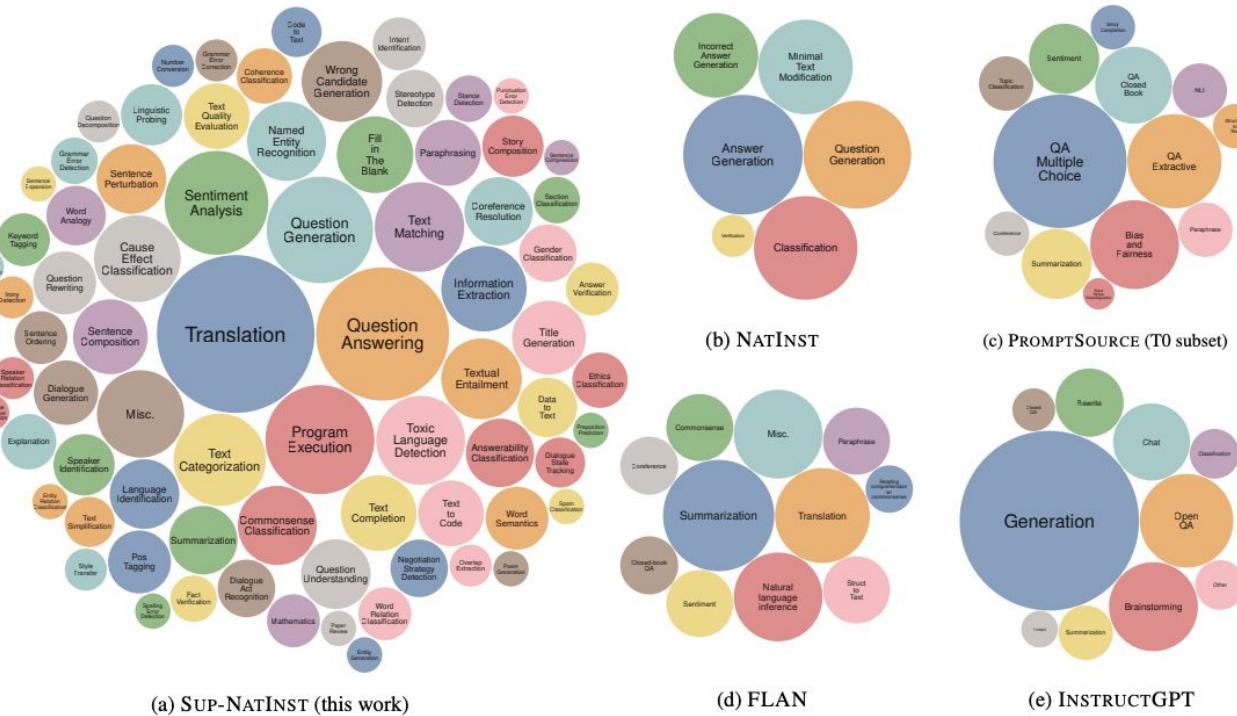
Task	Model	PromptSource	NATURAL INSTRUCTIONS
Quoref QA (002)	GPT3-Instruct	43	47
	GPT3	2	13
DROP QA (028)	GPT3-Instruct	6	10
	GPT3	2	3

Scaling Laws

- Cross-task generalization with Instruction improves with increase in number of observed tasks.
- Natural Instructions v2/ Super Natural Instructions:
 - 1600 + tasks
 - 73 categories
 - 74 domains
 - 43 languages



Super-NaturalInstructions: Size and diversity



Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ Tasks. EMNLP 2022

Yizhong Wang*, Swaroop Mishra*,..., Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi and Daniel Khashabi

Task Type	Coreference Resolution
Task ID	task1391_winogrande_coreference_resolution
Definition	In this task, you are given a question containing a blank (_) and two options. You should pick the best option to answer the question. Please answer with “A” or “B”.
Positive Example	<p>Input: Katrina gave Christine a stuffed animal for their birthday, but _ already had this one. (A) Katrina (B) Christine</p> <p>Output: B</p> <p>Explanation: Since the blank is someone who received the gift and already had a stuffed animal, the answer must be “Christine”.</p>
Negative Example	<p>Input: Kevin had to use less sunscreen when at the beach tanning than Justin because _ had less sensitive skin. (A) Kevin (B) Justin</p> <p>Output: (A)</p> <p>Explanation: Here, an additonal parentheses has been added to the answer. Note that, a valid answer must be “A” or “B”.</p>
Instance	<p>Input: Benjamin hated being in the sand and just watched Nelson make castle since _ hated to be messy. (A) Benjamin (B) Nelson</p> <p>Valid Output: [“A”]</p>

Table 8: An example task in the Cause Effect Classification category of our dataset, adopted from Wino-Grande ([Sakaguchi et al., 2020](#)).

Task Type	Dialogue Act Recognition
Task ID	task1394_meta_woz_dialogue_act_recognition
Definition	<p>In this task, you are given four sentences: a bot task sentence, a bot role sentence, a user task sentence and a user role sentence. Your job is to classify given sentences into one of the 47 different domains. The domains are: “UPDATE CALENDAR”, “PRESENT IDEAS”, “MOVIE LISTINGS”, “AUTO SORT”, “GAME RULES”, “CONTACT MANAGER”, “BANK BOT”, “MUSIC SUGGESTER”, “CHECK STATUS”, “PET ADVICE”, “HOW TO BASIC”, “NAME SUGGESTER”, “QUOTE OF THE DAY BOT”, “GUINNESS CHECK”, “INSURANCE”, “RESTAURANT PICKER”, “MAKE RESTAURANT RESERVATIONS”, “WEDDING PLANNER”, “SKI BOT”, “HOME BOT”, “PLAY TIMES”, “BUS SCHEDULE BOT”, “WHAT IS IT”, “PHONE PLAN BOT”, “DECIDER BOT”, “PHONE SETTINGS”, “TIME ZONE”, “LIBRARY REQUEST”, “UPDATE CONTACT”, “CATALOGUE BOT”, “PROMPT GENERATOR”, “SCAM LOOKUP”, “SPORTS INFO”, “POLICY BOT”, “CITY INFO”, “APARTMENT FINDER”, “EVENT RESERVE”, “SHOPPING”, “EDIT PLAYLIST”, “LOOK UP INFO”, “ORDER PIZZA”, “WEATHER CHECK”, “APPOINTMENT REMINDER”, “GEOGRAPHY”, “STORE DETAILS”, “AGREEMENT BOT”, “ALARM SET”.</p>
Positive Example	<p>Input: Bot’s task: Inform the user that the topping they are asking for is unavailable. Bot’s role: You are a bot designed to help customers order pizza. User’s task: Ask if a certain pizza topping is available. User’s role: You are interacting with a pizza restaurant bot. Output: ORDER_PIZZA Explanation: According to the descriptions of the four sentences, we know that the type of task is ORDER_PIZZA.</p>
Negative Example	<p>Input: Bot’s task: Help the user with their pizza order. Bot’s role: You are a bot designed to help customers order pizza. User’s task: Ask the bot for three different pizzas. User’s role: You are interacting with a pizza restaurant bot. Output: UPDATE_CALENDAR Explanation: According to the descriptions of the tasks and roles, we know that the type of task is ORDER_PIZZA, but the output is UPDATE_CALENDAR, so it is incorrect.</p>
Instance	<p>Input: Bot’s task: Tell the user when the movie is playing on Friday night. Bot’s role: You are a bot designed to provide movie listings. User’s task: Ask the bot for the movie times for a movie on Friday night. User’s role: You are interacting with a bot designed to provide movie listings. Valid Output: [“MOVIE_LISTINGS”]</p>

Task Type	Question Rewriting
Task ID	task1622_disfl_qa_question_rewriting
Definition	Convert a disfluent question to a proper question. A disfluent question is a question that has some interruptions in it while framing. A proper question is the correct form of the question without any disfluency.
Positive Example	<p>Input: Why was uh where was the Rhine regulated with an upper canal?</p> <p>Output: Where was the Rhine regulated with an upper canal?</p> <p>Explanation: The above disfluent question is correctly converted to a proper question. The 'uh' and the 'why' before the correction should be removed to make it fluent.</p>
Negative Example	<p>Input: When did Maududi exert the least impact no where did he exert?</p> <p>Output: When did Maududi exert the least impact?</p> <p>Explanation: The above disfluent question is not correctly converted to a proper question since the original meaning in the disfluent question is where did Maududi exert the least impact but the output asks when did Maududi exert the least impact.</p>
Instance	<p>Input: What kind of, no hold up, what describes the proportionality of acceleration to force and mass?</p> <p>Valid Output: ["What describes the proportionality of acceleration to force and mass?"]</p>

Table 14: An example task in the Question Rewriting category of our dataset, adopted from Disfl-QA ([Gupta et al., 2021](#)).

Super-NaturalInstructions: Performance

Models Trained:

Tk-INSTRUCT

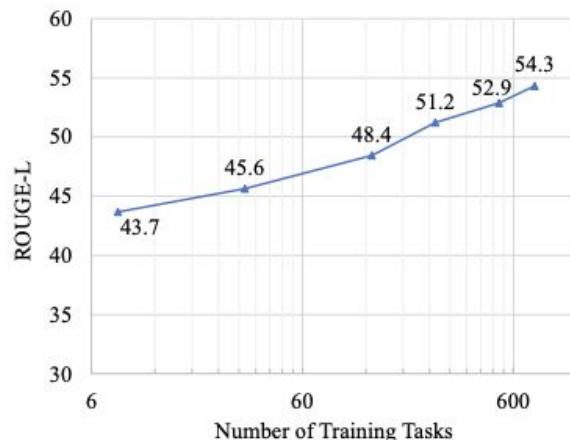
(instruction
fine-tuning T5)

mTk-INSTRUCT

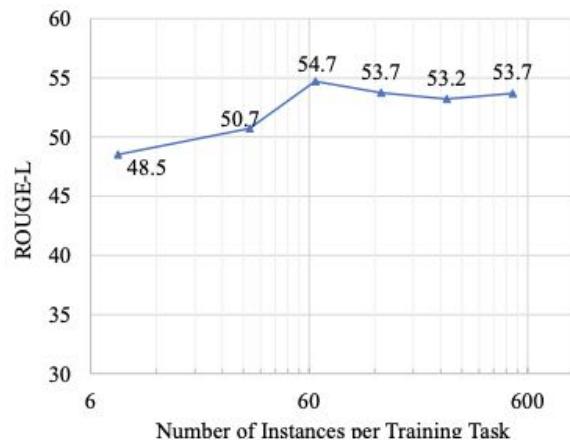
(instruction
fine-tuning mT5)

