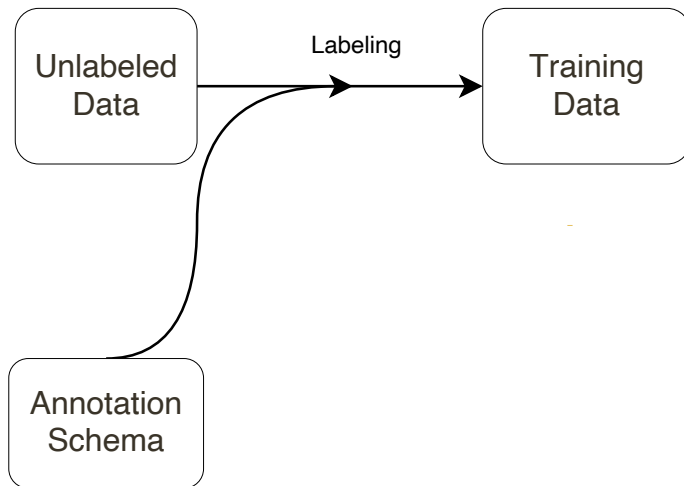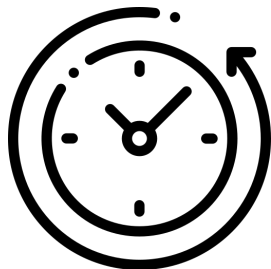# MEGAnno: Exploratory Labeling for NLP in Computational Notebooks
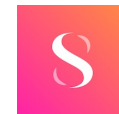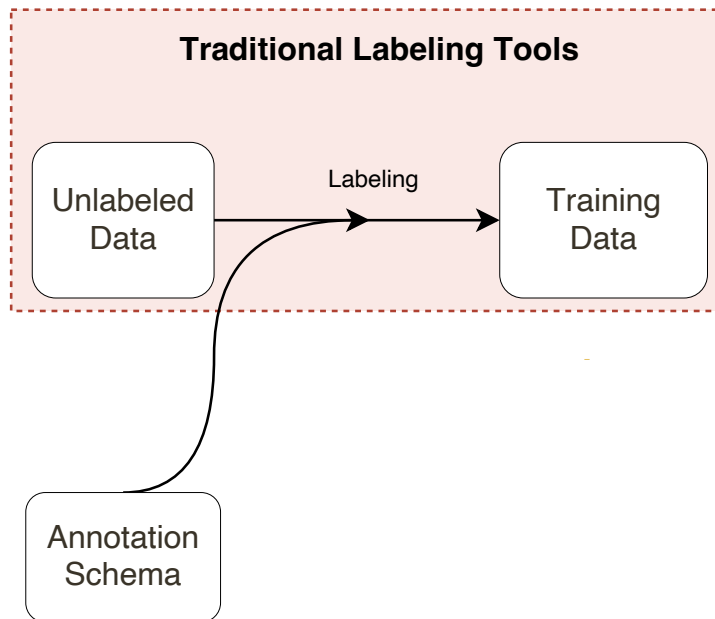
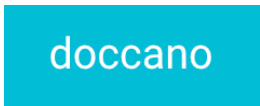Dan Zhang, Hannah Kim, Rafael Li Chen, Eser Kandogan, Estevam Hruschka

