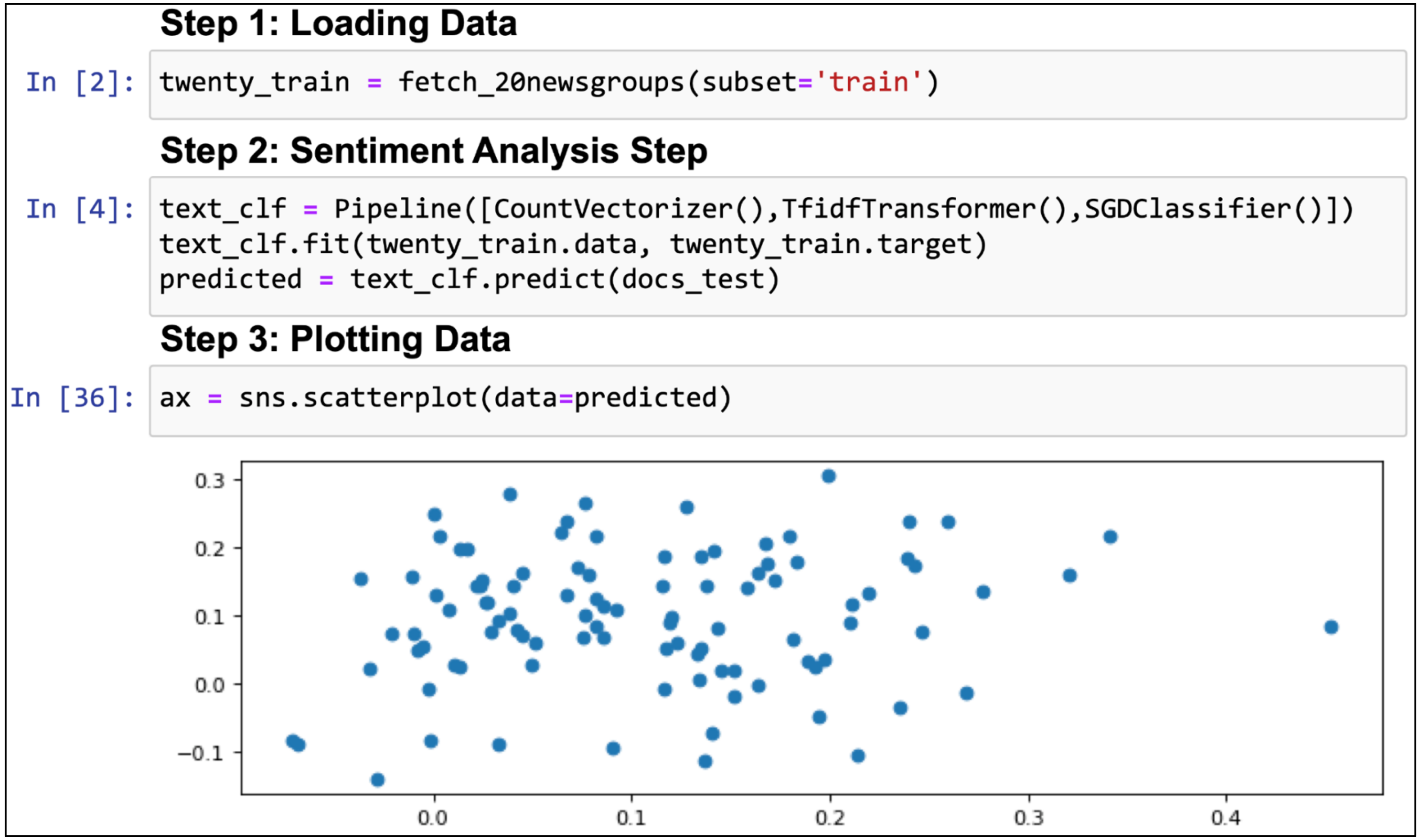
UCIrvine

VLDB2024
GUANGZHOU

Texera: A System for Collaborative and Interactive Data Analytics Using Workflows

Zuozhi Wang, **Yicong Huang**, Shengquan Ni, Avinash Kumar, Sadeem Alsudais, Xiaozhen Liu, Xinyuan Lin, Yunyan Ding, and Chen Li

The goto Data Science Tool - Notebook



Step 1: Loading Data

```
In [2]: twenty_train = fetch_20newsgroups(subset='train')
```

Step 2: Sentiment Analysis Step

```
In [4]: text_clf = Pipeline([CountVectorizer(),TfidfTransformer(),SGDClassifier()])
        text_clf.fit(twenty_train.data, twenty_train.target)
        predicted = text_clf.predict(docs_test)
```

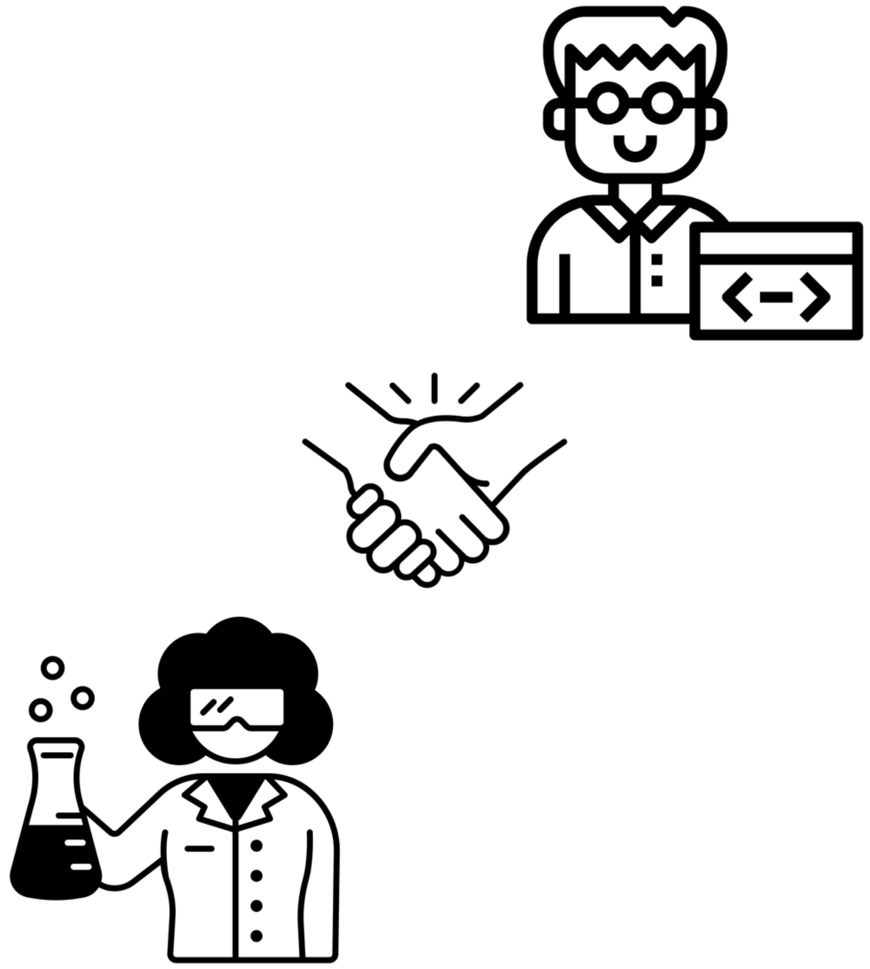
Step 3: Plotting Data

```
In [36]: ax = sns.scatterplot(data=predicted)
```

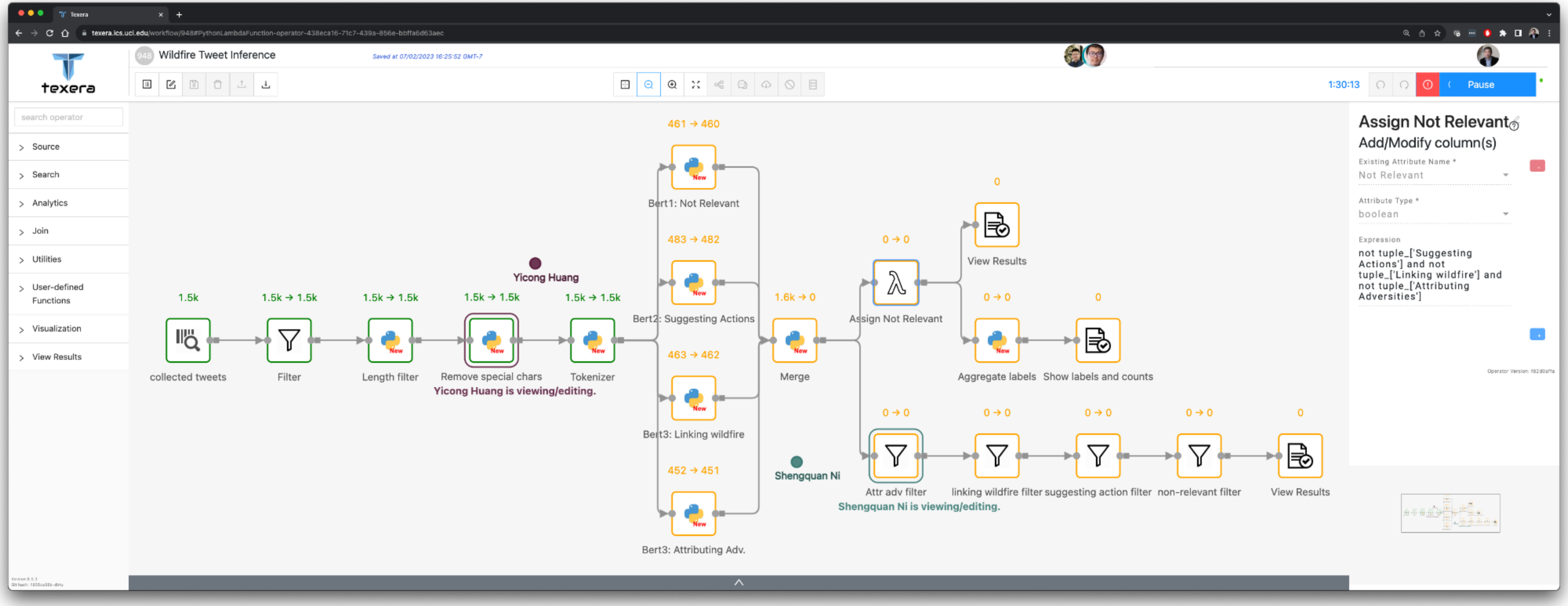
The Need for Collaborative Data Analytics