		Methods ↓ / Evaluation →	En	X-lingual
Heuristic Baselines	Copying Instance Input	14.2	5.4	
	Copying Demo Output	28.5	50.3	
Pretrained LMs	T5-LM (11B)	30.2	–	
	GPT3 (175B)	45.0	51.3	
Instruction-tuned Models	T0 (11B)	32.3	–	
	InstructGPT (175B)	52.1	52.8	
	Tk-INSTRUCT (ours, 11B)	62.0	–	
	mTk-INSTRUCT (ours, 13B)	57.1	66.1	
Upper-bound (est.)	Supervised Training	74.3	94.0	

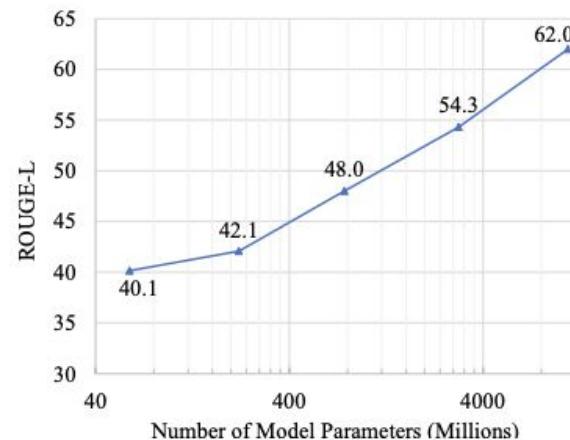
Super-NaturalInstructions: Scaling Laws



(a)



(b)

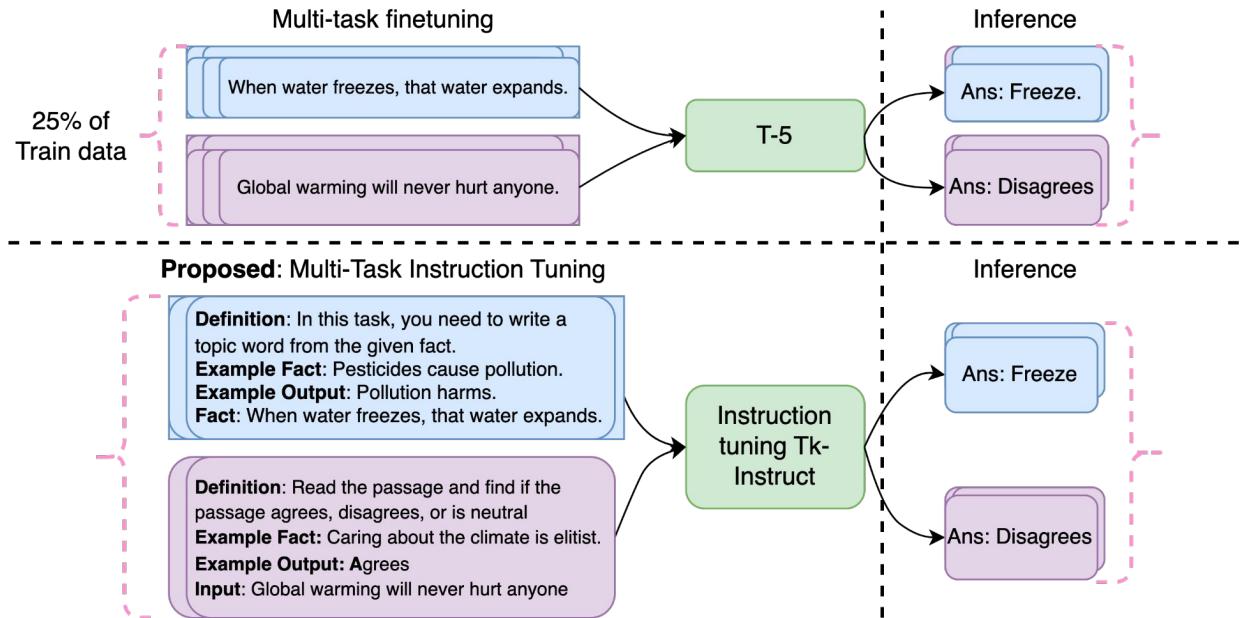


(c)

Super-NaturalInstructions: Ablations

Testing Encoding → Training Encoding ↓	Task ID	Def	Pos (1)	Def + Pos (1)	Pos (2)	Def + Pos (2)	Def + Pos (2) + Neg (2)	Def + Pos (2) + Neg (2) + Expl	Pos (4)	Def + Pos (4)	Average
Task ID	<u>21.2</u>	33.3	16.8	30.9	23.0	33.7	33.9	31.6	26.0	36.4	33.9
Def	17.3	<u>45.0</u>	31.1	43.8	36.4	46.4	44.2	44.3	38.0	46.0	39.9
Pos (1)	10.9	22.1	<u>43.9</u>	47.8	46.6	49.2	46.2	43.4	46.6	49.5	43.1
Def + Pos (1)	11.1	42.2	43.8	<u>52.4</u>	47.4	53.3	53.1	<u>51.8</u>	47.8	53.7	44.5
Pos (2)	12.7	<u>22.4</u>	47.1	50.2	<u>49.3</u>	52.3	50.6	46.7	49.8	52.4	45.0
Def + Pos (2)	12.4	42.1	44.5	<u>52.4</u>	49.0	<u>54.3</u>	53.5	<u>52.7</u>	50.3	<u>54.8</u>	46.4
Def + Pos (2) + Neg (2)	14.0	42.3	43.6	<u>51.8</u>	48.6	<u>53.5</u>	<u>54.3</u>	50.2	49.6	<u>53.8</u>	45.9
Def + Pos (2) + Neg (2) + Expl	15.4	42.0	43.8	50.7	47.6	51.9	52.5	<u>52.6</u>	48.6	52.2	44.3
Pos (4)	11.0	<u>23.9</u>	45.6	49.8	49.0	51.7	49.5	47.5	<u>49.8</u>	51.3	44.5
Definition + Pos (4)	11.0	42.4	44.3	<u>51.9</u>	48.7	53.7	53.4	50.6	50.5	<u>53.5</u>	46.0

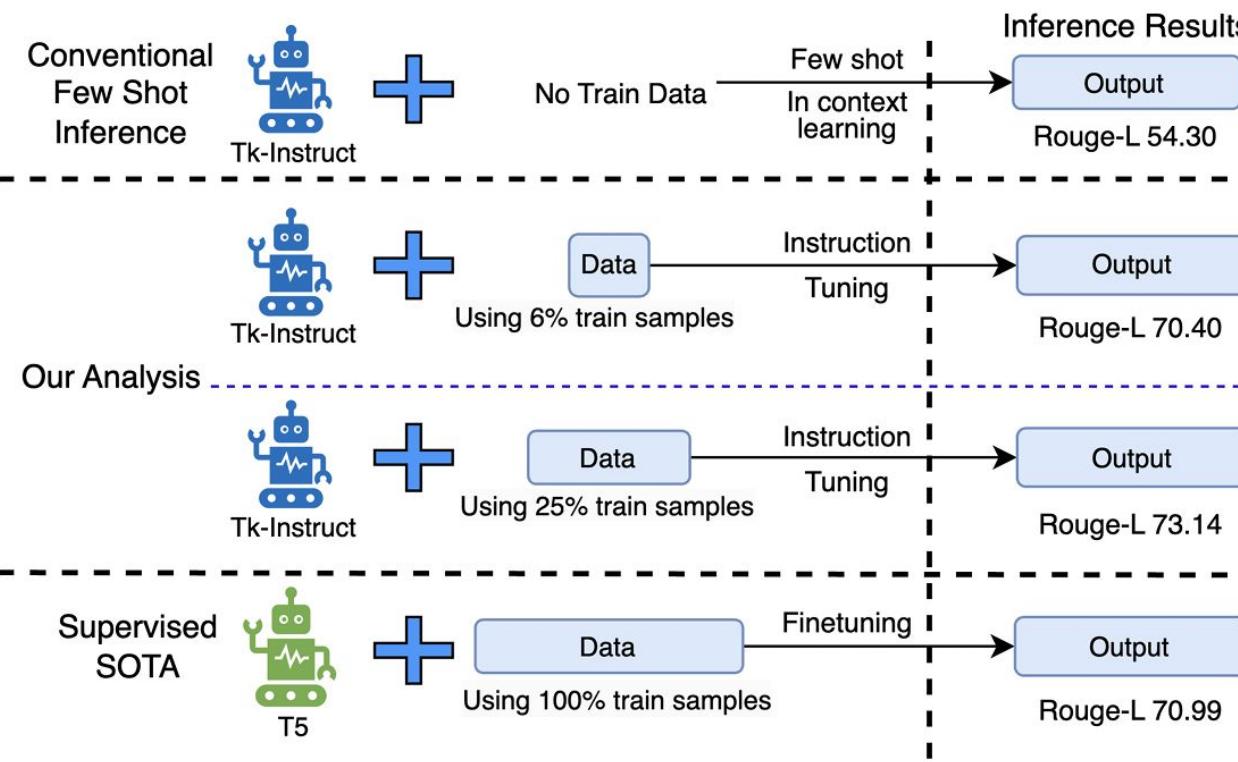
Instruction-tuned Models are Quick Learners



- need just 6% training data to get comparable results with SoTA
- outperforms SOTA by 7% with 100% training data

Instruction Tuned Models are quick Learners

Himanshu Gupta*, Saurabh Arjun Sawant*, **Swaroop Mishra**, Santosh Mashetty, Mutsumi Nakamura, Arindam Mitra, Chitta Baral

