Cross-Task Generalization via Natural Language Instructions

Daniel Khashabi
Allen Institute for AI

Joint w/ Swaroop Mishra and Hanna Hajishirzi
Task-Specific Models

- Task-specific models do not generalize across tasks.
- There are MANY tasks!

Text: She chose to make a salad for lunch tomorrow and Sunday.
Question: how long did it take for her to make a salad?

“event duration”

“30mins”, “an hour”

“How often …”

Question Typing
Question Answering
Sentence Modification
Text: She chose to make a salad for lunch tomorrow and Sunday. 
Question: how long did it take for her to make a salad?

- "event duration"
- "30mins", "an hour"
- "how often ..."
Beyond Task-Specific Models

Text: She chose to make a salad for lunch tomorrow and Sunday.
Question: how long did it take for her to make a salad?

- Question-typing
- Question-answering
- Sentence-modification

“event duration”
“30mins”, “an hour”
“how often ...”

Do not generalize to “unseen” tasks.

[Raffel et al. 2020]
Beyond Task-Specific Models

• Instruction **define** tasks explicitly in **natural language**.

**Text:** She chose to make a salad for lunch tomorrow and Sunday. **Question:** how long did it take for her to make a salad?

**In this task we ask you to write answer to the given question.**

**Indicate the type of temporal phenomenon in the question ...**

**Label a question that is free of any grammatical/logical errors w/ ‘yes’ ...**

“event duration”

“30mins”, “an hour”

“how often ...”
She chose to make a salad for lunch tomorrow and Sunday.

Question: how long did it take for her to make a salad?
Instructions Paradigm: Challenges

1. There is no benchmarks containing natural language instructions for a diverse range of tasks.

   We present a dataset of **natural instructions** for a **wide variety** of tasks!

2. Unclear whether models benefit from task “instructions”.

   We show empirical evidence of their **benefits**!
Natural-Instructions: Overview

• Natural Instructions:
  • 61 tasks and instructions
  • 160k instances (input -> outputs)

**Input:** She chose to make a salad for lunch tomorrow and Sunday.

**Instructions: generating “duration” questions task**

*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. ...*

**Output:** “how long did it take for her to make a salad?”
Use existing datasets and the instructions used to crowdsourc them

(Efrat & Levy, 2020)

A typical data construction pipeline
Natural-Instructions: Construction (1)

- Contacted dataset authors to access their crowdsourcing instructions and the associated annotations
  1. CosmosQA [Huang et al. 2019]
  2. DROP [Dua et al. 2019]
  3. Essential-Terms [Khashabi et al. 2017]
  4. MCTACO [Zhou et al. 2019]
  5. MultiRC [Khashabi et al. 2018]
  6. QASC [Khot et al. 2020]
  7. Quoref [Dasigi et al. 2019]
  8. ROPES [Lin et al. 2019]
  9. Winogrande [Sakaguchi et al. 2020]

collecting existing datasets

- dividing crowdsourcing instructions into minimal tasks

instructions schema

- mapping crowdsourcing instructions to the schema
Crowdsourcing instructions tend to involve multiple annotation steps.
Split them to self-contained tasks.

- **MC-TACO** [Zhou et al. 2019]
  - collecting existing datasets
  - dividing crowdsourcing instructions into minimal tasks
  - instructions schema
  - mapping crowdsourcing instructions to the schema
Crowdsourcing instructions are written in a variety of ways. A unified schema for consistent representation across tasks.
Natural-Instructions: Construction (4)

- This process was done by an expert annotator and verified by another.

- Mapping crowdsourcing instructions to our schema:
  - Retained the original phrasing.
  - Redacted verbose/repetitive content.
  - Created negative examples wherever they were absent.

- Took ~10 hours for each task.
Natural Instructions: Statistics

• 61 tasks
Natural-Instructions: Example

input instance (paragraph, story, etc.)

Positive Examples:

- Definition
- Things to Avoid
- Emphasize/Caution
- Prompt

Negative Examples:

- Input
- Output
- Reason

an output
(question, answer, label, etc.)
Generating “duration” questions task

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.

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".

Emphasize/Caution: The written questions are not required to have a single correct answer.