- **IT experts**
  - Limited domain knowledge
  - Strong coding skills

- **Domain experts**
  - Rich domain knowledge
  - Limited IT/coding skills



Introducing texera
The **GUI-based Workflow** System for Data Analytics



Wildfire Tweet Inference

Saved at 07/02/2023 16:25:52 GMT-7

Yicong Huang is viewing/editing.

Shengquan Ni is viewing/editing.

Assign Not Relevant

Add/Modify column(s)

Existing Attribute Name *
Not Relevant

Attribute Type *
boolean

Expression
not tuple_['Suggesting Actions'] and not tuple_['Linking wildfire'] and not tuple_['Attributing Adversities']

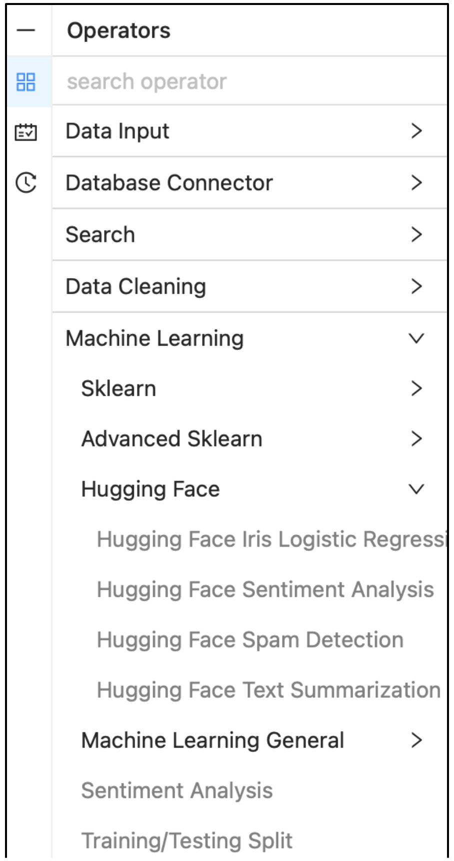
Cloud Service

Collaboration

Domain Friendly

AI/ML Access

More Features...



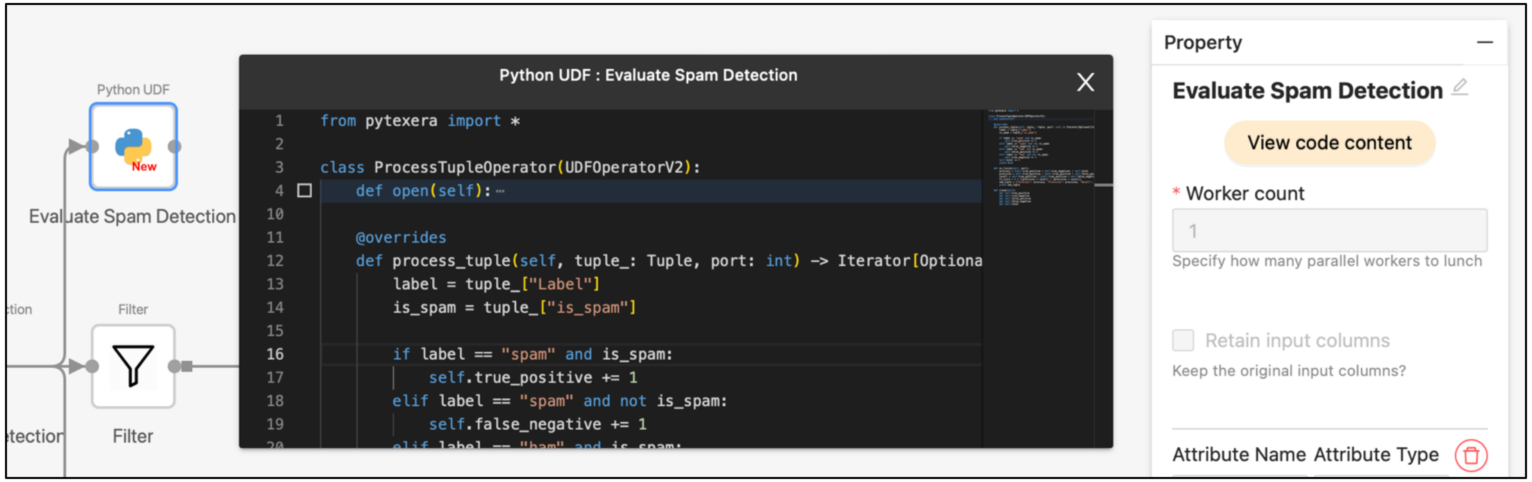
Operators

search operator

- Data Input >
- Database Connector >
- Search >
- Data Cleaning >
- Machine Learning ∨
  - Sklearn >
  - Advanced Sklearn >
  - Hugging Face ∨
    - Hugging Face Iris Logistic Regressi
    - Hugging Face Sentiment Analysis
    - Hugging Face Spam Detection
    - Hugging Face Text Summarization
  - Machine Learning General >
  - Sentiment Analysis
  - Training/Testing Split

Rich Built-in Operators

User Defined Functions (UDFs)
Java/Scala, Python, R



Python UDF : Evaluate Spam Detection

```python
from pytexera import *

class ProcessTupleOperator(UDFOperatorV2):
    def open(self):⋯

    @overrides
    def process_tuple(self, tuple_: Tuple, port: int) -> Iterator[Optiona
        label = tuple_["Label"]
        is_spam = tuple_["is_spam"]

        if label == "spam" and is_spam:
            self.true_positive += 1
        elif label == "spam" and not is_spam:
            self.false_negative += 1
        elif label == "ham" and is_spam:
```

Property

Evaluate Spam Detection

View code content

* Worker count

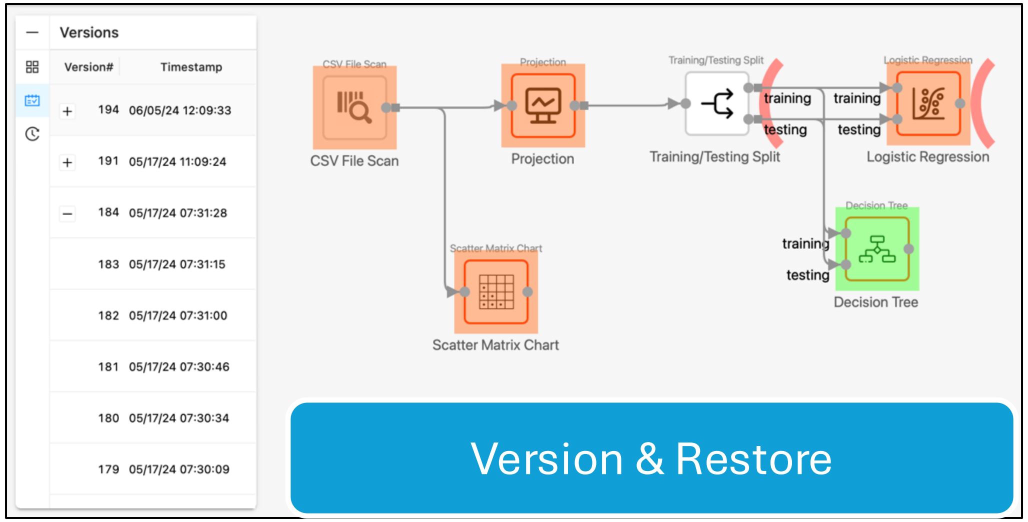
1

Specify how many parallel workers to lunch

□ Retain input columns

Keep the original input columns?

