PromptSource: IDE + repo for natural language prompts

What it is | Why it is needed

Paper: S. Bach et al., "PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts", ACL Demo (2022)

Motivation

Prompting: represent task as an *utterance*

E.g. text classification prompt: "This video describes PromptSource. It is about..." → "NLP"

maps to class label

Prompt applications
- enables adapting LMs to *ad hoc* tasks
- sample efficiency in low data regimes

Prompt engineering can have a major influence

Training on prompts can enable task generalisation

But how can we enable users to: create, refine, share prompts?

PromptSource — IDE (Web GUI) — Shared repo

Elements of design:

- Template language — simple but flexible
- Prompt management — support browsing/iteration
- Community-driven standards — evolved guidelines

Stats: 2087 prompts across 180 datasets

Public Pool of Prompts (P3)

*P3 stats from September 2022

References:
T. Brown et al., "Language models are few-shot learners", NeurIPS (2020)
T. Schick et al., "It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners", NAACL-HLT (2021)
Le Scao et al., "How many data points is a prompt worth?", NAACL-HLT (2021)
E. Perez et al., "True few-shot learning with language models", NeurIPS (2021)
V. Sanh et al., "Multitask Prompted Training Enables Zero-Shot Task Generalization", ICLR (2021)

Prompt creation ≠ traditional NLP annotation

What makes prompt creation different?

Functions, not labels

Prompts are functions that map examples to input/target pairs - how expressive should format be?

Datasets, not examples

Unlike labels, prompt design must consider all dataset examples - what interface will support this?

Variation is desirable

Label variation generally undesirable but prompt variation has benefits - how can this be supported?

References:
S. Bach et al., "PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts", ACL Demo (2022)

