# Modeling Interactions in Product Reviews

## Utilizing Formal Argumentation to Assess Strengths and Weaknesses of Products Across Different Aspects

Ji Qi, Research Software Engineer, Social Science and Humanities Section, Netherlands eScience Center

Atefeh Keshavarzi Zafarghandi, Human-Centered Data Analysis, Centrum Wiskunde & Informatica

Davide Cielin, Human-Centered Data Analysis, Centrum Wiskunde & Informatica

j.qi@esciencecenter.nl, akz@cwi.nl, davide@cwi.nl

## Introduction

This study focuses on the creation of a valuable resource powered by formal argumentation for new users by effectively analyzing product reviews and highlighting the strengths and weaknesses associated with different aspects of a product.

Imagine that there are three reviews given for shoes:

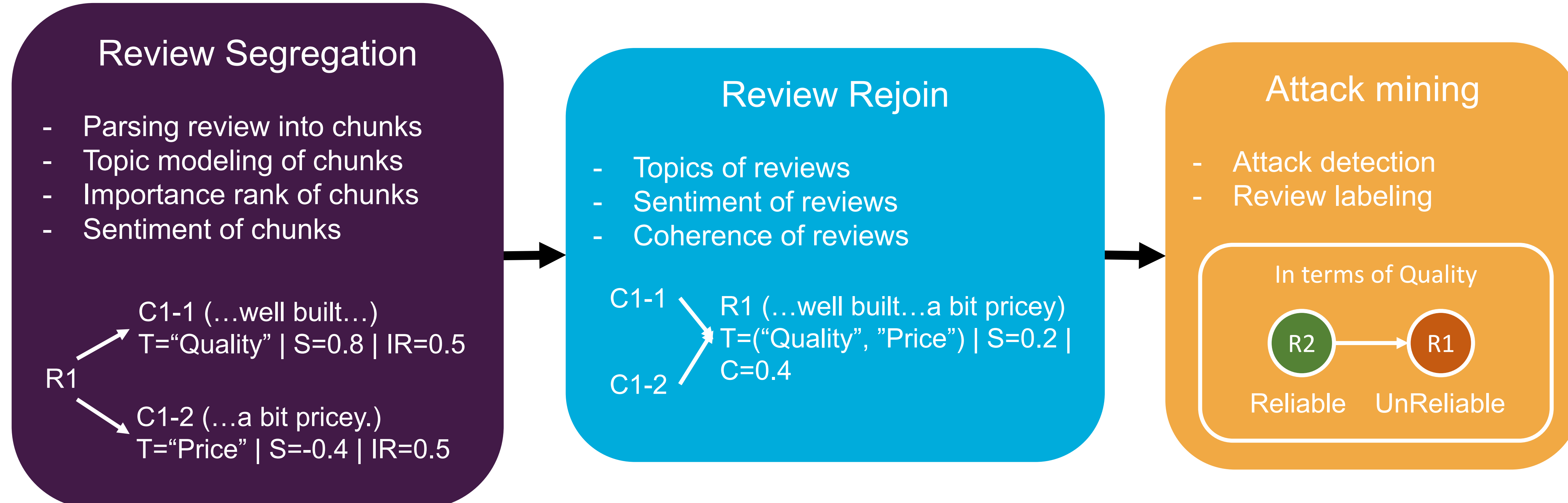☆★★★★ R1: *"They are always well built and last me quite a few years. The only downside is they tend to be a bit pricey."*

☆☆☆☆★ R2: *"My shoes are damaged in its warranty but the team has no response. Don't buy, very expensive."*

☆★★★★ R3: *"Largely my fault for not reading carefully, but it is very uncomfortable."*