Attribute Name Attribute Type

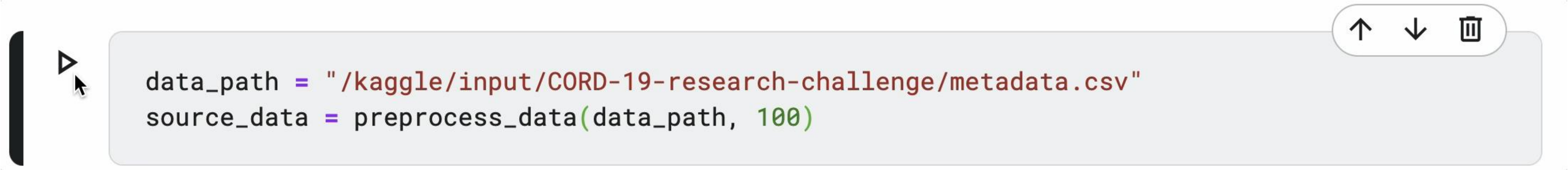


Versions

	Version#	Timestamp
+	194	06/05/24 12:09:33
+	191	05/17/24 11:09:24
−	184	05/17/24 07:31:28
	183	05/17/24 07:31:15
	182	05/17/24 07:31:00
	181	05/17/24 07:30:46
	180	05/17/24 07:30:34
	179	05/17/24 07:30:09

Version & Restore

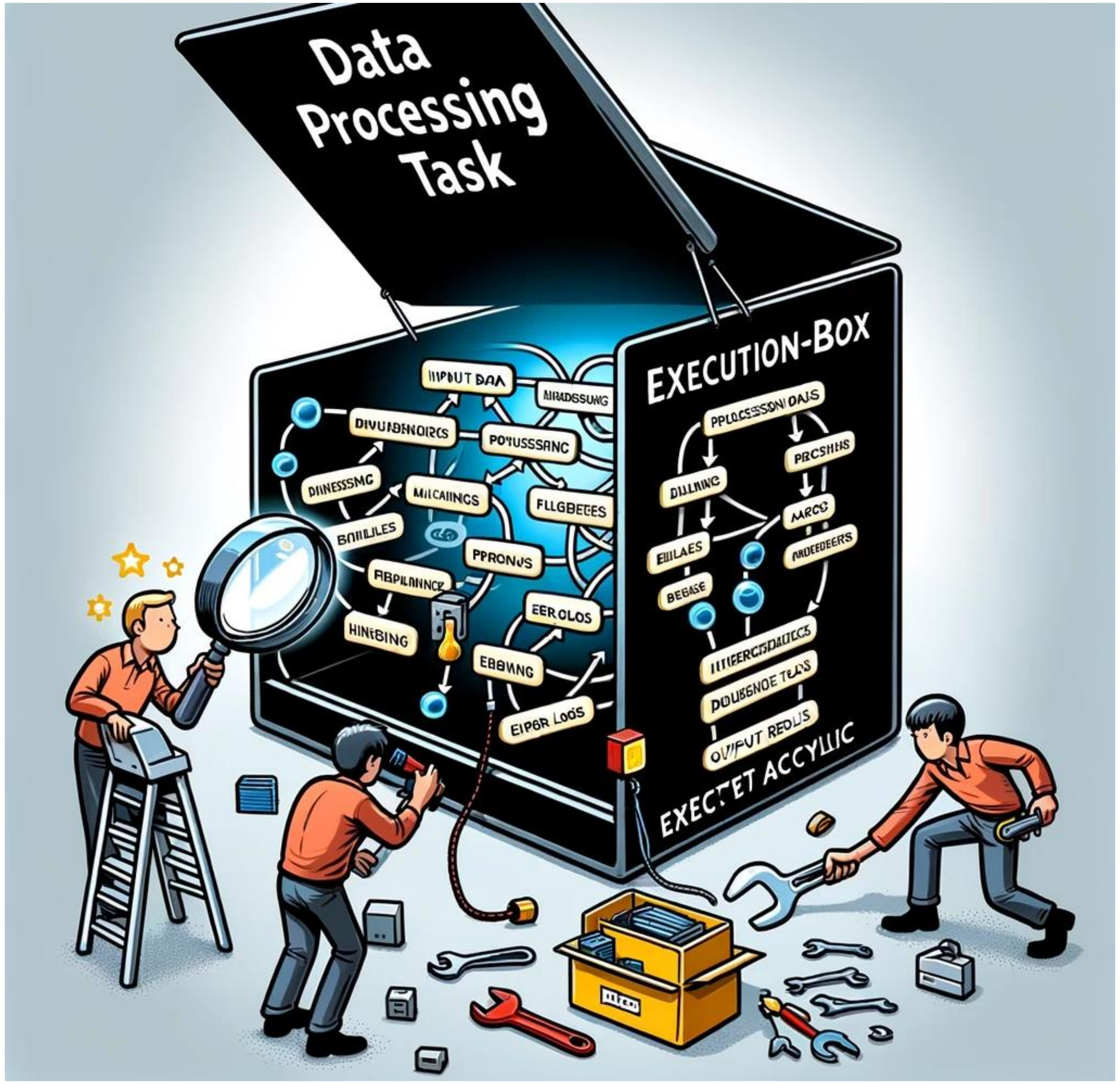
Collaboration only during Editing is not Enough for Data Analytics...



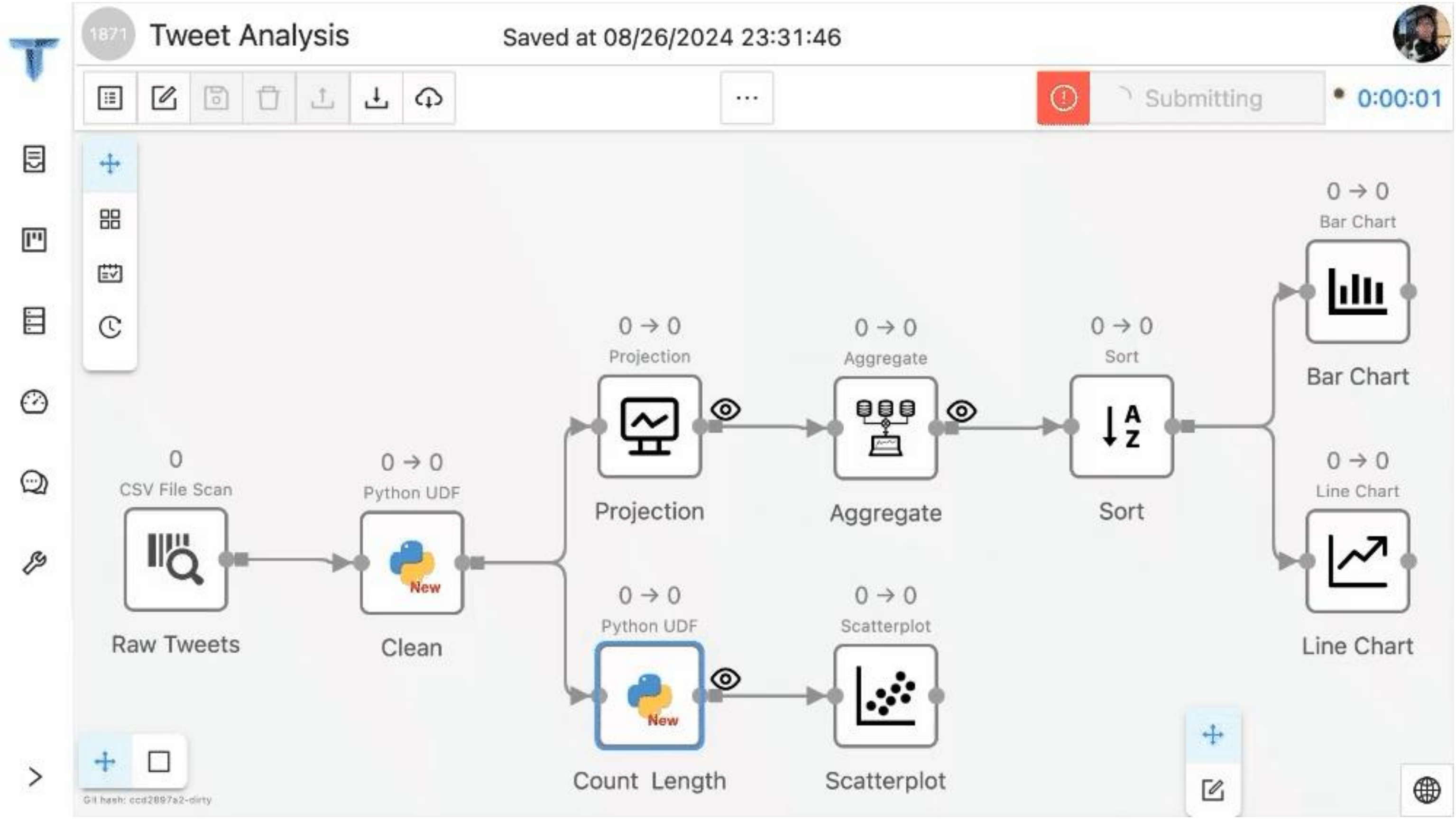
```
data_path = "/kaggle/input/CORD-19-research-challenge/metadata.csv"
source_data = preprocess_data(data_path, 100)
```

Unlike Google Doc or Overleaf, data analytics requires an (extensive) execution phase.