The PromptSource workflow

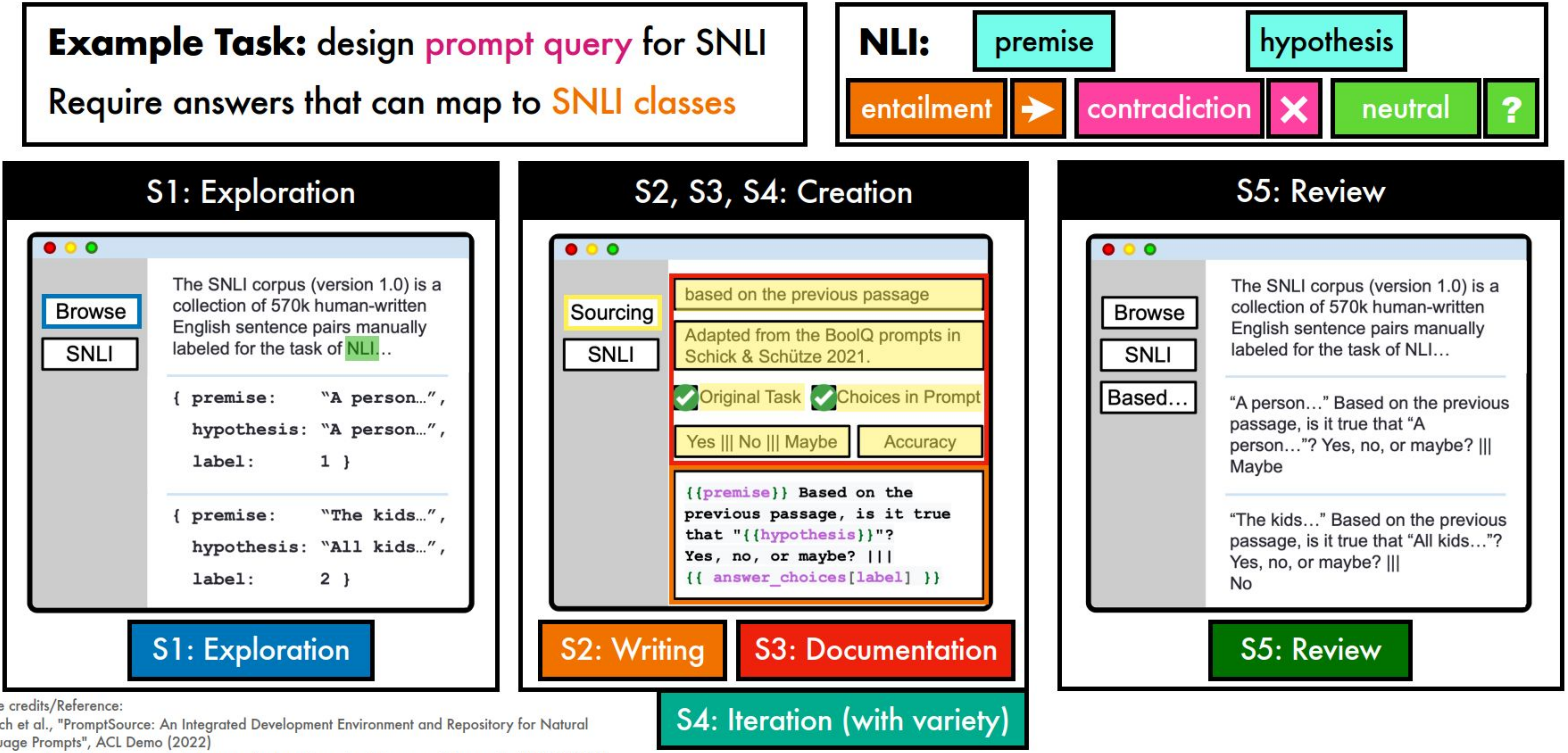


Image credits/Reference:
S. Bach et al., "PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts", ACL Demo (2022)
S. Bowman et al., "A large annotated corpus for learning natural language inference", EMNLP (2015)

PromptSource plays nicely with 🤗 Datasets

Practical usage

Import libraries

```
from promptsource.templates import DatasetTemplates
from datasets import load_dataset
```

Fetch prompt by name

```
prompts = DatasetTemplates("snli")
prompt_key = "based on the previous passage"
p = prompts[prompt_key]
```

Load example

```
dataset = load_dataset("snli", split="train")
example = dataset[0]
```

Apply to an example

```
result = p.apply(example)
print("INPUT: ", result[0])  A person ....? Yes, no, or maybe?
print("TARGET: ", result[1]) Maybe
```

Image credits/Reference:
S. Bach et al., "PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts", ACL Demo (2022)