DaSH
Data Science with Human-in-the-Loop

Megagon Labs

1

Data Labeling

Unlabeled Data → Labeling → Training Data

Annotation Schema

(Standalone) Data Labeling

Traditional Labeling Tools

Unlabeled Data → Labeling → Training Data

Annotation Schema

datasaur.ai

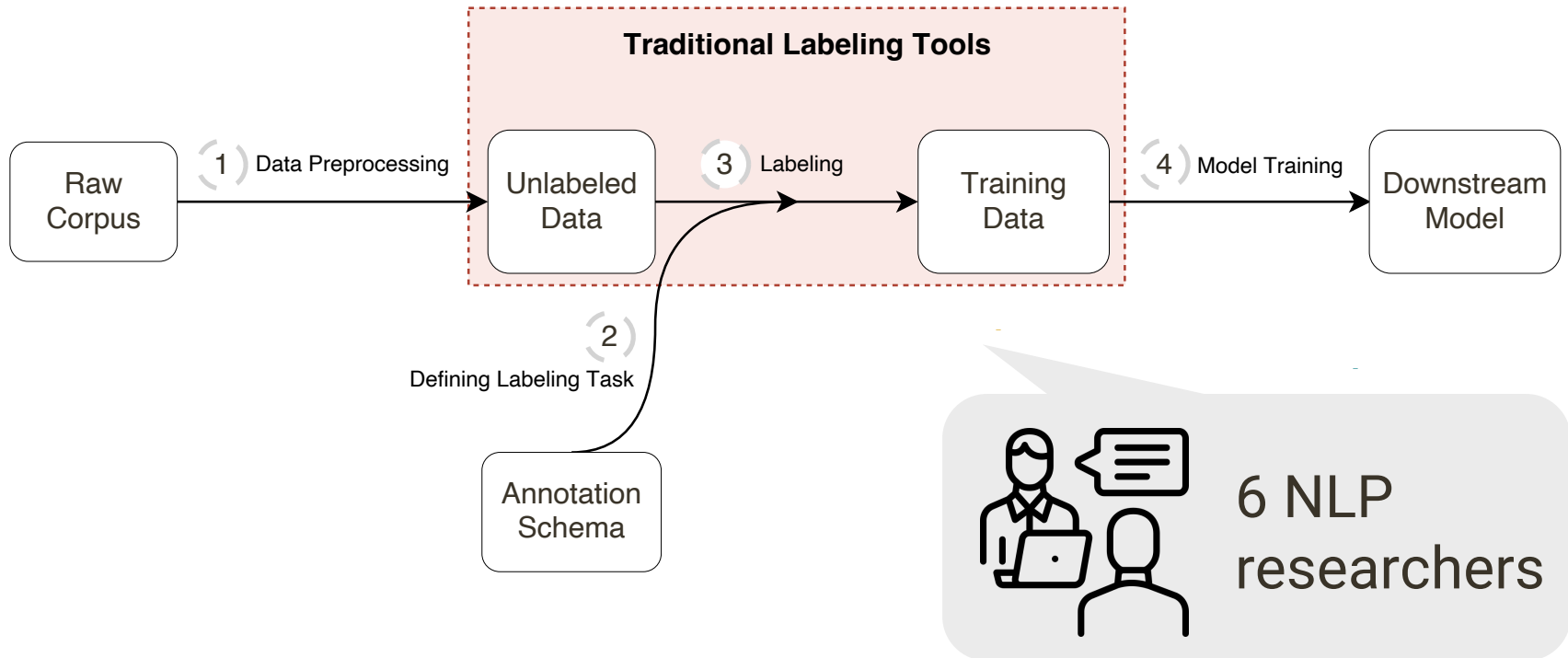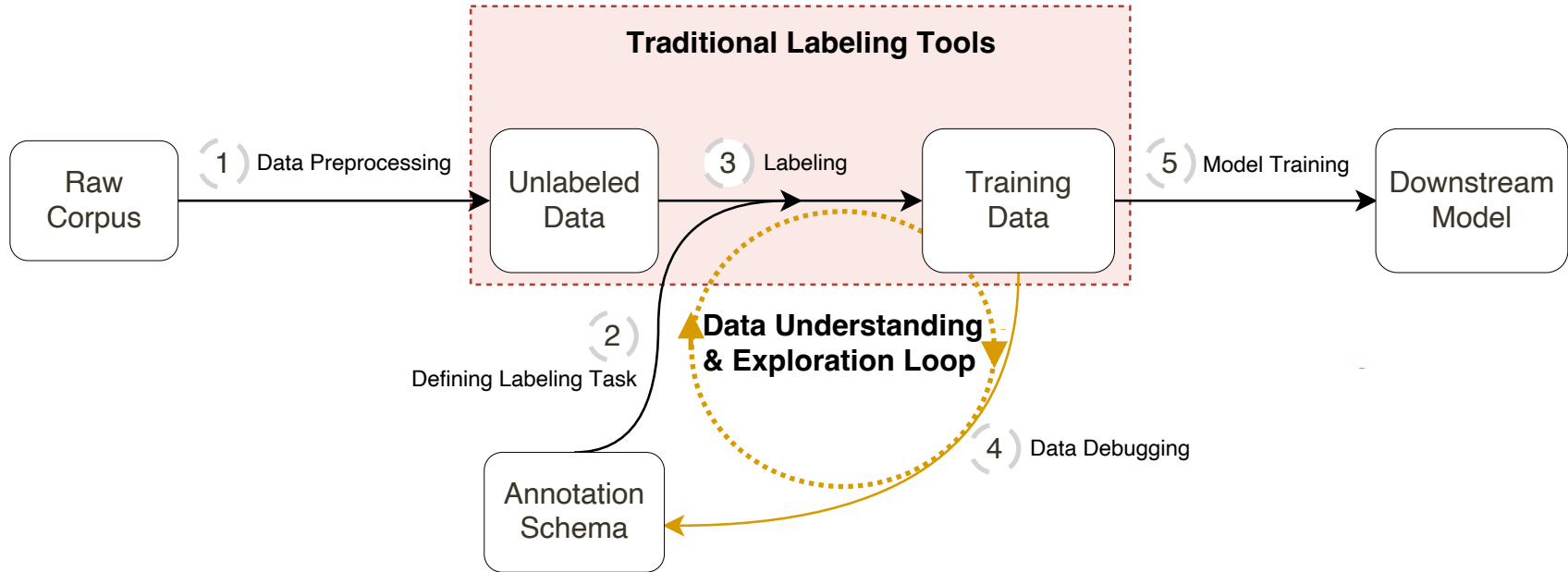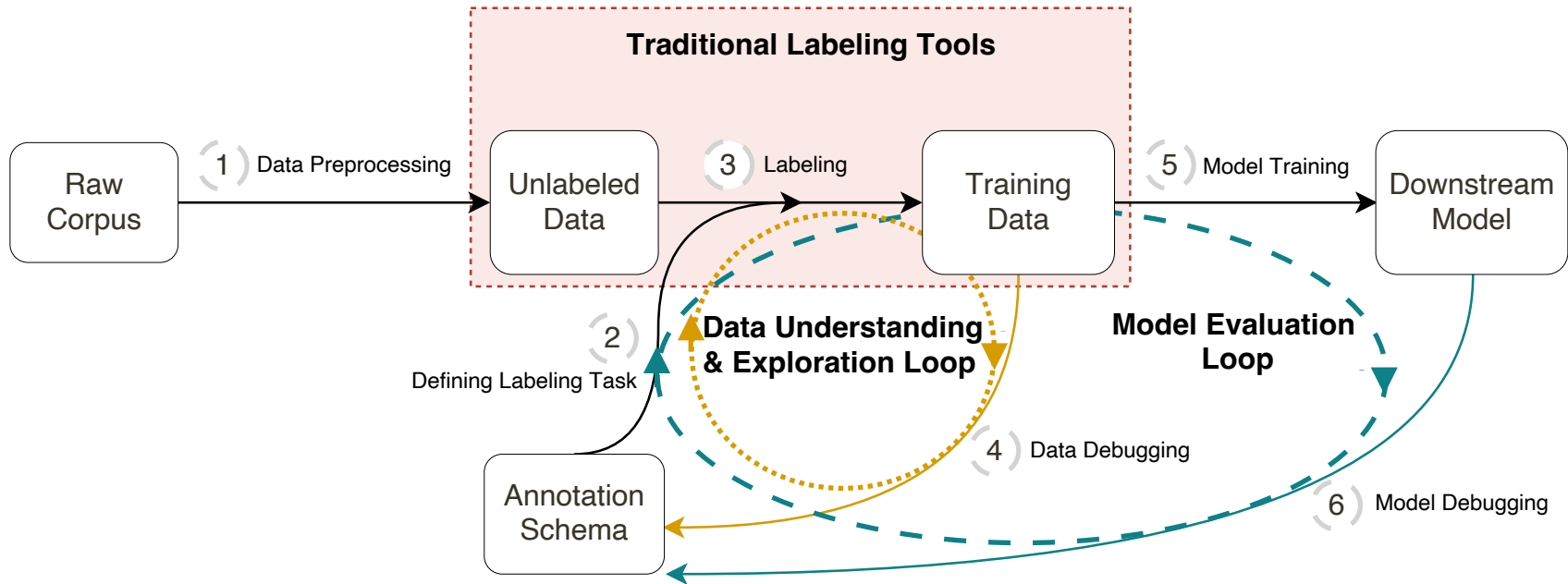Label Sleuth