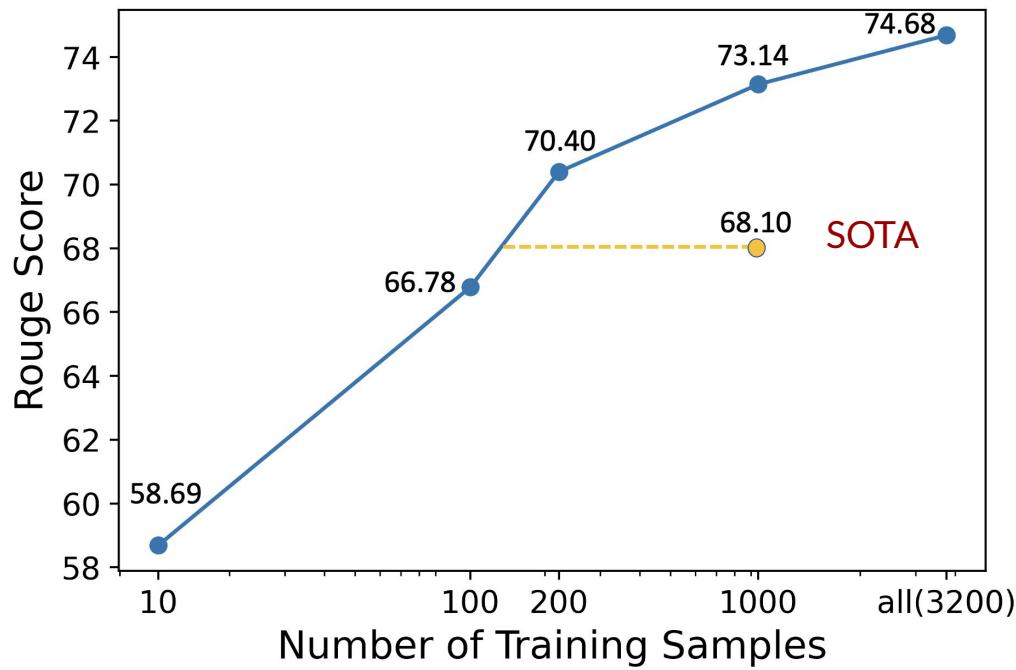


Instruction Tuned Models are quick Learners

Himanshu Gupta*, Saurabh Arjun Sawant*, Swaroop Mishra, Santosh Mashetty, Mutsumi Nakamura, Arindam Mitra, Chitta Baral

Figure 1: Showcasing the difference between the few shot inference, fully supervised finetuning, and our proposed analysis. The first row represents conventional few shot inference using Tk-Instruct which results in a score of 54.30. The fourth row indicates supervised SOTA that uses 100% of downstream train data to finetune T5-3B to get a SOTA score of 70.99. **Our findings** demonstrate the quick learning ability of the instruction tuned model. Using only 6% of downstream train data, Tk-Instruct achieved a score of 70.40. Surpassing SOTA by 2 points with 25% of downstream train data, our results highlight the MTL setting.

Instruction-tuned Models are Quick Learners...



Instruction Tuned Models are quick Learners

Himanshu Gupta*, Saurabh Arjun Sawant*, **Swaroop Mishra**, Santosh Mashetty, Mutsumi Nakamura, Arindam Mitra, Chitta Baral

Flan

- ▶ Pre-train, then fine-tune on a bunch of tasks, generalize to unseen tasks
- ▶ Scale the number of tasks, models size (Flan-T5, Flan-Palm), and fine-tuning on chain-of-thought data

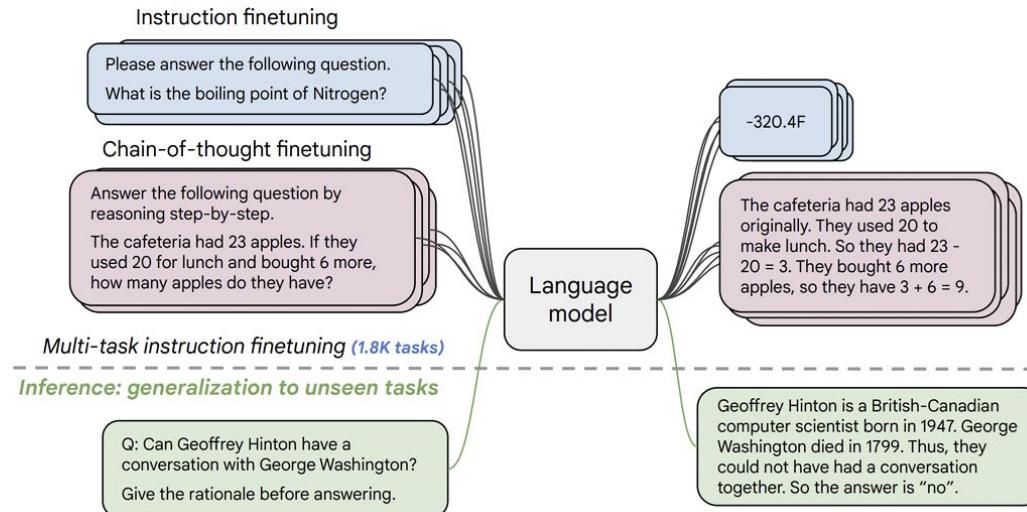
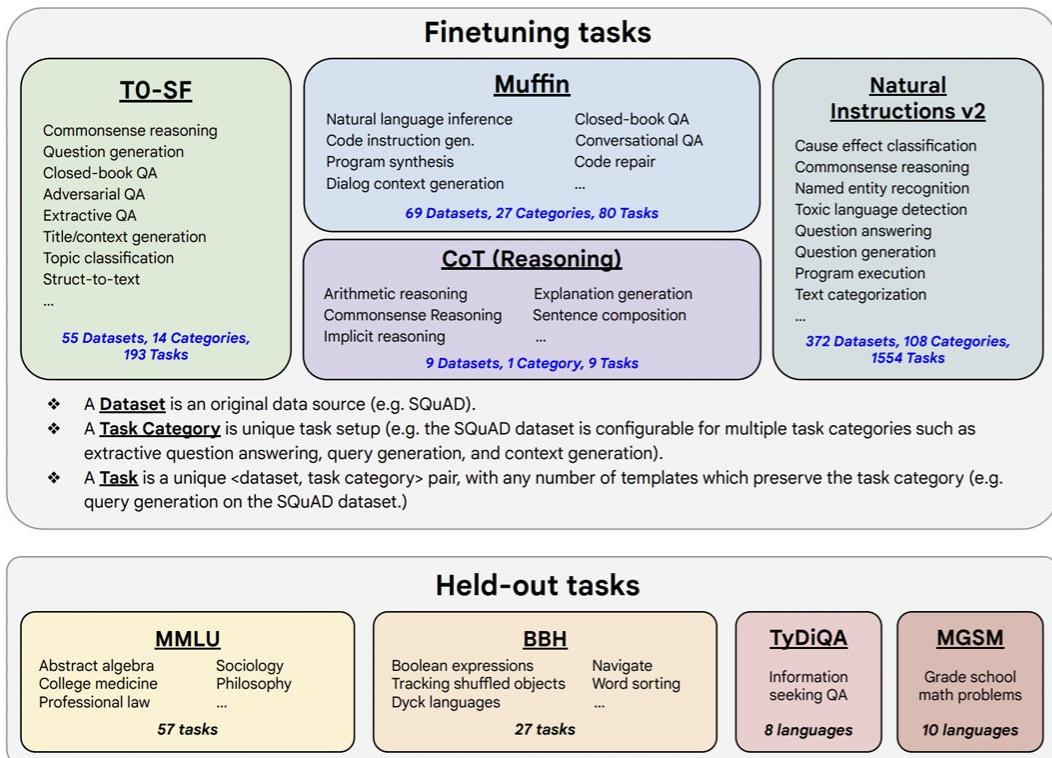


Figure 1: We finetune various language models on 1.8K tasks phrased as instructions, and evaluate them on unseen tasks. We finetune both with and without exemplars (i.e., zero-shot and few-shot) and with and without chain-of-thought, enabling generalization across a range of evaluation scenarios.

Chung et al. (2022)

Flan



- ▶ Fine-tuned on 473 datasets, 1836 tasks.
- ▶ Some datasets support multiple tasks
- ▶ E.g. SQuAD can be used for QA or question generation.

Chung et al. (2022)

Flan: Generating multiple templates from the same data

Premise

Russian cosmonaut Valery Polyakov set the record for the longest continuous amount of time spent in space, a staggering 438 days, between 1994 and 1995.

Hypothesis

Russians hold the record for the longest stay in space.

Target

Entailment
Not entailment



Options:
- yes
- no

Template 1

<premise>

Based on the paragraph above, can we conclude that
<hypothesis>?

<options>

Template 2

<premise>

Can we infer the following?

<hypothesis>

<options>

Template 3

Read the following and determine if the hypothesis can be inferred from the premise:

Premise: <premise>

Hypothesis: <hypothesis>

<options>

Template 4, ...

Figure 4: Multiple instruction templates describing a natural language inference task.

Flan

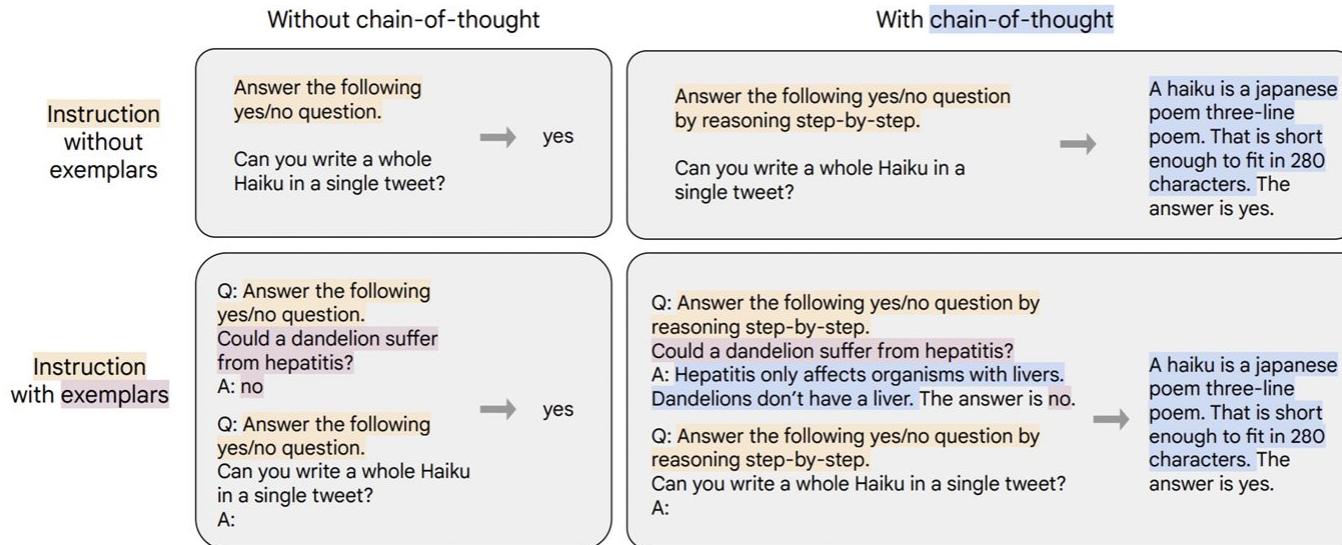


Figure 3: Combinations of finetuning data formats in this work. We finetune with and without exemplars, and also with and without chain-of-thought. In addition, we have some data formats without instructions but with few-shot exemplars only, like in Min et al. (2022) (not shown in the figure). Note that only nine chain-of-thought (CoT) datasets use the CoT formats.