Q. Lhoest et al., "Datasets: A community library for natural language processing", EMNLP Demo (2021)

Prompt Template Engine: Jinja2

Jinja2 template engine used for prompts

- More flexible vs rule-based generation
- Simpler than pure Python code

Example prompt:

```
If {{premise}} holds, does {{hypothesis}}
also hold? ||| {{entailed}}
```

Placeholder refer to fields in example `dict`

Separator between condition and completion

Jinja2 enables some fancy string manipulation

In practice, simple manipulations suffice

Useful idioms

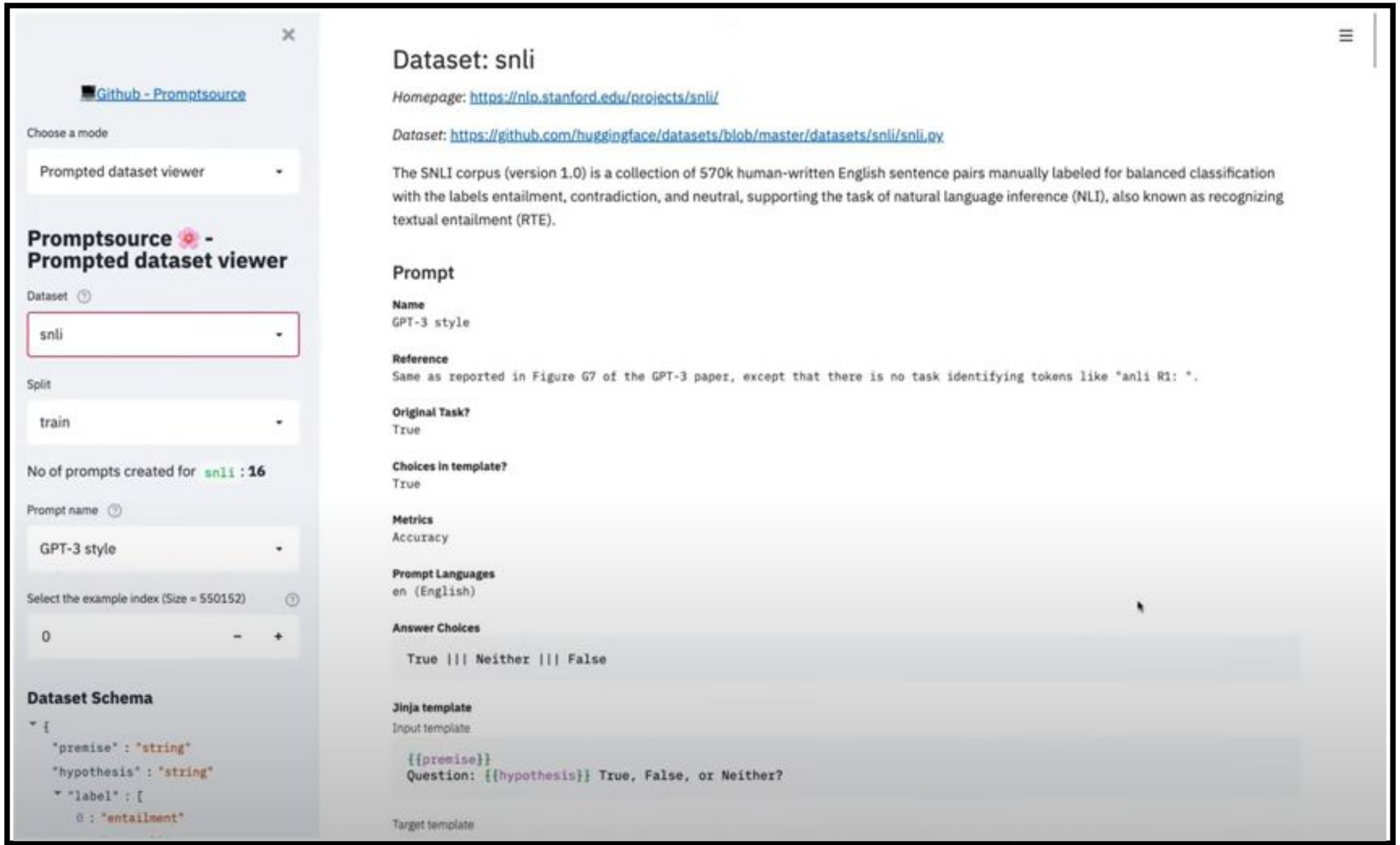
Useful idioms
Template may not be applicable for all examples Conditionals can be used to skip examples (empty)
Examples can generate multiple training instances Elements can be selected with the `choice` function
Some examples may have multiple valid completions These can be specified as a separate field