Positive example 1
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.
Generating “duration” questions task

**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.

**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".

**Emphasize/Caution:** The written questions are not required to have a single correct answer.

**Positive example 1**

**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.

---

**Input:** She chose to make a salad for lunch tomorrow and Sunday.

“how long did it take for her to make a salad?”

https://instructions.apps.allenai.org/explore
Encoding Instructions

**Input Instance (Paragraph, Story, etc.)**

**Positive Examples:**
- **Definition**
- **Things to Avoid**
- **Emphasize/Caution**
- **Prompt**

**Negative Examples:**
- **Input**
- **Output**
- **Reason**

**Some Output**
- (Question, Answer, Label, etc.)

**Text to Text Architectures:**
- **BART** [Lewis et al. 2019]
- **GPT3** [Brown et al. 2020]
Encoding Instructions

**input instance** *(paragraph, story, etc.)*

- **Positive Examples:**
  - **Definition**
  - **Things to Avoid**
  - **Emphasize/Caution**
  - **Prompt**

- **Negative Examples:**
  - **Input**
  - **Output**
  - **Reason**

**enc(.)**

- **some output** *(question, answer, label, etc.)*

**text to text architectures:**
- **BART** [Lewis et al. 2019]
- **GPT3** [Brown et al. 2020]
Empirical Questions

**input instance (paragraph, story, etc.)**

- **Q1:** What fields of the instructions are the most helpful for the models?
- **Q2:** Does GPT3 understand instructions?
- **Q3:** Can smaller model (such as BART) be tuned to follow instructions?

**text to text architectures:**
- BART [Lewis et al. 2019]
- GPT3 [Brown et al. 2020]
Experiment: Evaluating GPT3

• Does GPT3 understand task instructions?

• Instructions **improve** GPT3’s performance!

• All instruction elements (except negative examples) help!

• A wide margin to be solved 😐
  • A task-specific BART scores ~70%

![Bar chart showing output quality (ROUGE) for different instruction formats: prompt (30), prompt + definition (35), full instructions (24), full instructions - negative ex. (44).]
Evaluating Fine-tuned Models: Setup

• Splitting the data for fine-tuning a smaller model:

1. Randomly split the tasks
   • 12 evaluation tasks
   • 49 supervision tasks

2. Leave-one-category-out
Evaluating Fine-tuned Models: Setup

• Splitting the data for fine-tuning a smaller model:

1. Randomly split the tasks
   • 12 evaluation tasks
   • 49 supervision tasks

2. Leave-one-category-out
Evaluating Fine-tuned Models: Setup

• Splitting the data for fine-tuning experiments:

1. Randomly split the tasks
   • 12 evaluation tasks
   • 49 supervision tasks

2. Leave-one-category-out
Exp: Generalization to a Random Split

- Can models learn to act w.r.p. instructions?
  - BART (base) [Lewis et al. 2019]

- Small models, too, generalize to unseen tasks! 😁

- All instruction elements (except negative examples) help!
Exp: Generalization to Unseen Categories

• Evaluate on task of a particular category and train on the rest.

- Instructions improve generalization to tasks of unseen categories! ✨

---

<table>
<thead>
<tr>
<th>Instructions</th>
<th>Output Quality (ROUGE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>no instructions</td>
<td>11</td>
</tr>
<tr>
<td>prompt + definition</td>
<td>18</td>
</tr>
<tr>
<td>full instructions</td>
<td>19</td>
</tr>
<tr>
<td>full instructions - negative ex.</td>
<td>21</td>
</tr>
</tbody>
</table>

eval. on unseen “answer generation” tasks

<table>
<thead>
<tr>
<th>Instructions</th>
<th>Output Quality (ROUGE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>no instructions</td>
<td>6</td>
</tr>
<tr>
<td>prompt + definition</td>
<td>10</td>
</tr>
<tr>
<td>full instructions</td>
<td>17</td>
</tr>
<tr>
<td>full instructions - negative ex.</td>
<td>19</td>
</tr>
</tbody>
</table>

eval on unseen “question generation” tasks
**Exp: Generalization vs Size of Observed Tasks**

- How the number of observed tasks affects cross-task generalization?

- Generalization to unseen tasks **improves** with more observed tasks! 🔥

![Graph showing the relationship between the number of observed tasks and output quality. The graph indicates that output quality increases with more observed tasks.](image-url)
Lessons

• Motivating Hypothesis:
  • Can machines generalize to unseen tasks, via natural language instructions?

• Natural-Instructions: a dataset of many tasks and their crowdsourcing instructions/annotations.

• Empirical evidence that:
  • Instructions help w/ generalization to unseen tasks!
  • There is notable room to make progress!

http://instructions.apps.allenai.org
That’s it!