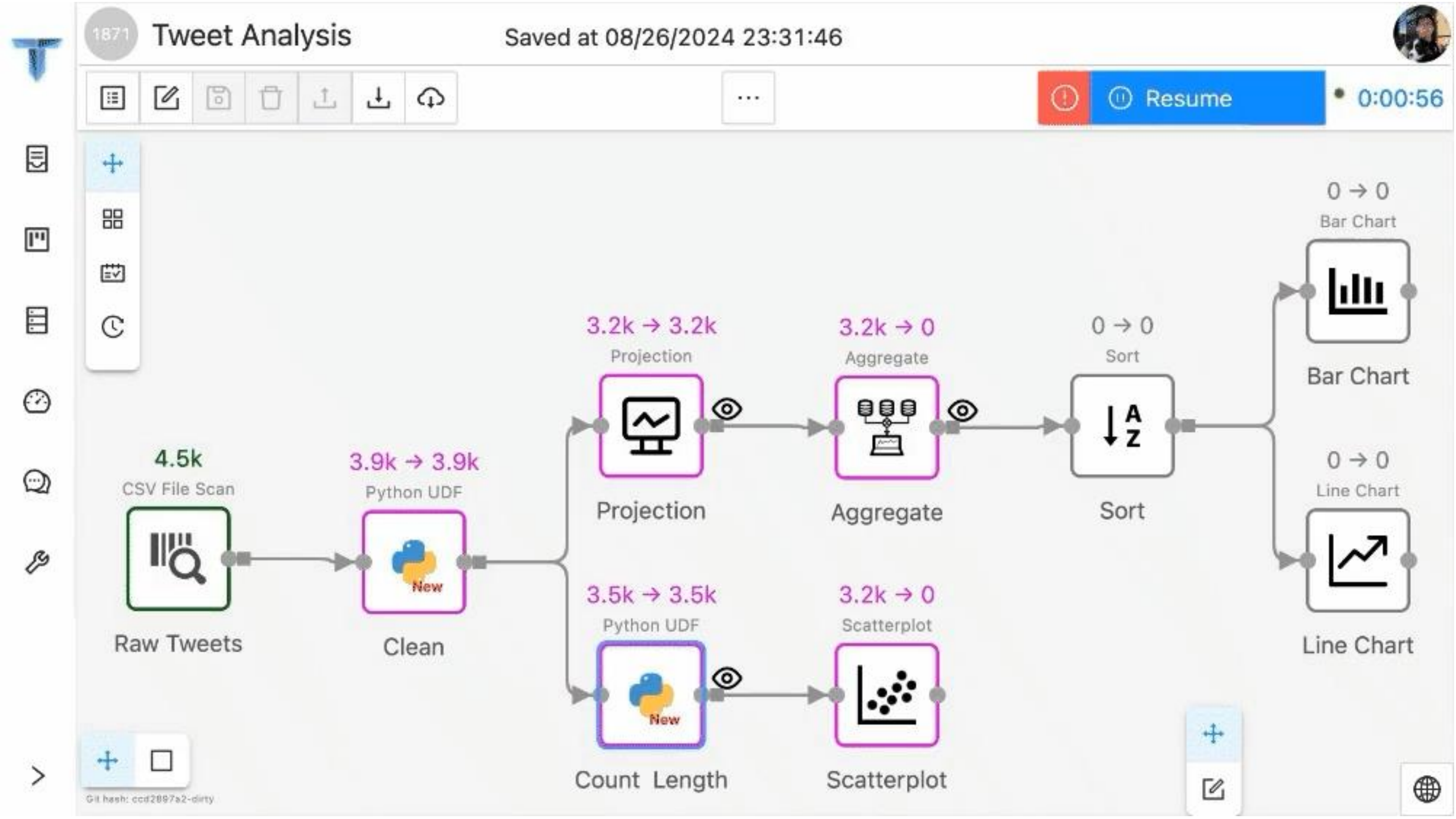
Interactions during Collaborative Execution



What Kind of Interactions?