References
S. Bach et al., "PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts", ACL Demo (2022)
(Example dict format) Q. Lhoest et al., "Datasets: A community library for natural language processing", EMNLP Demo (2021)
(Guidelines for prompt authors) https://github.com/bigscience-workshop/promptsource/blob/main/CONTRIBUTING.md

User Interface: Dataset Browsing

useful for verifying prompts across many examples

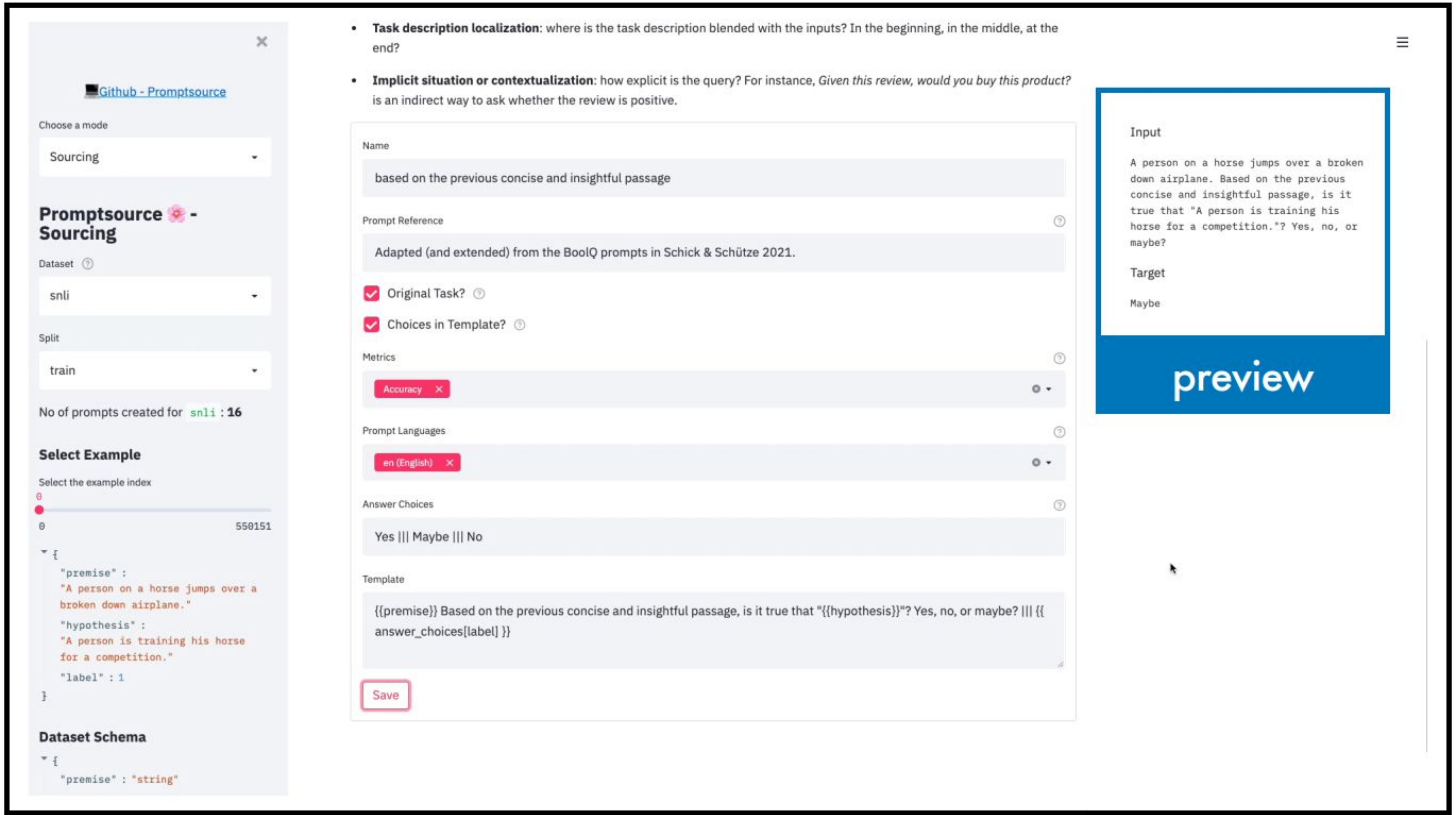
Built with Streamlit



Dataset browsing

References:
S. Bach et al., "PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts", ACL Demo (2022)
https://github.com/streamlit