clarifai

And many more...

Data Labeling *within ML Workflow*

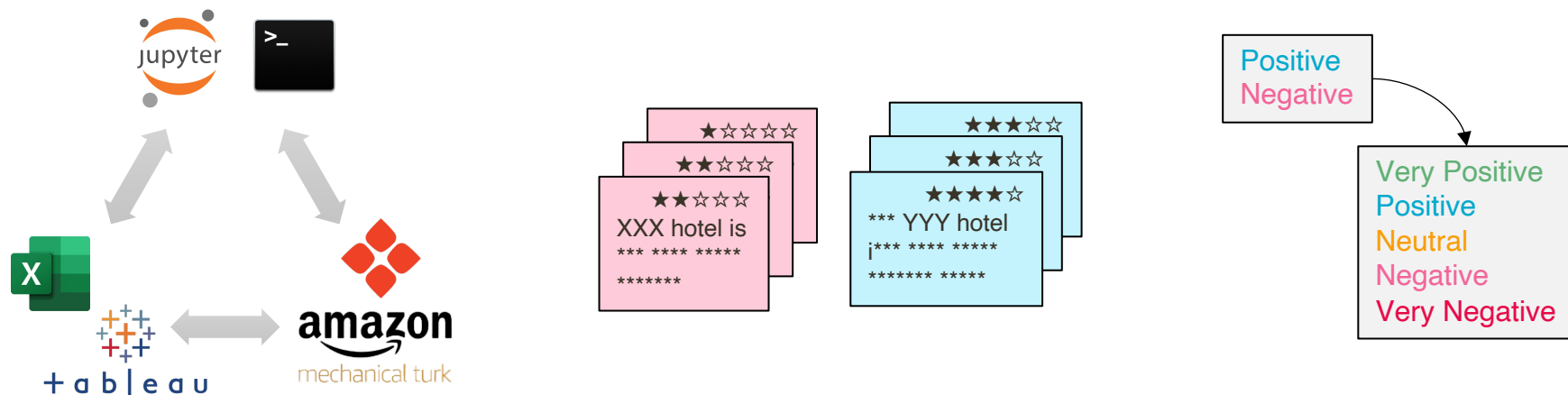# Dual-Loop Model of Data Labeling

Dual-Loop Model of Data Labeling

Challenges from Dual-Loop Model

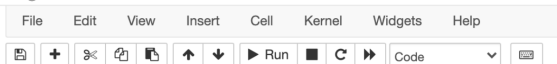Gaps between ML tools

Lack of customizable and fine-grained control

Lack of support for project evolution

We present MEGAnno, a flexible, exploratory, efficient, and seamless labeling framework for NLP researchers and practitioners.

Use Case: Sentiment Analysis
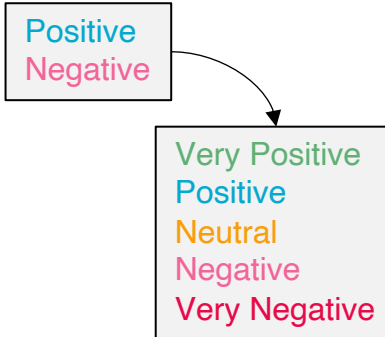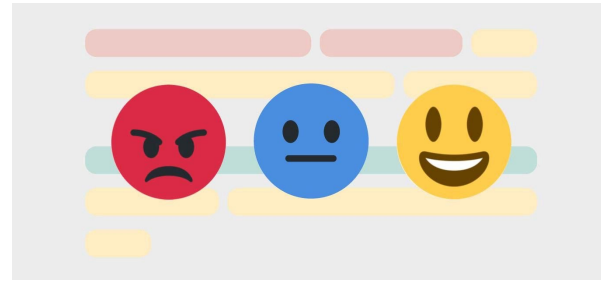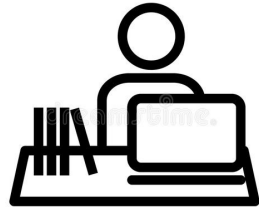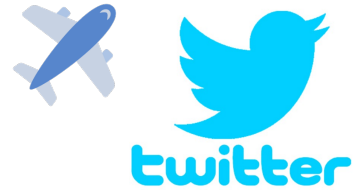
Import Data and Set Up Schema

```python
from meganno import Project
L = Project(<auth>, "Tweet Sentiment")
L.import_data(<doc_url>, format="csv")
```

```python
label_schema = [{
    "label_name": "sentiment_label",
    "level": "record",
    "options": ["Positive", "Negative"]
}]
L.set_config(config1)
```

Explore and Annotate with Our Widget

```
# search results => subset s1
s1 = demo.search(keyword='', limit=10, start=0)
# bring up a widget
s1.show()
```

Switching between
Single View and **Table View**

Batch labeling

Sort and filter based on text, label, and metadata (e.g., # of RT, timestamp, author, …)