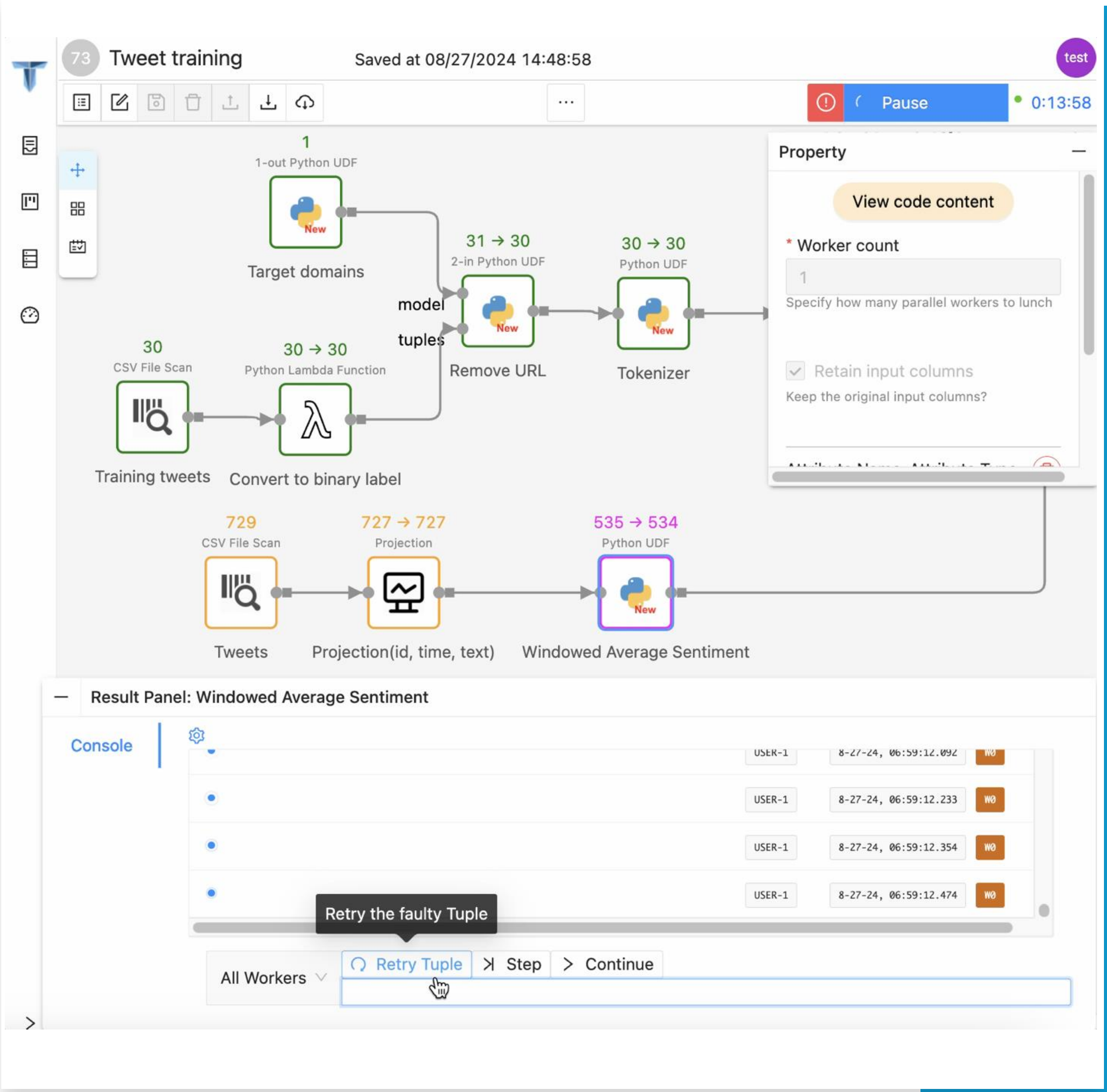


Interaction: Pause a Workflow



Interaction: Resume a Workflow

Interaction: Read a Workflow's State



* Worker count: 1

Specify how many parallel workers to lunch

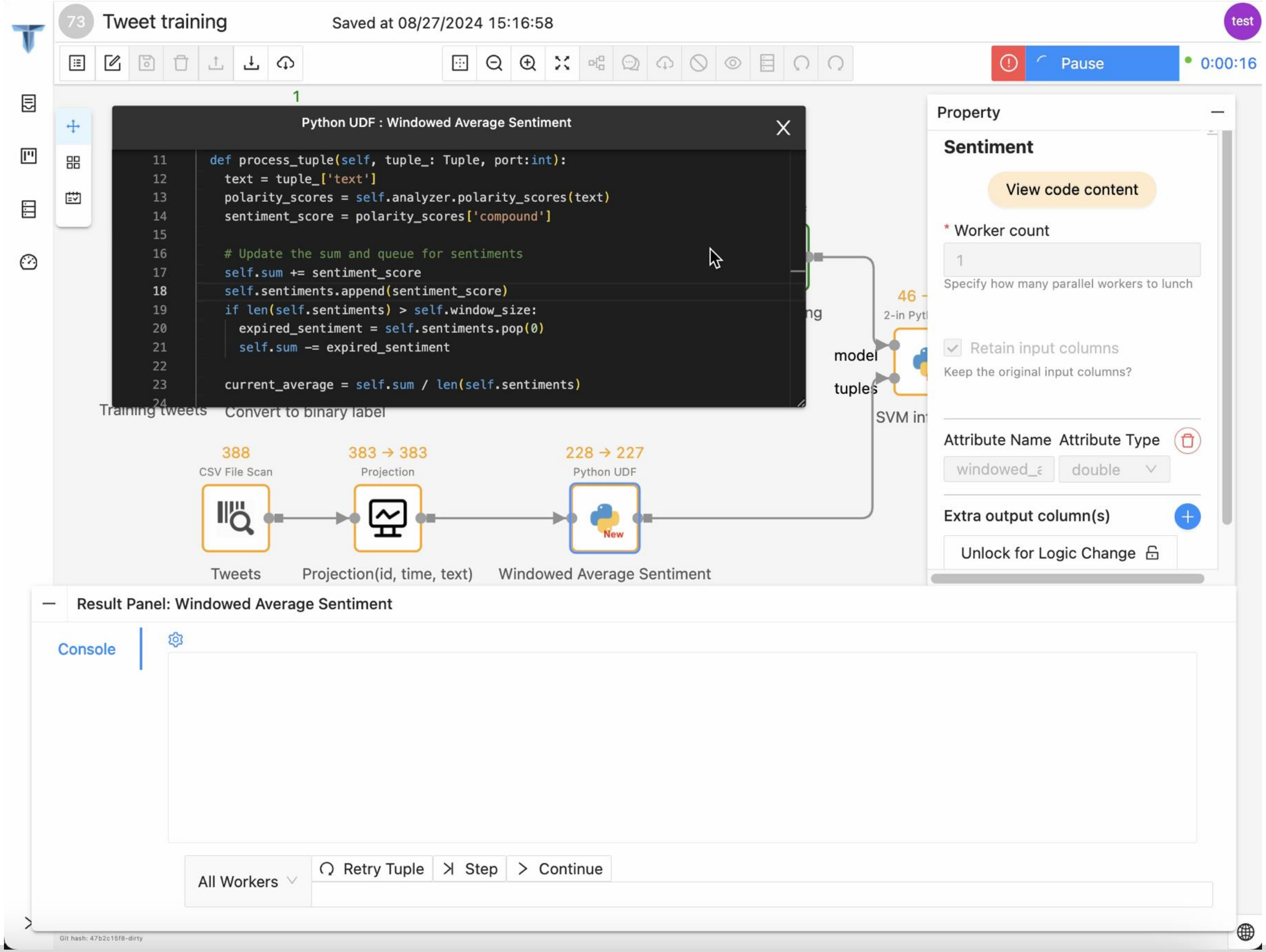
☑ Retain input columns

Keep the original input columns?

Result Panel: Windowed Average Sentiment

```
USER-1   8-27-24, 06:59:12.092   W0
USER-1   8-27-24, 06:59:12.233   W0
USER-1   8-27-24, 06:59:12.354   W0
USER-1   8-27-24, 06:59:12.474   W0
```

Interaction: Modify a Workflow's Logic



```
def process_tuple(self, tuple_: Tuple, port:int):
  text = tuple_['text']
  polarity_scores = self.analyzer.polarity_scores(text)
  sentiment_score = polarity_scores['compound']

  # Update the sum and queue for sentiments
  self.sum += sentiment_score
  self.sentiments.append(sentiment_score)
  if len(self.sentiments) > self.window_size:
    expired_sentiment = self.sentiments.pop(0)
    self.sum -= expired_sentiment

  current_average = self.sum / len(self.sentiments)
```

* Worker count: 1

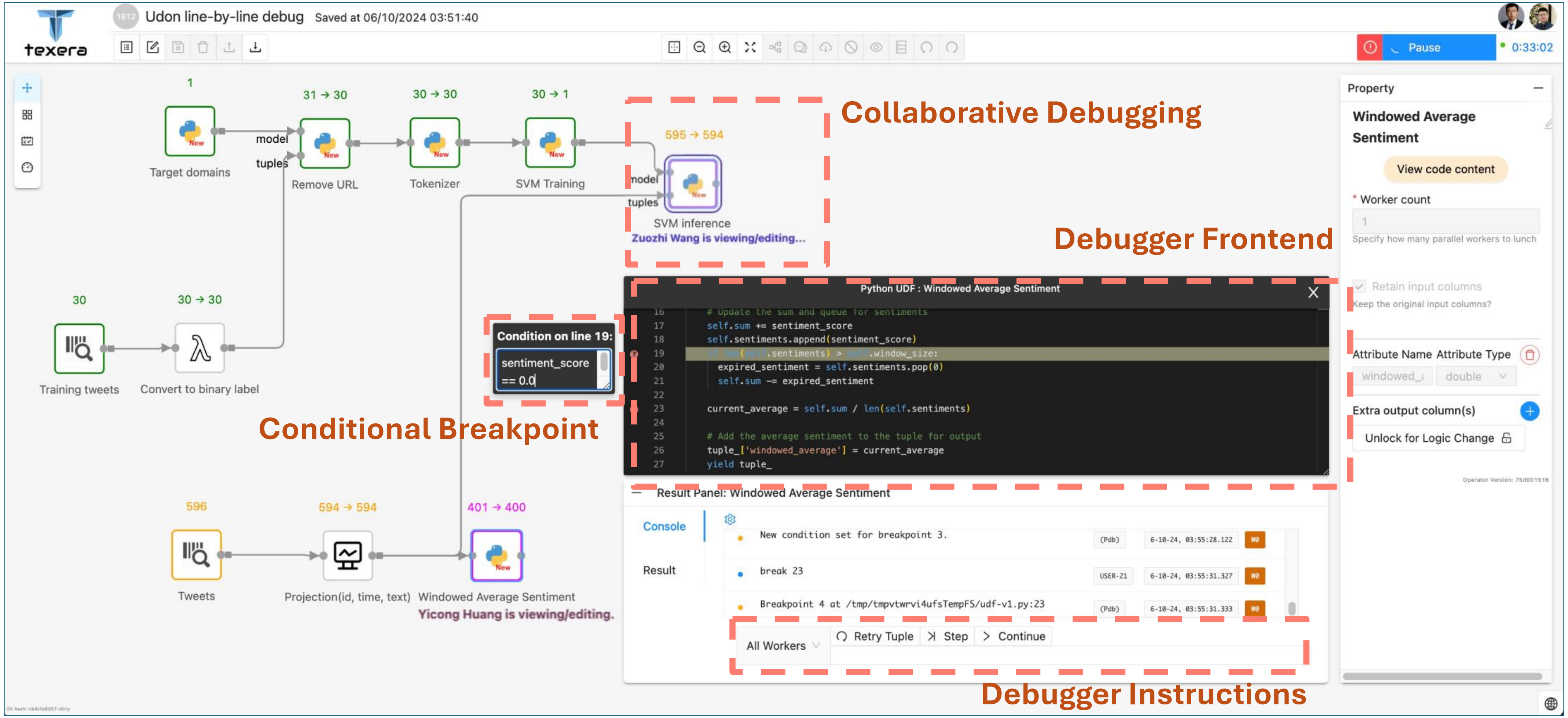
Specify how many parallel workers to lunch

☑ Retain input columns

Keep the original input columns?

Attribute Name	Attribute Type
windowed_a	double

Advanced Interaction – Debug



Condition on line 19:

sentiment_score == 0.0

Python UDF : Windowed Average Sentiment

```
16  # Update the sum and queue for sentiments
17  self.sum += sentiment_score
18  self.sentiments.append(sentiment_score)
19  if len(self.sentiments) > self.window_size:
20      expired_sentiment = self.sentiments.pop(0)
21      self.sum -= expired_sentiment
22
23  current_average = self.sum / len(self.sentiments)
24
25  # Add the average sentiment to the tuple for output
26  tuple_['windowed_average'] = current_average
27  yield tuple_
```

Result Panel: Windowed Average Sentiment

Console

Result

```
● New condition set for breakpoint 3.                      (Pdb)    6-10-24, 03:55:28.122  W0
● break 23                                                 USER-21  6-10-24, 03:55:31.327  W0
● Breakpoint 4 at /tmp/tmpvtwrvi4ufsTempFS/udf-v1.py:23    (Pdb)    6-10-24, 03:55:31.333  W0
```

All Workers ∨ | Retry Tuple | Step | Continue

Property

Windowed Average Sentiment

View code content

* Worker count

1

Specify how many parallel workers to lunch

☑ Retain input columns

Keep the original input columns?

Attribute Name Attribute Type

windowed_a | double

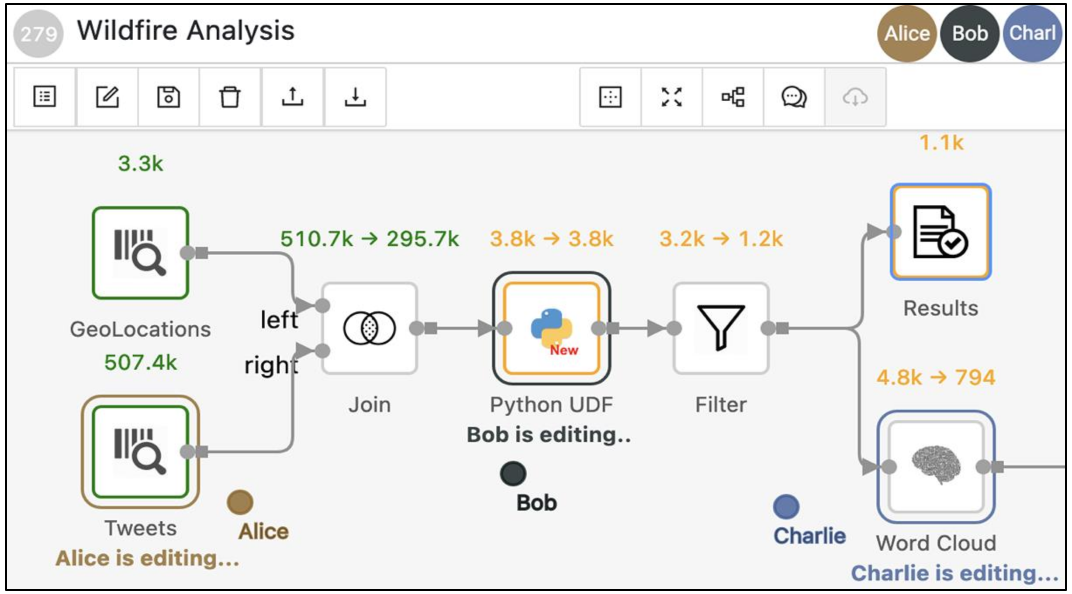
Extra output column(s)