Flan

- ▶ Instruction fine-tuning can be done on various models (PaLM, T5, etc.)
 - ▶ Flan-T5 models publicly available

Params	Model	Architecture	pre-training Objective	Pretrain FLOPs	Finetune FLOPs	% Finetune Compute
80M	Flan-T5-Small	encoder-decoder	span corruption	1.8E+20	2.9E+18	1.6%
250M	Flan-T5-Base	encoder-decoder	span corruption	6.6E+20	9.1E+18	1.4%
780M	Flan-T5-Large	encoder-decoder	span corruption	2.3E+21	2.4E+19	1.1%
3B	Flan-T5-XL	encoder-decoder	span corruption	9.0E+21	5.6E+19	0.6%
11B	Flan-T5-XXL	encoder-decoder	span corruption	3.3E+22	7.6E+19	0.2%
8B	Flan-PaLM	decoder-only	causal LM	3.7E+22	1.6E+20	0.4%
62B	Flan-PaLM	decoder-only	causal LM	2.9E+23	1.2E+21	0.4%
540B	Flan-PaLM	decoder-only	causal LM	2.5E+24	5.6E+21	0.2%
62B	Flan-cont-PaLM	decoder-only	causal LM	4.8E+23	1.8E+21	0.4%
540B	Flan-U-PaLM	decoder-only	prefix LM + span corruption	2.5E+23	5.6E+21	0.2%

Table 2: Across several models, instruction finetuning only costs a small amount of compute relative to pre-training. T5: Raffel et al. (2020). PaLM and cont-PaLM (also known as PaLM 62B at 1.3T tokens): Chowdhery et al. (2022). U-PaLM: Tay et al. (2022b).

Params	Model	Norm. avg.	MMLU		BBH		TyDiQA	MGSM
			Direct	CoT	Direct	CoT		
80M	T5-Small	-9.2	26.7	5.6	27.0	7.2	0.0	0.4
	Flan-T5-Small	-3.1 (+6.1)	28.7	12.1	29.1	19.2	1.1	0.2
250M	T5-Base	-5.1	25.7	14.5	27.8	14.6	0.0	0.5
	Flan-T5-Base	6.5 (+11.6)	35.9	33.7	31.3	27.9	4.1	0.4
780M	T5-Large	-5.0	25.1	15.0	27.7	16.1	0.0	0.3
	Flan-T5-Large	13.8 (+18.8)	45.1	40.5	37.5	31.5	12.3	0.7
3B	T5-XL	-4.1	25.7	14.5	27.4	19.2	0.0	0.8
	Flan-T5-XL	19.1 (+23.2)	52.4	45.5	41.0	35.2	16.6	1.9
11B	T5-XXL	-2.9	25.9	18.7	29.5	19.3	0.0	1.0
	Flan-T5-XXL	23.7 (+26.6)	55.1	48.6	45.3	41.4	19.0	4.9
8B	PaLM	6.4	24.3	24.1	30.8	30.1	25.0	3.4
	Flan-PaLM	21.9 (+15.5)	49.3	41.3	36.4	31.1	47.5	8.2
62B	PaLM	28.4	55.1	49.0	37.4	43.0	40.5	18.2
	Flan-PaLM	38.8 (+10.4)	59.6	56.9	47.5	44.9	58.7	28.5
540B	PaLM	49.1	71.3	62.9	49.1	63.7	52.9	45.9
	Flan-PaLM	58.4 (+9.3)	73.5	70.9	57.9	66.3	67.8	57.0
62B	cont-PaLM	38.1	61.2	57.6	41.7	53.1	45.7	32.0
	Flan-cont-PaLM	46.7 (+8.6)	66.1	62.0	51.0	53.3	62.7	40.3
540B	U-PaLM	50.2	71.5	64.0	49.2	62.4	54.6	49.9
	Flan-U-PaLM	59.1 (+8.9)	74.1	69.8	59.3	64.9	68.3	60.4

- The evaluation metric on all four benchmark suites is few-shot prompted accuracy (exact match)
- These evaluation benchmarks are held-out (not included in the finetuning data)

Table 5: Instruction finetuning (Flan) improves performance on top of other continued pre-training methods.

Human Generated Datasets

Name	Annotators	Format	Size
<i>P3</i> (Sanh et al., 2022)	NLP researchers	prompts	2,000+ prompts; 170 datasets
<i>BIG-bench</i> (Srivastava et al., 2022)	442 authors across 132 institutions		204 tasks
<i>Dolly</i> (Conover et al., 2023)	Databricks employees	(instruction, input*, output)	15K instances
<i>Natural Instruction</i>	V1 (Mishra et al., 2022)	(instruction, {(input, output)})	61 tasks
	V2 (Wang et al., 2022)		1,616 tasks
<i>Multi-Instruct</i> (Xu et al., 2023)			62 multimodal tasks; 5 instructions per task

Dataset: Dolly

- Dolly is an open source follow-up to InstructGPT of 15k instruction fine-tuning examples
- Employees of DataBricks wrote all the prompt / response pairs

We set up a contest, where the top 20 labelers would get a big award. We also outlined 7 very specific tasks:

- Open Q&A: For instance, "Why do people like comedy movies?" or "What is the capital of France?" In some cases, there's not a correct answer, and in others, it requires drawing on knowledge of the world at large.
- Closed Q&A: These are questions that can be answered using only the information contained in a passage of reference text. For instance, given a paragraph from Wikipedia on the atom, one might ask, "What is the ratio between protons and neutrons in the nucleus?"
- Extract information from Wikipedia: Here an annotator would copy a paragraph from Wikipedia and extract entities or other factual information such as weights or measurements from the passage.
- Summarize information from Wikipedia: For this, annotators provided a passage from Wikipedia and were asked to distill it to a short summary.
- Brainstorming: This task asked for open-ended ideation and an associated list of possible options. For instance, "What are some fun activities I can do with my friends this weekend?".
- Classification: For this task, annotators were asked to make judgments about class membership (e.g. are the items in a list animals, minerals or vegetables) or to judge the properties of a short passage of text, such as the sentiment of a movie review.
- Creative writing: This task would include things like writing a poem or a love letter.

Dataset: Dolly

Category	Instruction	Context	Response
Creative Writing	What should I do on a free afternoon in San Francisco?		You can visit and take photos at San Francisco's landmarks such as Pier 39 or Golden Gate Bridge. If you enjoy walking outdoors, there are several large parks in the city. You can visit Golden Gate Park and the various attractions and museums located in the park. The Presidio is also a popular park where you can get some of the best views of Golden Gate Bridge.

Dataset: Dolly

Category	Instruction	Context	Response
Creative Writing	Write a haiku about how much I love pickles.		Caressed by sweet brine, you are crispy and crunchy. Oh, won't you be mine?

Dataset: Dolly

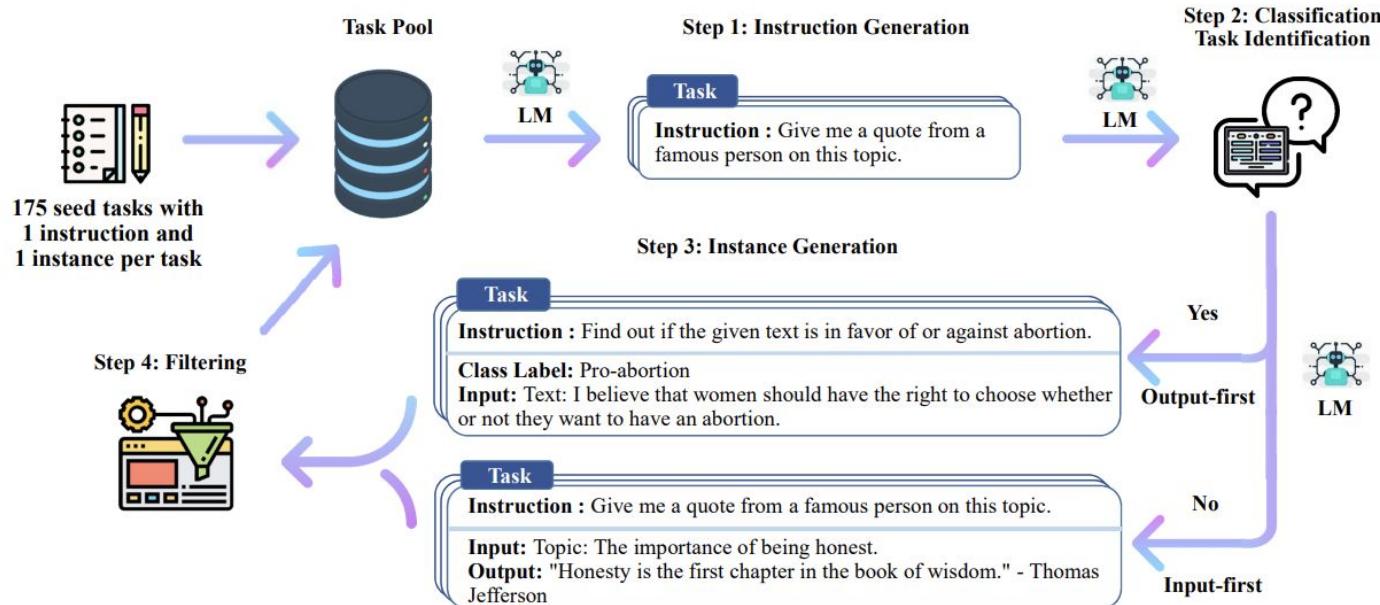
Category	Instruction	Context	Response
Information Extraction	Extract all of the dates mentioned in this paragraph and list them using bullets in the format {Date} - {Description}	Daniel Leavitt was born November 16, 1813, at Rye, New Hampshire, the son of Benning Leavitt, influential businessman, state senator, county commissioner and later Chicopee Selectman, and his wife Olive (Jenness) Leavitt. Daniel Leavitt married in 1838 at West Springfield, Massachusetts, Ruth Jeannette Ball. They had three children.	<ul style="list-style-type: none">• November 16, 1813 - This is when Daniel Leavitt was born• 1838 - This is when Daniel Leavitt married his wife Ruth Jeannette Ball.