Exploratory Labeling - Search

API

By keyword:
```
L.search(keyword='amazing')
```

By regex:
```
L.search(regex=
'.*#fai[^ ]* .*')
```

By user-defined metadata:
```
L.search(pattern=
"terrible <ADJ> <NOUN>",
tagger=spacy.load("en_core_w
eb_sm"))
```

Widget

Exploratory Labeling - Suggestion

```python
# Example 1, adding sentence-bert embedding.
# metadata searches.
from sentence_transformers import SentenceTransformer
model = SentenceTransformer('all-MiniLM-L6-v2')
# set metadata generation function
sv.set_metadata("bert-embedding",lambda x: list(model.encode(x).astype(float)), 500)
```

Find similar or diverse samples to label

...

```python
s3 = sv.search(keyword='delay', limit=3, start=0)
s4 = s3.suggest_similar('bert-embedding', limit=4)# needs to provide a valid meta_name
s4.show()
```

show(7fd1aff79a60, self=<labeler_ui.widgets.Annotation.Annotation object at 0x7fd1b1033d00>)

			data	sentiment
				⊟ Bulk label
1		☐	@USAirways UR service is so shitty. Pilot never showed up so we waited hours because another pi	-
2		☐	@SouthwestAir Being old I will miss my connection and can sleep at OAK or get a hotel on my own	-
3		☐	@united my flight was at 1 pm. Still at the airport gee what service. Do they care, nope. Delayed ur	-

14

Update Annotation Schema

```python
sv.get_schemas().set_schemas({
    'label_schema': [
        {
            "name": "sentiment",
            "level": "record",
            "options": [
                { "value": "pos", "text": "positive" },
                { "value": "neg", "text": "negative" },
                { "value": "neu", "text": "neutral" },

            ]
        })
```

sentiment

○ positive (pos)
● negative (neg)
○ neutral (neu)

@united You allow shady 3rd party services to sell your tickets and then refuse to help

the customer, who's left holding the bag. #flyunited

```python
sv.get_schemas().set_schemas({
    'label_schema': [
        {
            "name": "sentiment",
            "level": "record",
            "options": [
                { "value": "pos", "text": "positive" },
                { "value": "neg", "text": "negative" },
                { "value": "neu", "text": "neutral" },

            ]
        },
        # adding a span-level label
        {
            "name": "sp",
            "level": "span_ch",
            "options": [
                { "value": "pos", "text": "positive" },
                { "value": "neg", "text": "negative" },
            ]
        }
    ]
})
```

Manage Annotation Status
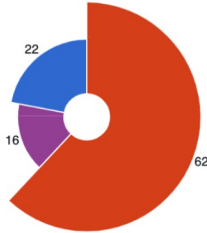
Monitor progress, distribution

Reconcile conflicts from multiple annotators

MEGAnno Features

Data Model

Task Schema

API

Interactive Widgets

...
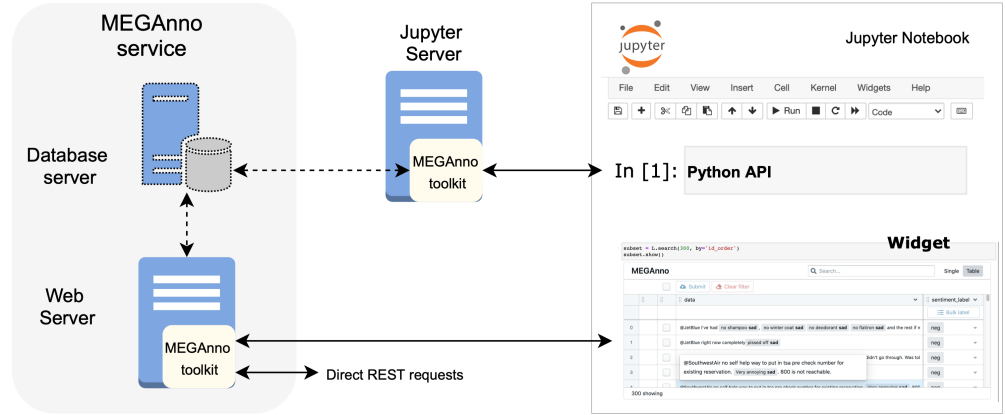
Please refer to our paper!

MEGAnno:
Exploratory Labeling for NLP in Computational Notebooks

Flexible annotation schema to support evolving projects

Efficient, exploratory labeling powered by our API and widget

Seamlessly integrated into existing ML environments

Demo at our booth (#7)
& meganno.github.io

Megagon Labs

18