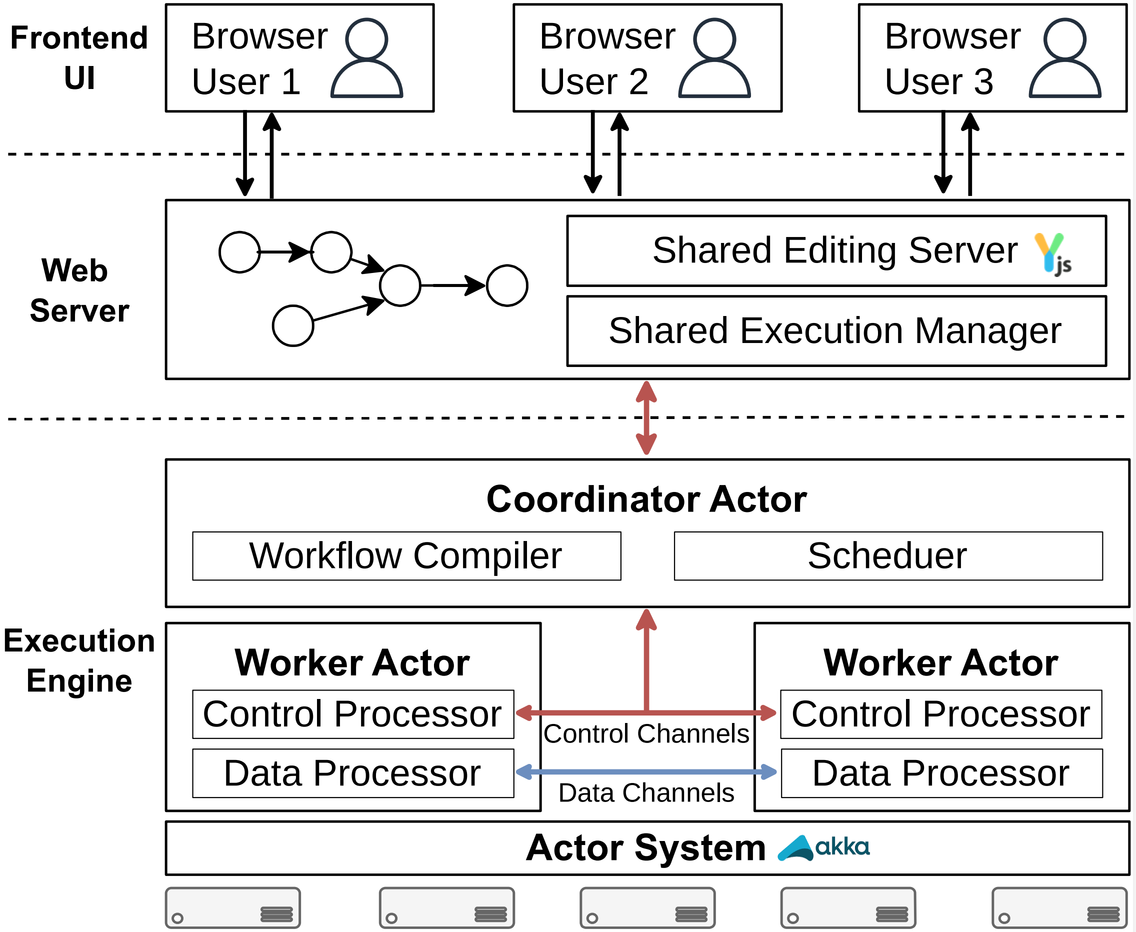
Unlock for Logic Change

Operator Version: 75d001516

How to Support Interactions?

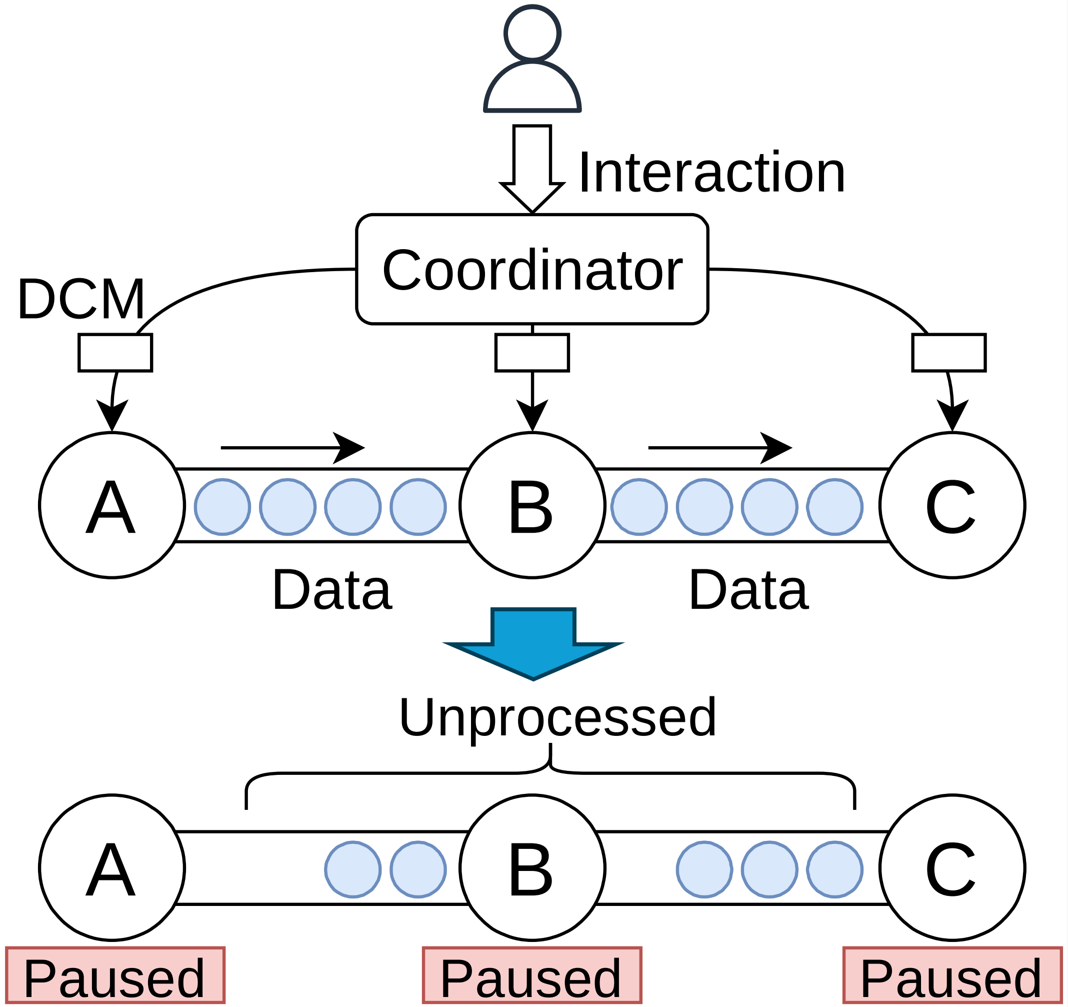
texera System Architecture

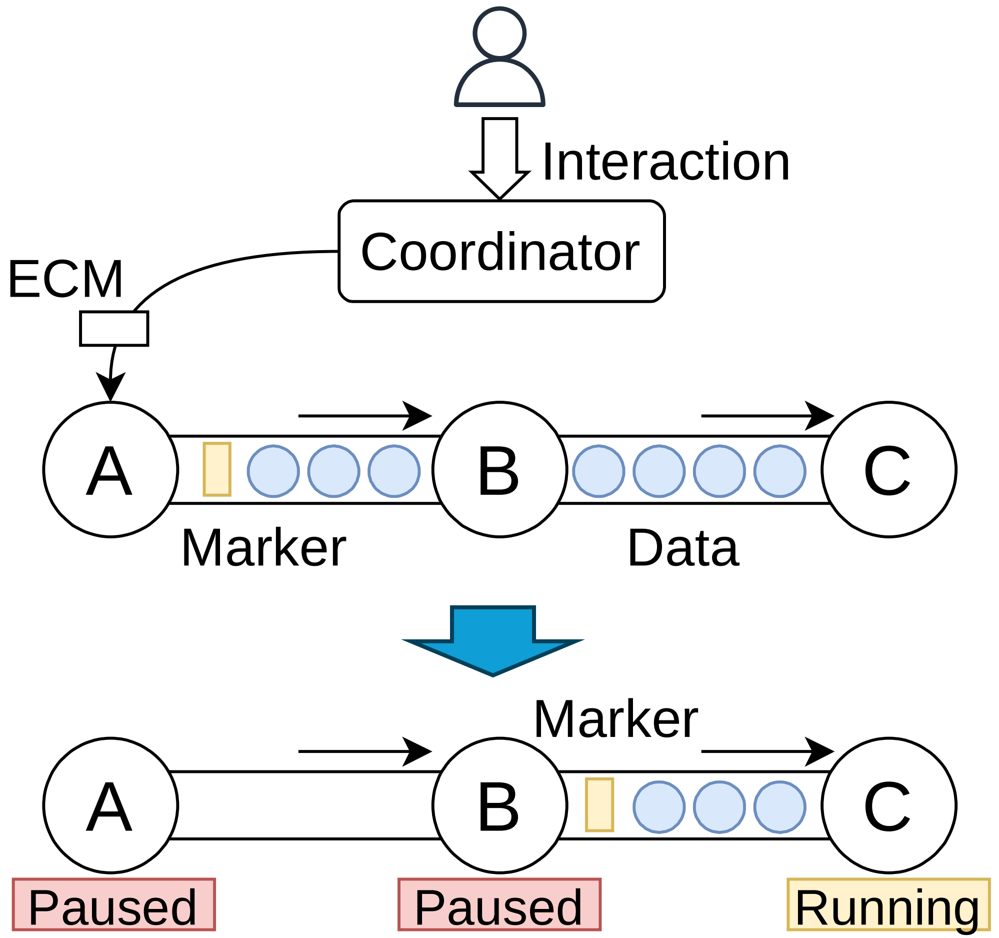




How to Support Pause?