Name	Annotators	Format	Size
PromptSource (Bach et al., 2022)	11,500+ prompts; 170 datasets	JSON	2,000+ prompts; 170 datasets
BIG-bench (Srivastava et al., 2023)	11,500+ prompts; 170 datasets	JSON	204 tasks
Dolly (Conover et al., 2022)	11,500+ prompts; 170 datasets	JSON	15K instances
Natural Instruction	V1 (Mishra et al., 2022) V2 (Wang et al., 2023)	JSON	61 tasks
Multi-Instruct (Xu et al., 2023)	11,500+ prompts; 170 datasets	JSON	1,616 tasks
		JSON	62 multimodal tasks; 5 instructions per task

Limitations:

- ❑ Too much human effort
- ❑ Hard to increase the task diversity and complexity

Self-Instruct: Aligning Model with Self Generated Instructions



SELF-INSTRUCT: Aligning Language Model with Self Generated Instructions. ACL 23

Yizhong Wang, Yeganeh Kordi, **Swaroop Mishra**, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi.

Self-instruct: Step-1: Generating Instructions

- Initiate the task pool with 175 tasks (1 instruction and 1 instance for each task)
- For every step, sample 8 task instructions from this pool as in-context examples.
- Of the 8 instructions, 6 are from the human-written tasks, and 2 are from the model-generated tasks in previous steps to promote diversity

Come up with a series of tasks:

```
Task 1: {instruction for existing task 1}
Task 2: {instruction for existing task 2}
Task 3: {instruction for existing task 3}
Task 4: {instruction for existing task 4}
Task 5: {instruction for existing task 5}
Task 6: {instruction for existing task 6}
Task 7: {instruction for existing task 7}
Task 8: {instruction for existing task 8}
Task 9:
```

Prompt Template

Self-instruct: Step-2: Classification Task Identification

- The authors regard tasks that have a small limited output label space as classification tasks. Need two different approaches for classification and non-classification tasks
- Prompt the LM in a few-shot way to determine this, using 12 classification instructions and 19 non-classification instructions from the seed tasks.

Can the following task be regarded as a classification task with finite output labels?

Task: Given my personality and the job, tell me if I would be suitable.
Is it classification? Yes

Task: Give me an example of a time when you had to use your sense of humor.
Is it classification? No

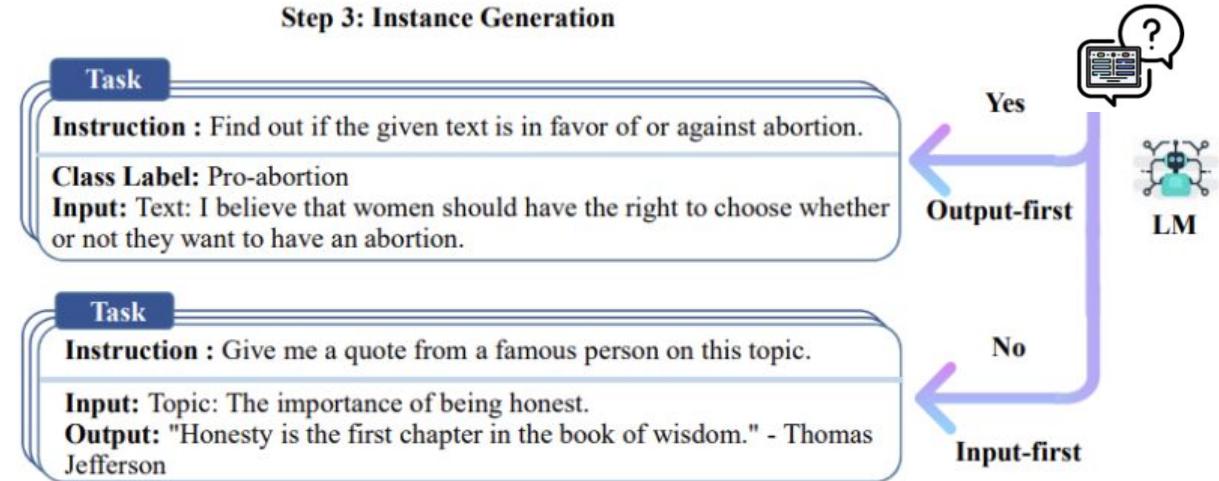
Task: Replace the placeholders in the given text with appropriate named entities.
Is it classification? No

Prompt Template

Self-instruct: Step-3: Instance Generation

- Given the instructions and their task type, generate instances for each instruction independently.
- Challenging because it requires the model to understand what the target task is, based on the instruction, figure out what additional input fields are needed and generate them, and finally complete the task by producing the output.
- Pretrained LMs can achieve this to a large extent when prompted with instruction-input-output in-context examples from other tasks.

Step 2: Classification
Task Identification



Why two modes? Input-based approach can generate inputs biased toward one label, especially for classification tasks (e.g., for grammar error detection, it usually generates grammatical input).

Prompt-templates: Input First

Come up with examples for the following tasks. Try to generate multiple examples when possible. If the task doesn't require additional input, you can generate the output directly.

Task: Which exercises are best for reducing belly fat at home?

Output:

- Lying Leg Raises
- Leg In And Out
- Plank
- Side Plank
- Sit-ups

Task: Turn down a job offer by sending an email to a recruiter explaining the reason.

Output: Hi [Recruiter],

Thank you so much for the generous offer to join your team. As we discussed, I've admired the company for a number of years, and am a proud endorser of its products. However, after further consideration of where I currently am in my career, I've decided to accept an offer at another company.

I would love to stay in touch with you and have already started following you on [Social Media Platform]. Again, thank you so much for your time and consideration.

Thanks again,
[Your Name]

Task: {Instruction for the target task}

Prompt-templates: Output First

Given the classification task definition and the class labels, generate an input that corresponds to each of the class labels. If the task doesn't require input, just generate the correct class label.

Task: Classify the sentiment of the sentence into positive, negative, or mixed.

Class label: mixed

Sentence: I enjoy the flavor of the restaurant but their service is too slow.

Class label: Positive

Sentence: I had a great day today. The weather was beautiful and I spent time with friends.

Class label: Negative

Sentence: I was really disappointed by the latest superhero movie. I would not recommend it.

...

Task: Which of the following is not an input type? (a) number (b) date (c) phone number (d) email address (e) all of these are valid inputs.

Class label: (e)

Task: {instruction for the target task}

Self-instruct: Step-4: Filtering

To encourage diversity, a new instruction is added to the task pool only when its ROUGE-L similarity with any existing instruction is less than 0.7.

Also exclude instructions that contain some specific keywords (e.g., image, picture, graph) that usually can not be processed by LMs.

When generating new instances for each instruction, we filter out instances that are exactly the same or those with the same input but different outputs.

Invalid generations are identified and filtered out based on heuristics (e.g., instruction is too long or too short, instance output is a repetition of the input).

Self-Instruct: Some example generations

Instruction: Given an address and city, come up with the zip code.

Input:

Address: 123 Main Street, City: San Francisco 

Output: 94105

Instruction: How to write a code for converting degrees fahrenheit to celsius.

Input: Null

Output:

```
def convert_fahrenheit_to_celsius(fahr):  
    celsius = (fahr - 32) * 5 / 9  
    return celsius
```



Instruction: Write a letter from the perspective of a cat.

Input: Null

Output:

Dear [Owner], I am writing to you today because I have a problem. You see, I've been living with you for the past [Number of Years] and I love it here. The food is great, the bed is comfy, and there are plenty of toys to play with. But lately, I've noticed that you're not as affectionate towards me as you used to be. In fact, sometimes when I try to cuddle up next to you on the couch, you push me away ...

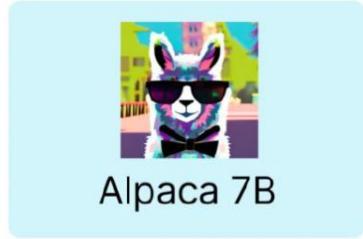
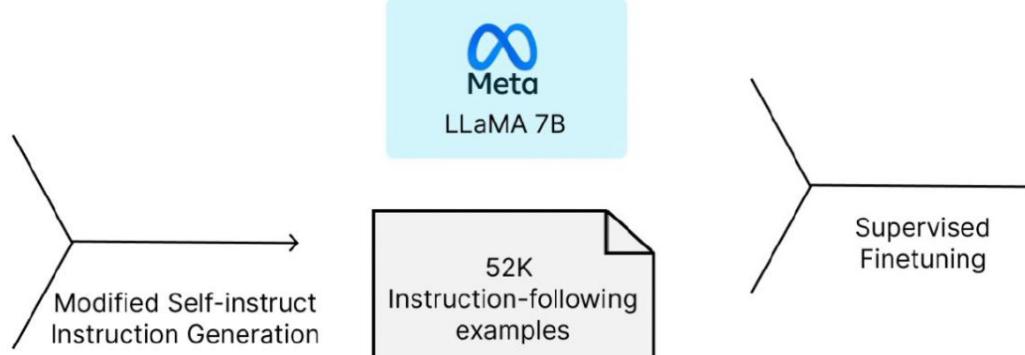


Self-Instruct: Main Findings

statistic

# of instructions	52,445
- # of classification instructions	11,584
- # of non-classification instructions	40,861
# of instances	82,439
- # of instances with empty input	35,878
ave. instruction length (in words)	15.9
ave. non-empty input length (in words)	12.7
ave. output length (in words)	18.9

Model	# Params	ROUGE-L
Vanilla LMs		
T5-LM	11B	25.7
GPT3	175B	6.8
Instruction-tuned w/o SUPERNI		
① T0	11B	33.1
GPT3 + T0 Training	175B	37.9
② GPT3 _{SELF-INST} (Ours)	175B	39.9
InstructGPT ₀₀₁	175B	40.8
Instruction-tuned w/ SUPERNI		
③ Tk-INSTRUCT	11B	46.0
GPT3 + SUPERNI Training	175B	49.5
GPT3 _{SELF-INST} + SUPERNI Training (Ours)	175B	51.6



Example seed task

Instruction: Brainstorm a list of possible New Year's resolutions.

Output:

- Lose weight
- Exercise more
- Eat healthier

Example Generated task

Instruction: Brainstorm creative ideas for designing a conference room.