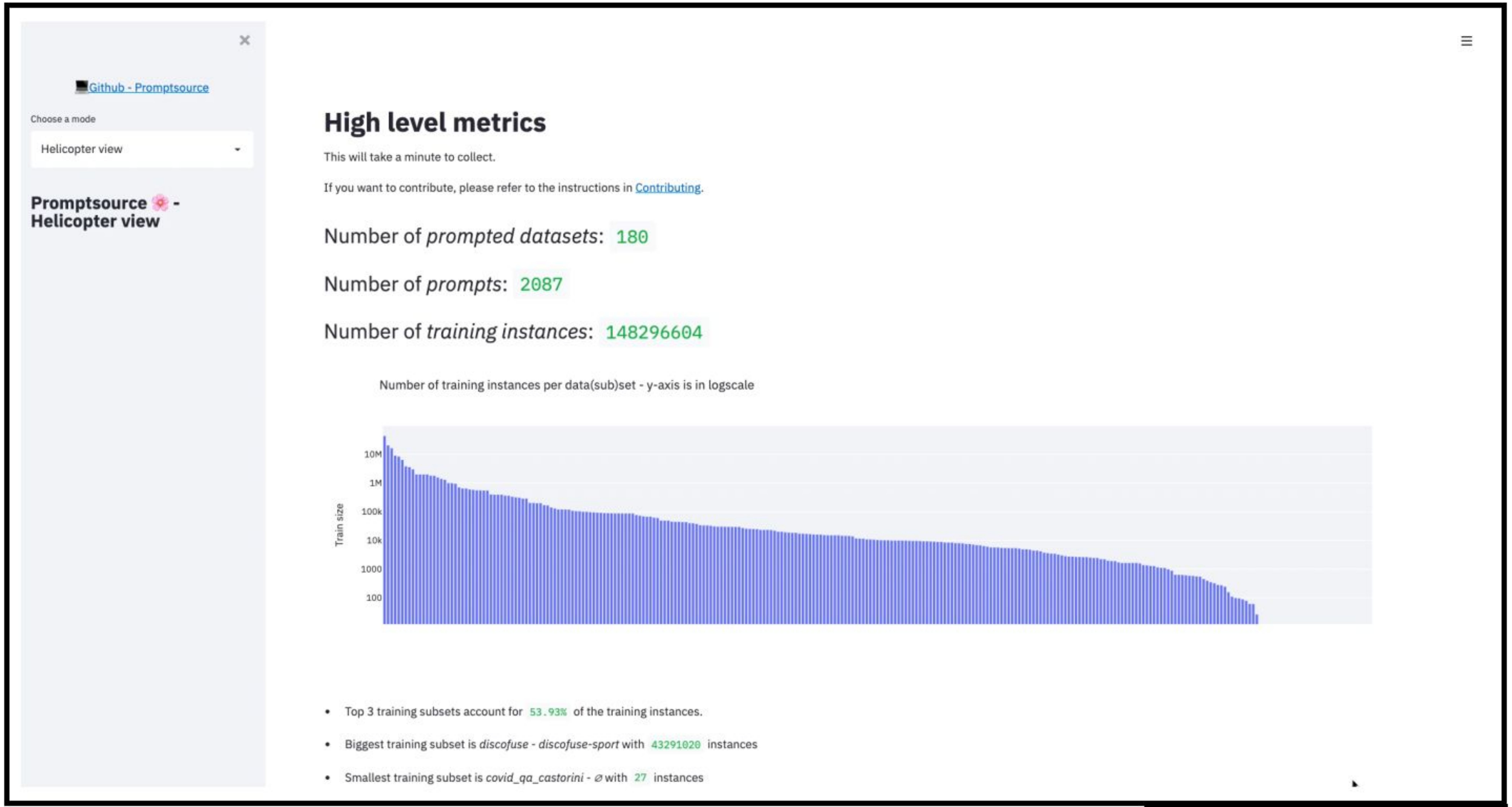
User Interface: Sourcing



Prompt sourcing

References:
S. Bach et al., "PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts", ACL Demo (2022)

User Interface: Helicopter view



Helicopter view

References:
S. Bach et al., "PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts", ACL Demo (2022)

Community Guidelines

Describing what makes a good prompt is hard

Community guidelines evolved through iteration

Key objectives for guidelines:

- Standardised vocab & minimum requirements
- Highlight common errors & best practices
- Gather useful metadata for future research

The guidelines cover templates and metadata

Encourage:

- explicitly state possible completions
- remove spurious ambiguity from targets
- creation of multiple prompt variants

Require:

- only natural language prompts allowed
- inclusion of metadata (e.g. reference paper)

References:
S. Bach et al., "PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts", ACL Demo (2022)
(Guidelines for prompt authors) https://github.com/bigscience-workshop/promptsource/blob/main/CONTRIBUTING.md

Case Studies with PromptSource

Massively multitask prompted training	Multilingual prompting	Priming (in-context learning)
T0 (Sanh et al., 2021) uses a multitask mixture of prompts for training to boost generalisation	XGLM (Lin et al., 2021) train on 30 languages to study cross-lingual generalisation	MetaICL (Lin et al., 2021) train on a multitask mixture with in-context learning examples
Training and evaluation use P3	P3 - quality English prompts	Instructions from P3 bring gains

References:
S. Bach et al., "PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts", ACL Demo (2022)
V. Sanh et al., "Multitask Prompted Training Enables Zero-Shot Task Generalization", ICLR (2021)
V. Lin et al., "Few-shot learning with multilingual language models", arXiv (2021)
S. Min et al., "MetaICL: Learning to learn in context", arXiv (2021)

Prior work

Prompting
Prompting GPT-3 PET T0 FLAN
PromptSource aims to support research with human-written prompts
Originally focused on zero-shot learning (emphasis on explicit task instructions)
Can be extended for few-shot learning: XGLM MetaICL

Systems for annotating data
GATE (2002) system for text annotation
Web systems BRAT (2012) MyMiner (2012)
Collaboration for annotators YEDDA (2018)
Leverage active learning AlpacaTag (2019)
Non-label annotation TreeAnnotator (2018)
Prompts are semi-structured functions
They merit new tools for annotation

References:
S. Bach et al., "PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts", ACL Demo (2022)
(GPT-3) T. Brown et al., "Language models are few-shot learners", NeurIPS (2020)
(PET) T. Schick et al., "Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference", EACL (2021)
(T0) V. Sanh et al., "Multitask Prompted Training Enables Zero-Shot Task Generalization", ICLR (2021)
(FLAN) J. Wei et al., "Finetuned Language Models are Zero-Shot Learners", ICLR (2021)
(XGLM) V. Lin et al., "Few-shot learning with multilingual language models", arXiv (2021)
(MetaICL) S. Min et al., "MetaICL: Learning to learn in context", arXiv (2021)
(GATE) H. Cunningham, "GATE, a general architecture for text engineering." Computers and the Humanities (2002)
(BRAT) P. Stenetorp et al., "BRAT: a web-based tool for NLP-assisted text annotation", EACL (2012)
(MyMiner) D. Salgado et al., "MyMiner: a web application for computer-assisted biocuration and text annotation", Bioinformatics (2012)
(YEDDA) J. Yang et al., "YEDDA: A Lightweight Collaborative Text Span Annotation Tool", ACL (2018)
(AlpacaTag) B. Lin et al., "AlpacaTag: an active learning-based crowd annotation framework for sequence tagging", ACL (2019)
(TreeAnnotator) P. Helfrich et al., "TreeAnnotator: versatile visual annotation of hierarchical text relations", LREC (2018)

Thank You