Method 1: Pause a Workflow with **Direct** Control Message (DCM)

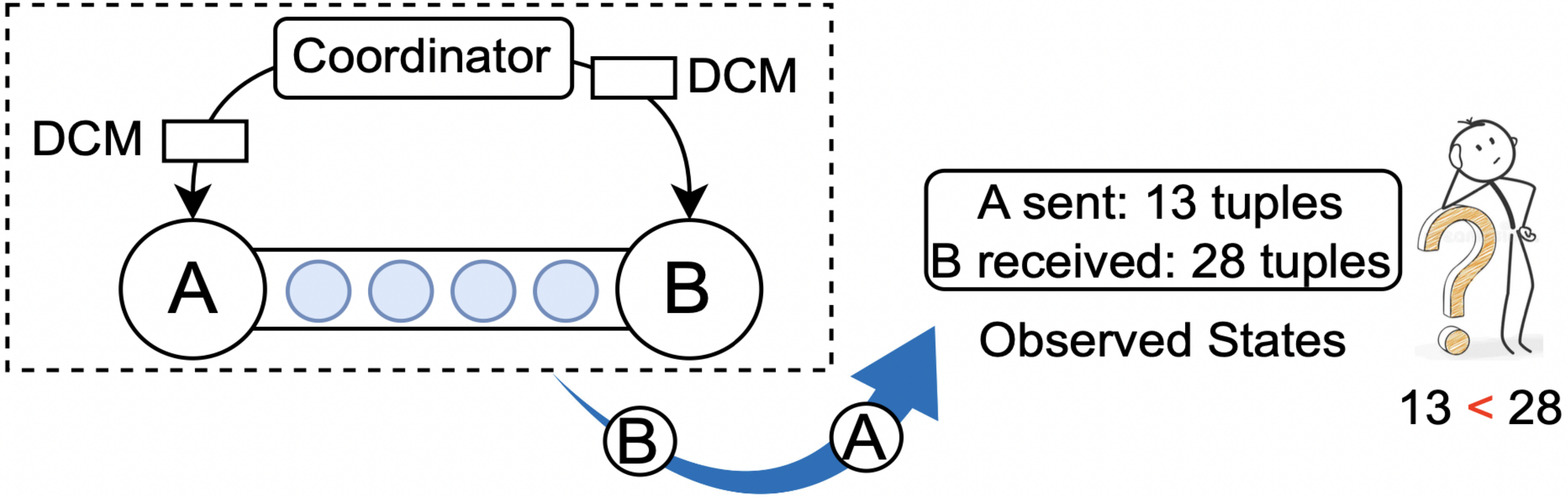




How to Support Pause?

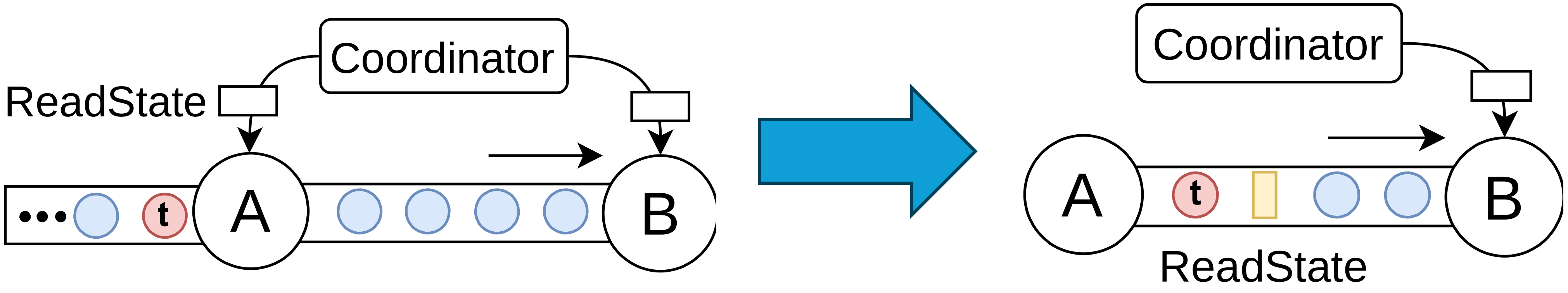
Method 2: Pause a Workflow with **Embedded** Control Message (ECM)

How to Support Read States?



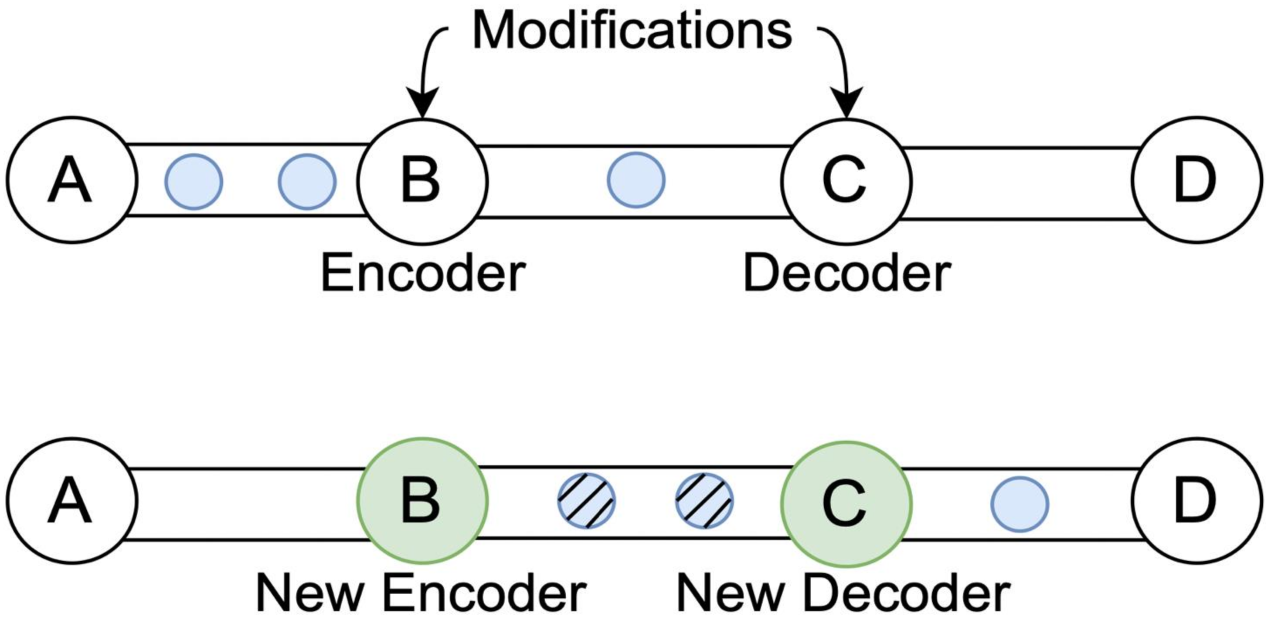
Incorrect global state introduced by DCM

Combining ECMs and DCMs to Read States



How to Support Modifications?

Strict Consistency



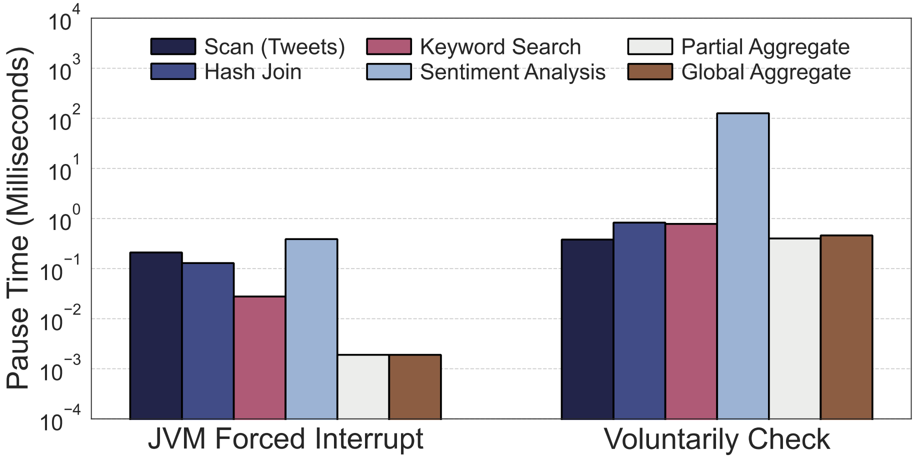
A tuple should be processed by the **same** version.