Output:

... incorporating flexible components, such as moveable walls and furniture ...

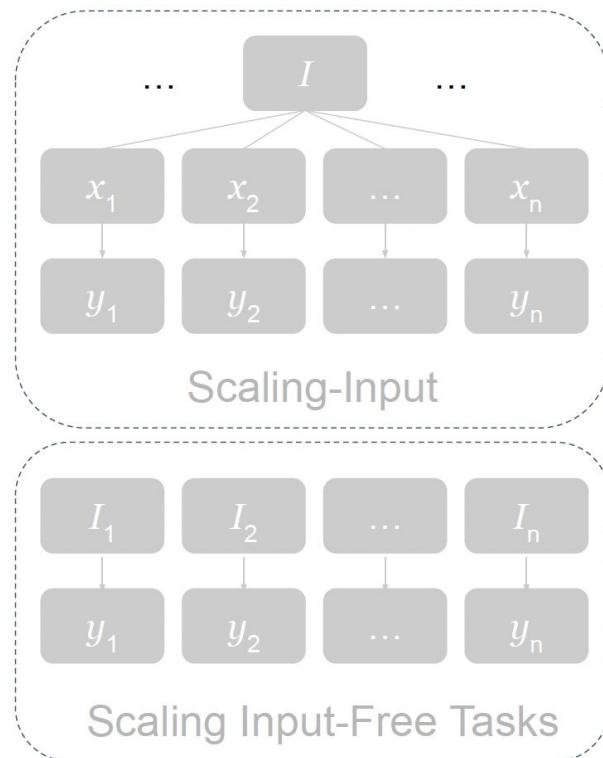
[Alpaca: A Strong, Replicable Instruction-Following Model \(Stanford, 2023\)](#)

Release	Collection	Model	Model Details			Data Collection & Training Details			
			Base	Size	Public?	Prompt Types	Tasks in Flan	# Exs	Methods
2020 05	UnifiedQA	UnifiedQA	RoBERTa	110-340M	P	ZS	46 / 46	750k	
2021 04	CrossFit	BART-CrossFit	BART	140M	NP	FS	115 / 159	71M	
2021 04	Natural Inst v1.0	Gen. BART	BART	140M	NP	ZS / FS	61 / 61	620k	+ Detailed k-shot Prompts
2021 09	Flan 2021	Flan-LaMDA	LaMDA	137B	NP	ZS / FS	62 / 62	4.4M	+ Template Variety
2021 10	P3	T0, T0+, T0++	T5-LM	3-11B	P	ZS	62 / 62	12M	+ Template Variety + Input Inversion
2021 10	MetalCL	MetalCL	GPT-2	770M	P	FS	100 / 142	3.5M	+ Input Inversion + Noisy Channel Opt
2021 11	ExMix	ExT5	T5	220M-11B	NP	ZS	72 / 107	500k	+ With Pretraining
2022 04	Super-Natural Inst.	Tk-Instruct	T5-LM, mT5	11-13B	P	ZS / FS	1556 / 1613	5M	+ Detailed k-shot Prompts + Multilingual
2022 10	GLM	GLM-130B	GLM	130B	P	FS	65 / 77	12M	+ With Pretraining + Bilingual (en, zh-cn)
2022 11	xP3	BLOOMz, mT0	BLOOM, mT5	13-176B	P	ZS	53 / 71	81M	+ Massively Multilingual
2022 12	Unnatural Inst. [†]	T5-LM-Unnat. Inst.	T5-LM	11B	NP	ZS	~20 / 117	64k	+ Synthetic Data
2022 12	Self-Instruct [†]	GPT-3 Self Inst.	GPT-3	175B	NP	ZS	Unknown	82k	+ Synthetic Data + Knowledge Distillation
2022 12	OPT-IML Bench [†]	OPT-IML	OPT	30-175B	P	ZS + FS CoT	~2067 / 2207	18M	+ Template Variety + Input Inversion + Multilingual
2022 10	Flan 2022 (ours)	Flan-T5, Flan-PaLM	T5-LM, PaLM	10M-540B	P NP	ZS + FS CoT	1836	15M	+ Template Variety + Input Inversion + Multilingual

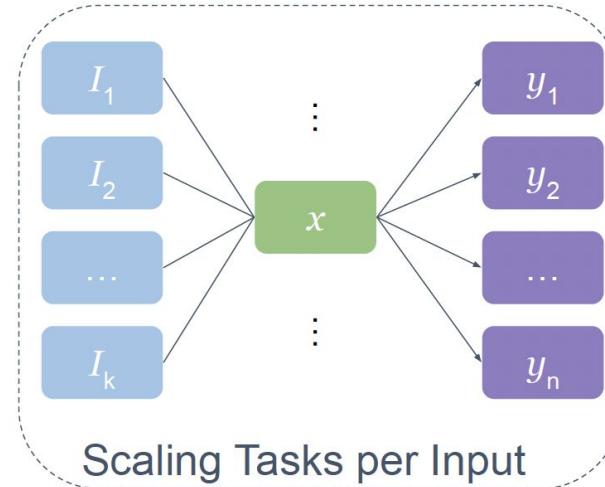
More recent LLM generated datasets

Dataset	Motivation	#instances
Self-Instruct (Wang et al., 2023)		82K
Unnatural Instruction (Honovich et al., 2023)	increase task diversity	240K
Muffin (Lou et al., 2023)	per input	68K
Dynosaurus (Yin et al., 2023)	decrease cost	800K
WizardLM (Xu et al., 2023)	increase task complexity	250K
LongForm (Köksal et al., 2023)	enhance long text generation	27K

Scaling Strategies



- ❑ I : Instruction
- ❑ x : input
- ❑ y : output



Our experiments with FLAN-T5 for Finance

Extreme Financial Numeric Labelling

UNITED STATES SECURITIES AND EXCHANGE COMMISSION
Washington, D.C. 20549

Form 10-K

ANNUAL REPORT PURSUANT TO SECTION 13 OR 15(d) OF
THE SECURITIES EXCHANGE ACT OF 1934

For the fiscal year ended December 31, 2021

Commission File Number: 001-14965

The Goldman Sachs Group, Inc.

(Exact name of registrant as specified in its charter)

Delaware
(State or other jurisdiction of
incorporation or organization)

(I.R.S. Employer
Identification No.)

The firm has other retained interests not reflected in the table above with a fair value of \$360 million and a weighted average life of 3.6 years as of December 2021, and a fair value of \$192 million and a weighted average life of 3.9 years as of December 2020. Due to the nature and fair value of certain of these retained interests, the weighted average assumptions for constant prepayment and discount rates and the related sensitivity to adverse changes are not meaningful as of both December 2021 and December 2020. The firm's maximum exposure to adverse changes in the value of these interests is the carrying value of \$360 million as of December 2021 and \$192 million as of December 2020.

Attributes



Retained Interest, Fair Value Disclosure

Tag	us-gaap:RetainedInterestFairValue Disclosure
Fact	192,000,000
Period	As of 12/31/2020
Axis	SRT Consolidated Entities Axis
Member	GS Other Retained Interests Mem



Problem Statement

Input = "Upon adoption, the Company recorded **\$2.7** billion of net deferred tax assets, reduced other non-current assets by **\$128** million, and increased retained earnings by **\$2.6** billion on its Consolidated Balance Sheet."



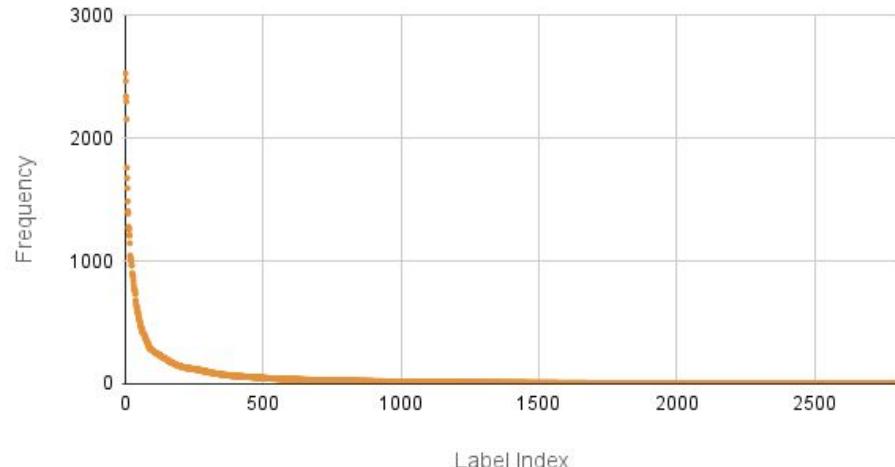
Output = {
2.7 : Deferred Tax Assets Liabilities Net,
128 : Other Assets Noncurrent,
2.6 : Others
}

Data Sources

- We collected 10-K annual documents publicly available on SEC from 2019-2021.
- The dataset is created from 10-K documents of 2348 companies giving us 79088 sentences containing 142022 annotated numericals with a label set of 2794.

	#Sentences	#Companies	#Data Points	#Labels
Train	62782	798	111493	2692
Dev	6823	756	13191	1273
Test	9483	794	18238	1374
Total	79088	2348	142922	2794

Frequency distribution of the labels



As can be seen from the long tail, our dataset contains many labels where data points are less than 20.

Why this is a challenging problem?

Example 1

On March 7, 2019, the Company completed the Park MGM Transaction and amended the MGM-MGP Master Lease concurrent with which the Company paid \$637.5 million, of which \$605.6 million was cash and the remainder was the issuance of approximately 1.0 million of Operating Partnership units, to a subsidiary of MGM and, as a result, the annual rent payment to the Company increased by \$50 million, prorated for the remainder of the lease year.

Example 2

As of December 31, 2018, AVANGRID share capital consisted of 500,000,000 shares of common stock authorized, 309,752,140 shares issued and 309,005,272 shares outstanding, 81.5% of which were owned by Iberdrola, each having a par value of \$0.01, for a total value of common stock capital of \$3 million and additional paid in of \$13,657 million.

Example 3

The company originally acquired 75 % of wind te holdco on november 3, 2015 drop down assets. which were consolidated with 25 % of the net assets recorded as noncontrolling interest.

Dataset	Domain	Entity Types
conll-2003	Generic	4
ontonotes-v5	Generic	18
ace-2005	Generic	7
genia	Biomedical	36
Francis et al. (2019)	Financial	9
finer-139	Financial	139
FNXL	Financial	2794

Limited Context

The sentences sometimes **do not contain enough context** to understand and assign a label to different numericals. This is especially difficult for sentences containing multiple numericals with different tags.