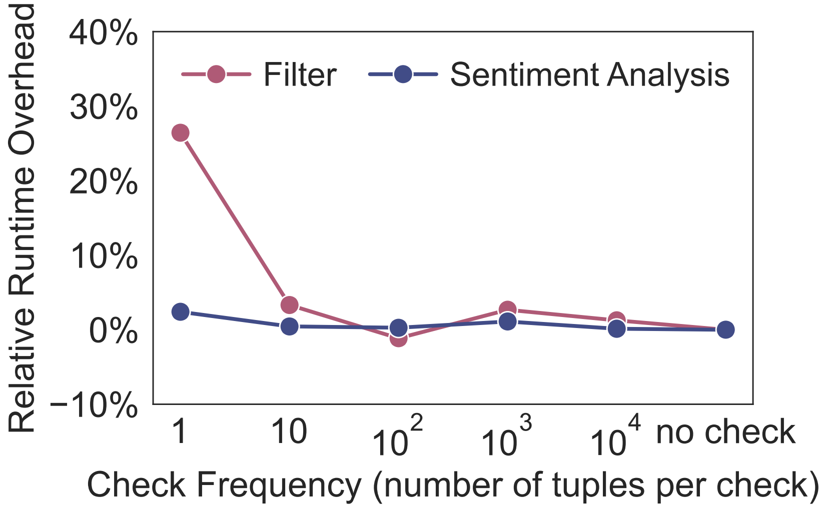
Experiments

Low Interaction Latency

Low Runtime Overhead

High Scalability





Summary of texera

WORKFLOW

COLLABORATION

INTERACTIONS

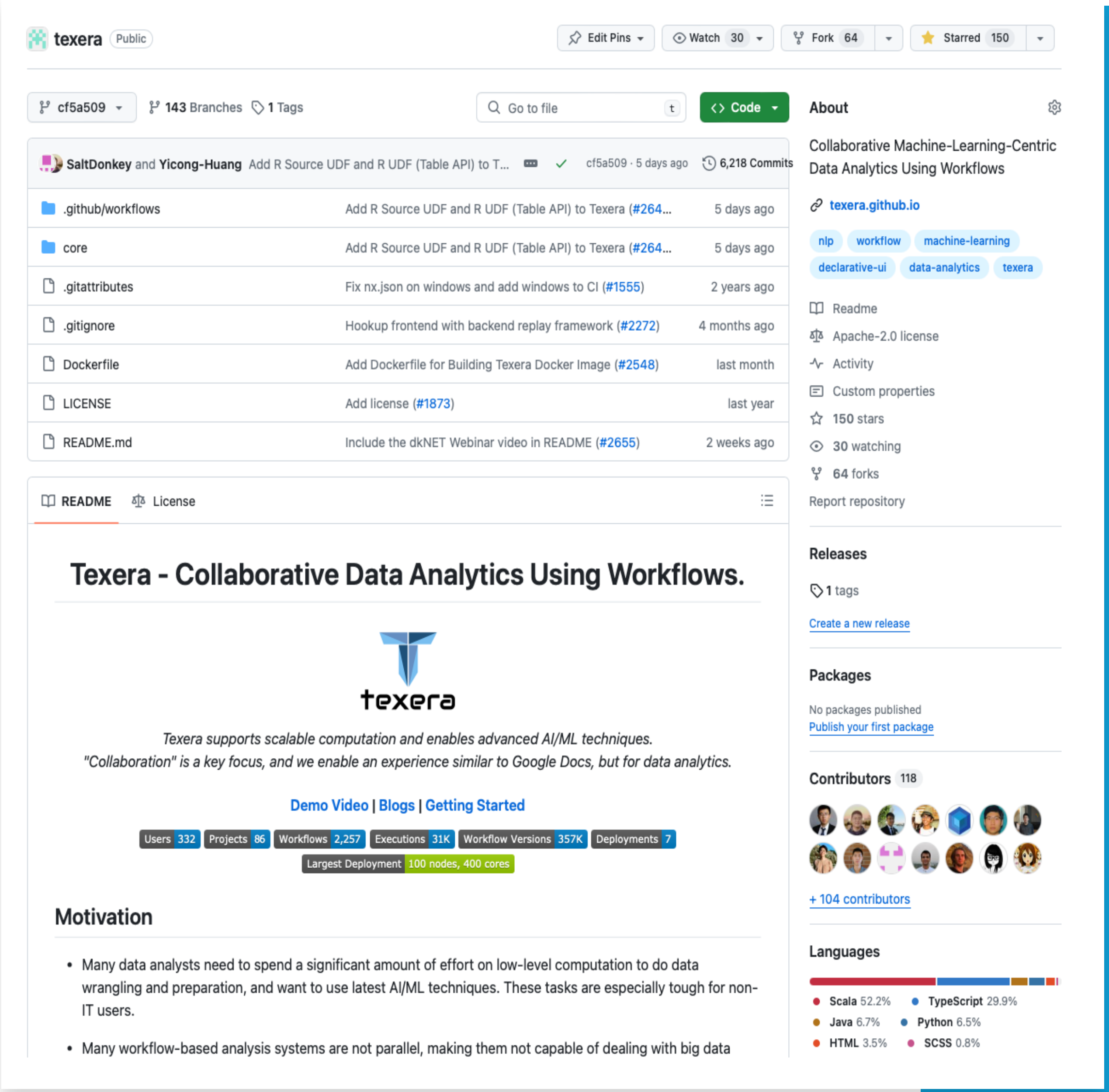
Texera Live Service

Texera GitHub Repo

Apache-2.0 License

Open Source

National Institute of Diabetes and Digestive and Kidney Diseases



texera Public

Edit Pins | Watch 30 | Fork 64 | Starred 150

cf5a509 | 143 Branches | 1 Tags | Go to file | Code

SaltDonkey and Yicong-Huang Add R Source UDF and R UDF (Table API) to T... ✓ cf5a509 · 5 days ago | 6,218 Commits

.github/workflows	Add R Source UDF and R UDF (Table API) to Texera (#264...	5 days ago
core	Add R Source UDF and R UDF (Table API) to Texera (#264...	5 days ago
.gitattributes	Fix nx.json on windows and add windows to CI (#1555)	2 years ago
.gitignore	Hookup frontend with backend replay framework (#2272)	4 months ago
Dockerfile	Add Dockerfile for Building Texera Docker Image (#2548)	last month
LICENSE	Add license (#1873)	last year
README.md	Include the dkNET Webinar video in README (#2655)	2 weeks ago

README | License

Texera - Collaborative Data Analytics Using Workflows.

texera

Texera supports scalable computation and enables advanced AI/ML techniques.
"Collaboration" is a key focus, and we enable an experience similar to Google Docs, but for data analytics.

Demo Video | Blogs | Getting Started

Users 332 | Projects 86 | Workflows 2,257 | Executions 31K | Workflow Versions 357K | Deployments 7

Largest Deployment 100 nodes, 400 cores

Motivation

- Many data analysts need to spend a significant amount of effort on low-level computation to do data wrangling and preparation, and want to use latest AI/ML techniques. These tasks are especially tough for non-IT users.
- Many workflow-based analysis systems are not parallel, making them not capable of dealing with big data

About

Collaborative Machine-Learning-Centric Data Analytics Using Workflows

texera.github.io

nlp | workflow | machine-learning | declarative-ui | data-analytics | texera

Readme
Apache-2.0 license
Activity
Custom properties
150 stars
30 watching
64 forks
Report repository

Releases

1 tags

Create a new release

Packages

No packages published
Publish your first package

Contributors 118

+ 104 contributors

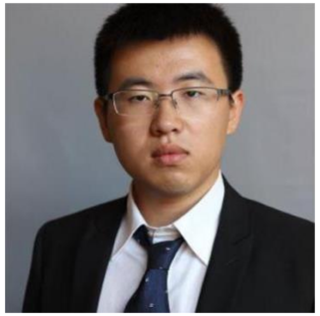
Languages

● Scala 52.2%	● TypeScript 29.9%
● Java 6.7%	● Python 6.5%
● HTML 3.5%	● SCSS 0.8%

Project and Usage Metrics

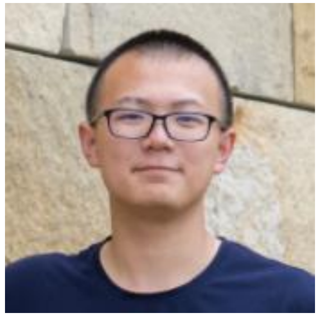
# of user accounts	332	# of projects	86
# of workflows	2,481	# of executions	50,950
# of workflow versions	357,000	# of publications	23
# of deployed servers	7	# of CPU cores in the largest deployment	400
# of files on GitHub	1,291	# of lines of code on GitHub	101,690
# of pull requests on GitHub	2,252	# of current PhD students	7
# of collaborating professors	17	# of involved undergraduates	80+
# of completed PhD theses	3	# of development years	8

Texera: A System for Collaborative and Interactive Data Analytics Using Workflows



Zuozhi Wang

Yicong Huang

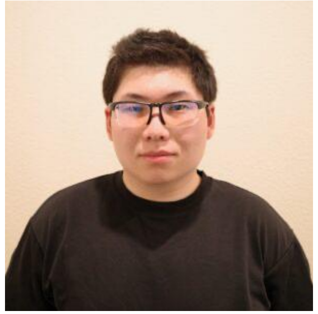


Shengquan Ni

Avinash Kumar

Sadeem Alsudais

Xiaozhen Liu



Xinyuan Lin

Yunyan Ding

Chen Li

Texera at https://texera.io

Texera Live Service

Texera GitHub Repo

Acknowledgements: This work was supported by NSF awards IIS-1745673 and IIS-2107150.

UCI University of California, Irvine