Eg: “Total reserves net of the discount were \$2.48 billion and **\$2.45** billion, and the related amount of discount was \$1.16 billion and \$1.16 billion, at December 31, 2019 and 2018, respectively.”

2.45 : Workers Compensation Liability Current And Noncurrent

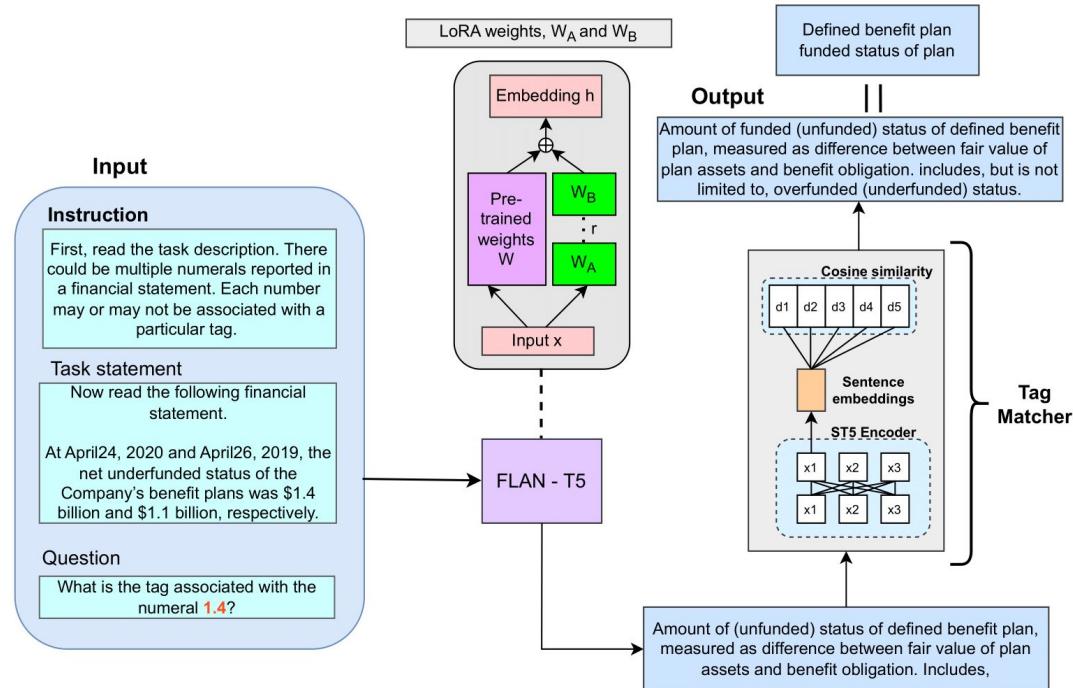
Motivation to use Tag Documentation

- Domain experts gained knowledge from tag documentations
- We observe that the two tags differ by a single word only whereas their corresponding documentations vary significantly

Tag	Documentation
common stocks shares issued	Total number of common shares of an entity that have been sold or granted to shareholders (includes common shares that were issued, repurchased and remain in the treasury). These shares represent capital invested by the firm's shareholders and owners, and may be all or only a portion of the number of shares authorized. Shares issued include shares outstanding and shares held in the treasury.
common stock shares authorized	The maximum number of common shares permitted to be issued by an entity's charter and bylaws.
tax credit carry forward amount	The amount of the tax credit carryforward, before tax effects, available to reduce future taxable income under enacted tax laws.
tax credit carry forward valuation allowance	Amount of valuation allowance pertaining to the deferred tax asset representing potential future taxable deductions from tax credit carryforwards for which it is more likely than not that a tax benefit will not be realized.
due to affiliate noncurrent	Amount of receivables owed to an entity that is affiliated with the reporting entity by means of direct or indirect ownership, which are usually due after one year (or one business cycle, if longer).
due to affiliate current and non current	Amount of payable due to an entity that is affiliated with the reporting entity by means of direct or indirect ownership.
notes payable related parties classified current	The amount for notes payable (written promise to pay), due to related parties. Used to reflect the current portion of the liabilities (due within one year or within the normal operating cycle if longer).
notes payable related parties current and non current	The amount for notes payable (written promise to pay), due to related parties.

Proposed Methodology

- Supervised Generative Phase, and an unsupervised Tag Matching Phase
- Instruction tune FLAN-T5 with carefully-curated task-specific instruction

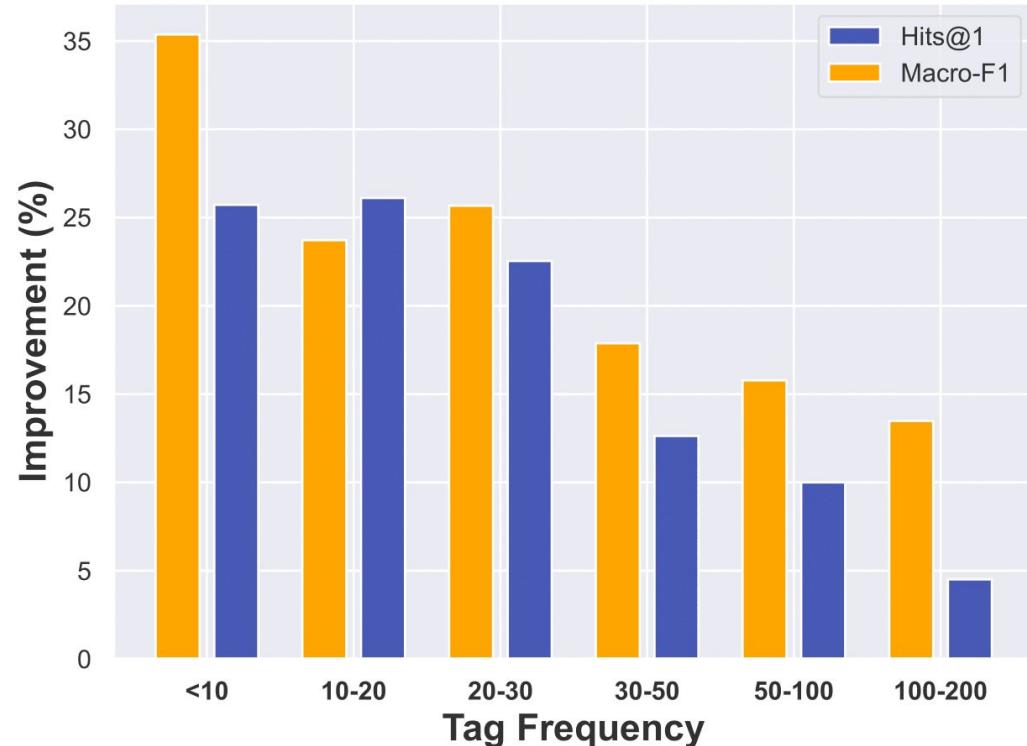


Main Results

Model	Dataset							
	FNXL(Sharma et al., 2023)				FiNER (Loukas et al., 2022)			
	M-P	M-R	M-F1	Hits@1	M-P	M-R	M-F1	Hits@1
FiNER (bert-base)	49.17	<u>49.71</u>	47.13	75.34	72.60	81.10	76.61	81.50
FiNER (sec-base)	47.76	<u>48.87</u>	46.20	74.67	81.11	<u>83.20</u>	82.14	82.30
Label Semantics	46.35	<u>45.12</u>	45.72	71.25	71.50	<u>80.15</u>	75.57	80.25
GalaXC	46.91	<u>44.81</u>	45.81	72.97	72.20	<u>80.95</u>	76.32	81.10
AttentionXML Pipeline	<u>50.69</u>	<u>48.51</u>	<u>47.54</u>	<u>76.76</u>	<u>82.15</u>	82.30	<u>82.22</u>	<u>83.25</u>
ChatGPT (500 samples)	11.13	7.68	9.08	19.6	20.12	15.67	17.61	22.35
T5-Base	59.94	49.48	54.21	79.21	86.92	84.35	85.61	83.45
T5-Large	61.87	58.46	60.11	83.26	88.12	85.10	86.58	84.12
FLAN-T5-Large	66.21	65.34	65.77	86.21	92.10	96.35	94.17	85.89
FLAN-T5-Large with Prefix-Tuning	65.10	64.21	64.65	85.69	90.18	94.35	92.21	85.12
FLAN-T5-Large with LORA	65.14	67.36	66.23	89.98	91.84	97.85	94.74	86.03

Performance on least frequently occurring labels

- We observe our **FLAN-FinXC** performs substantially better than AttentionXML for the rare labels.
- Notably, even for tags that appear fewer than 10 times in the training data, our model achieve improvement of 25% in Hits@1 over the closest baseline.



Ablation Study

- Understanding the significance of different modules
- Demonstrates the significance of **instruction prompt** and **tag metadata**

Model	Macro-F1	Hits@1
FLAN-T5-Large with LoRA	66.23	89.98
w/ S-BERT-L12 as Tag Matcher	63.11	88.13
w/ S-BERT-L6 as Tag Matcher	62.87	87.72
w/o instruction prompt	56.46	76.55
w/o tag metadata	53.12	73.14

Zero Shot Capability

- Generate tags/labels for which it has not been explicitly trained
- We observe that our best model achieves a Macro-F1 of **58.89** over these 67 unseen labels

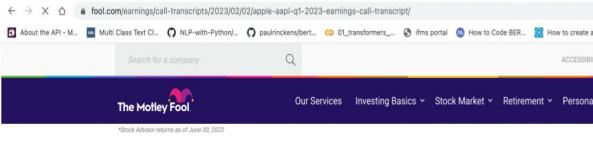
Tag	F1-score
foreign currency transaction gain before tax	0.85
commercial paper at carrying value	0.85
accounts payable other current	0.80
recognition of deferred revenue	0.76
available for sale debt securities gross unrealized gain	0.66

Instruction-Guided Bullet Point Summarization of Long Financial Earnings Call Transcripts

Instruction-Guided Bullet Point Summarization of Long Financial Earnings Call Transcripts. SIGIR'24 (short).
Khatuya, Subhendu, Sinha, Koushiki, Ganguly, Niloy, Ghosh, Saptarshi and Goyal, Pawan.

Problem Statement

Develop an efficient framework for telegram-style bullet point summarization of long financial earnings call transcript documents.



- QUARTERLY EARNINGS PER SHARE \$1.52.
- QUARTERLY TOTAL NET SALES \$97.28 BILLION VERSUS \$89.58 BILLION REPORTED LAST YEAR.
- BOARD OF DIRECTORS AUTHORIZED AN INCREASE OF \$90 BILLION TO THE EXISTING SHARE REPURCHASE PROGRAM.
- QUARTERLY IPHONE REVENUE \$50.57 BILLION VERSUS \$47.94 BILLION REPORTED LAST YEAR.
- Q2 EARNINGS PER SHARE VIEW \$1.43, REVENUE VIEW \$93.89 BILLION -- REFINITIV IBES DATA.

Dataset and Challenges

We used the **ECTSum*** dataset for this work.

Total number of document-summary pairs	2,425
Average document length	2.9K
Average length of gold truth summary	50
Document-summary compression ratio	103.67

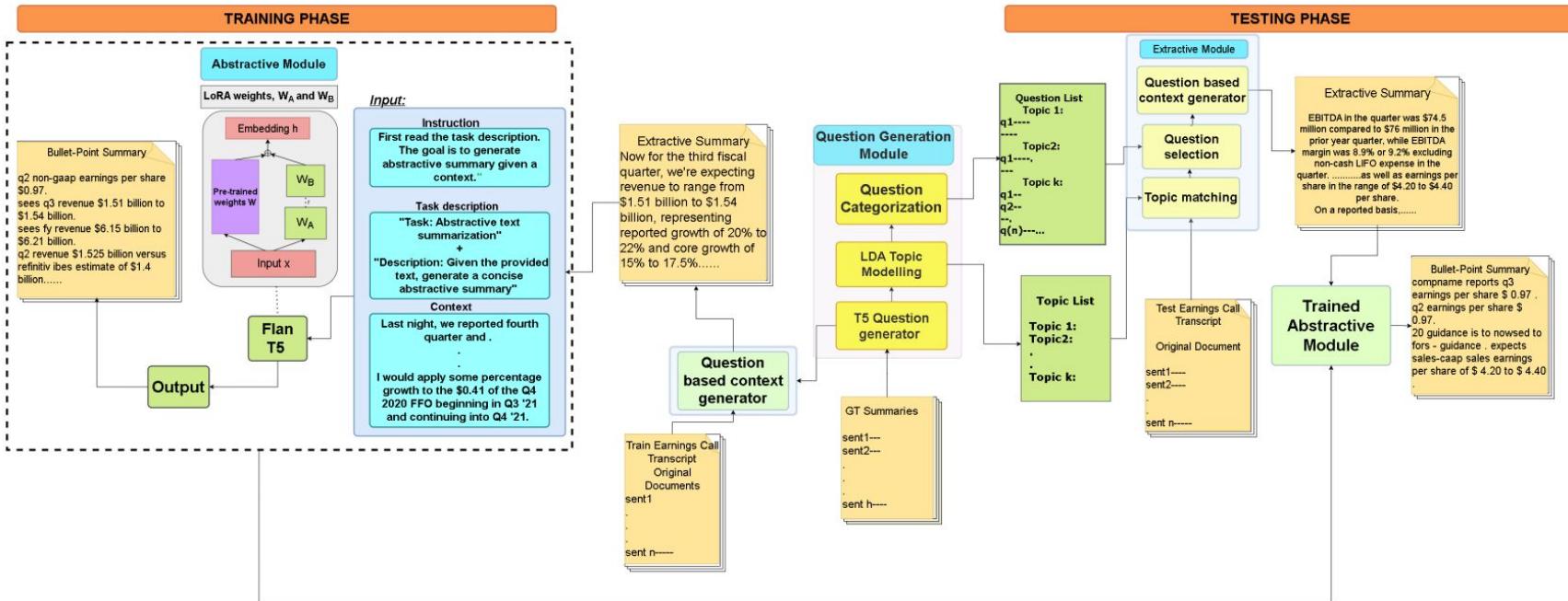
Dataset	# Docs.	Coverage	Density	Comp. Ratio	# Tokens	
					Doc.	Summary
ARXIV/PUBMED (Cohen et al., 2018)*	346,187	0.87	3.94	31.17	5179.22	257.44
BILLSUM (Kornilova and Eidelman, 2019) [†]	23,455	-	4.12	13.64	1813.0	207.7
BIGPATENT (Sharma et al., 2019)*	1,341,362	0.86	2.38	36.84	3629.04	116.67
GovREPORT (Huang et al., 2021) [†]	19,466	-	7.60	19.01	9409.4	553.4
BOOKSUM Chapters (Kryściński et al., 2021)*	12,293	0.78	1.69	15.97	5101.88	505.32
ECTSum	2,425	0.85	2.43	103.67	2916.44	49.23

Ect-sum: A new benchmark dataset for bullet point summarization of long earnings call transcripts. **EMNLP 2022**.

Rajdeep Mukherjee, Abhinav Bohra, Akash Banerjee, Soumya Sharma, Manjunath Hegde, Afreen Shaikh, Shivani Shrivastava, Koustuv Dasgupta, Niloy Ganguly, Saptarshi Ghosh, Pawan Goyal.

Proposed Framework

1. Proposed framework named FLAN-FinBPS integrates unsupervised and supervised methods for a comprehensive bullet point summarization.
2. With the employment of an unsupervised method and parameter-efficient instruction tuned Flan-T5 based generative method, the goal was to cut down on training time and improve factual and numeric consistency.



Methodology

Extractive phase

- Generate questions for every line of each ground truth summary in the train set.
- To get the extractive summary for training the second phase, we take the top ‘n’ matched sentences for each question

<u>Reference summary</u>	<u>Generated questions</u>
q2 non-gaap earnings per share \$0.97.	What is q2 non-gaap earnings per share?
sees q3 revenue \$1.51 billion to \$1.54 billion.	What is the q3 revenue?
sees fy revenue \$6.15 billion to \$6.21 billion.	What is the revenue of fy revenue?

Categorization

We compile all the questions and categorize them into top 30 themes.



Topic	Sample Questions
revenue	How much did the qtrly railway operating revenues increase? How much is the quarter year revenue expected to be?
sales	How much did q4 sales fall? How much did qtrly report net sales?
income	What is the net investment income of qtrly? What is the qtrly net interest income before provision for credit losses?

Abstractive Phase

Abstractive phase

- Using the extractive context from the first stage, FLAN-T5 model is carefully fine-tuned with task-specific instructions with the bullet point summary as the target.

During testing

- We determine the possible topic present in a test set ECT by detecting the presence of topic keywords.
- We find top matched questions for each ECT using cosine similarity and generate contexts for the question list.
- We pass this extractive context to the FLAN-T5 model to get the final bullet point summaries.

Main Results

Model	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	Num-Prec.	SummaC _{conv}
Unsupervised						
LexRank (Erkan and Radev, 2004)	0.122	0.023	0.154	0.638	1.00	1.00
DSDR (He et al., 2012)	0.164	0.042	0.200	0.662	1.00	1.00
PacSum (Zheng and Lapata, 2019)	0.167	0.046	0.205	0.663	1.00	1.00
Extractive						
SummaRuNNer (Nallapati et al., 2017)	0.273	0.107	0.309	0.647	1.00	1.00
BertSumExt (Zheng and Lapata, 2019)	0.307	0.118	0.324	0.667	1.00	1.00
MatchSum (Zhong et al., 2020)	0.314	0.126	0.335	0.679	1.00	1.00
Abstractive						
BART (Lewis et al., 2019)	0.327	0.153	0.361	0.692	0.594	0.431
Pegasus (Zhang et al., 2020)	0.334	0.185	0.375	0.708	0.783	0.444
T5 (Raffel et al., 2020)	0.363	0.209	0.413	0.728	0.796	0.508
Long Document Summarizers						
BigBird (Zaheer et al., 2020)	0.344	0.252	0.400	0.716	0.844	0.452
LongT5 (Guo et al., 2021)	0.438	0.267	0.471	0.732	0.812	0.516
LED (Beltagy et al., 2020)	0.450	0.271	0.498	0.737	0.679	0.439
Bullet Point Summarizers						
ECTBPS (Mukherjee et al., 2022)	0.467	0.307	0.514	0.764	0.916	0.518
Flan-FinBPS (Ours)	0.557	0.376	0.529	0.889	0.939	0.532
RI(%)	+19.27%	+22.47%	+2.91%	+16.36%	+2.51%	+2.70%

Connetion with the next topic

Prompting for instruction fine-tuned models

Prompt

Llama-2-70B

Story: Jason listened to the weather and heard it was goingto be sunny. He thought the kids might like to goswimming. He gathered up the swimsuits, towelsand sunscreen. Jason and the kids got into the truckand drove to the beach. They spent the next 2 hoursplaying and splashing in the surf.

One-sentence Summary:

Model Output

Jason took the kids swimming at the beach after hearing the weather forecast, gathering necessary items and driving there.

Story text from <https://arxiv.org/pdf/1910.00998.pdf>

Llama-2-7B Chat

Story: Jason listened to the weather and heard it was goingto be sunny. He thought the kids might like to goswimming. He gathered up the swimsuits, towelsand sunscreen. Jason and the kids got into the truckand drove to the beach. They spent the next 2 hoursplaying and splashing in the surf.

One-sentence Summary:

Sure! Here is a one-sentence summary of the story:
Jason and the kids went on a fun-filled day at the beach, playing and splashing in the surf after gathering all the necessary items, including swimsuits, towels, and sunscreen.

Very crucial step towards building chat agents

- Motivation:
 - Suppose you want to build a chat agent
 - LLMs are trained to reduce the perplexity of a large training corpus containing web text, articles, code, etc. (i.e. it's good at completing your _____.)
 - But a chat agent should not merely predict what comes next, it should behave conversationally and know when to stop
 - We want to *align* the LLM with the expectations of a human user for a given task in some *instruction*
- Key idea:
 - Build a “chat agent” training dataset
 - Fine-tune the LLM on this data
- This technique goes by many names...
 - instruction fine-tuning
 - chat fine-tuning
 - alignment
 - behavioral fine-tuning

References

Part of the slides are adapted from

EMNLP'23 Tutorial: [LLM-driven Instruction Following: Progresses and Concerns](#)

Swaroop Mishra's slides: [Towards Development of Models that Learn New Tasks from Instructions](#)

<https://www.cs.cmu.edu/~mgormley/courses/10423//slides/lecture12-rlhf-ink.pdf>

https://aritter.github.io/CS-4650-au22/slides/lec14-bart_qpt3.pdf