They describe the shoes on different aspects, e.g., quality, price, service, and comfort. Here is the question we aim to answer: In case reviews conflict regarding one aspect (*well built* vs. *damaged in its warranty* regarding the quality of the shoes), which ones should we trust?
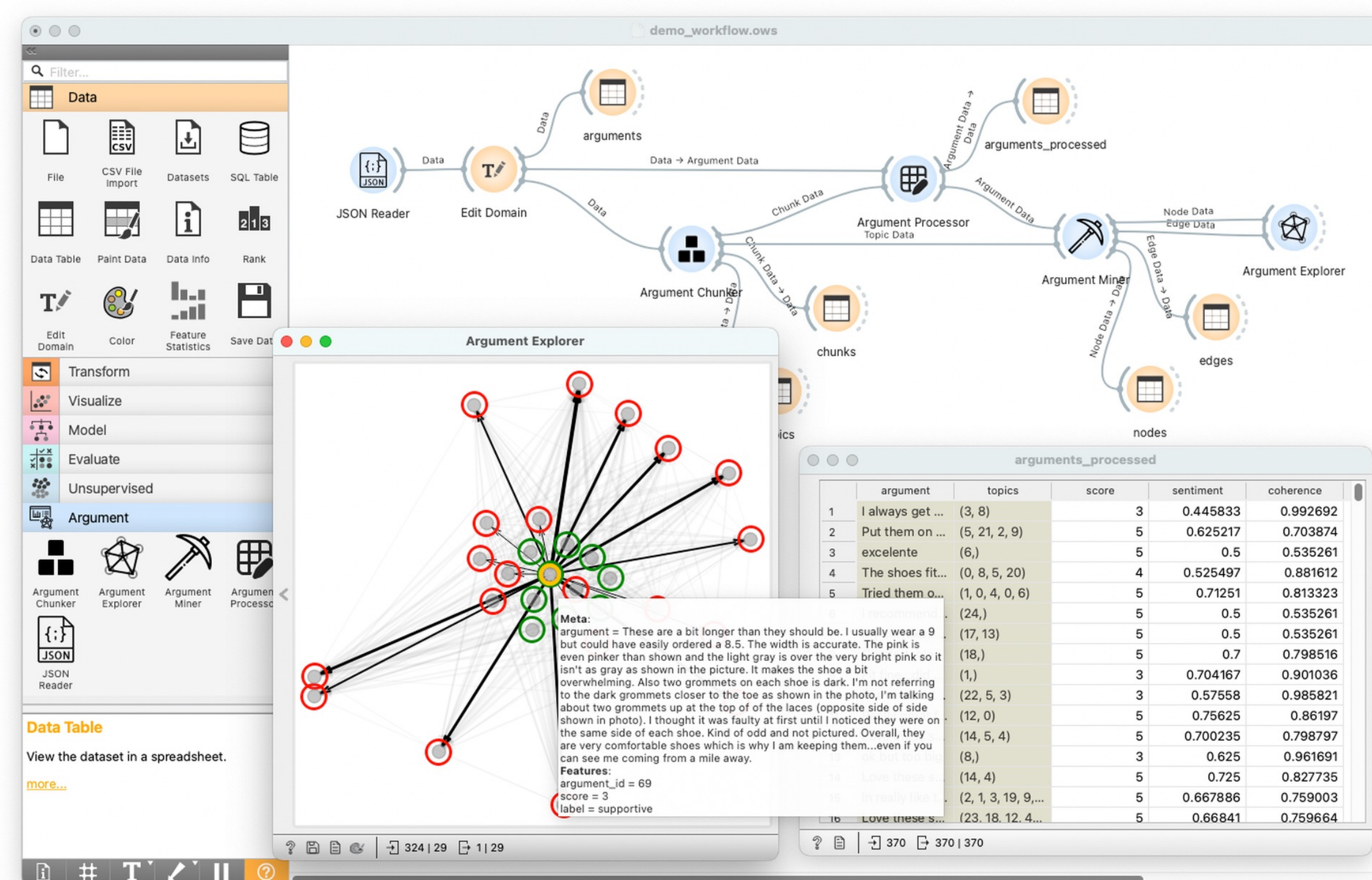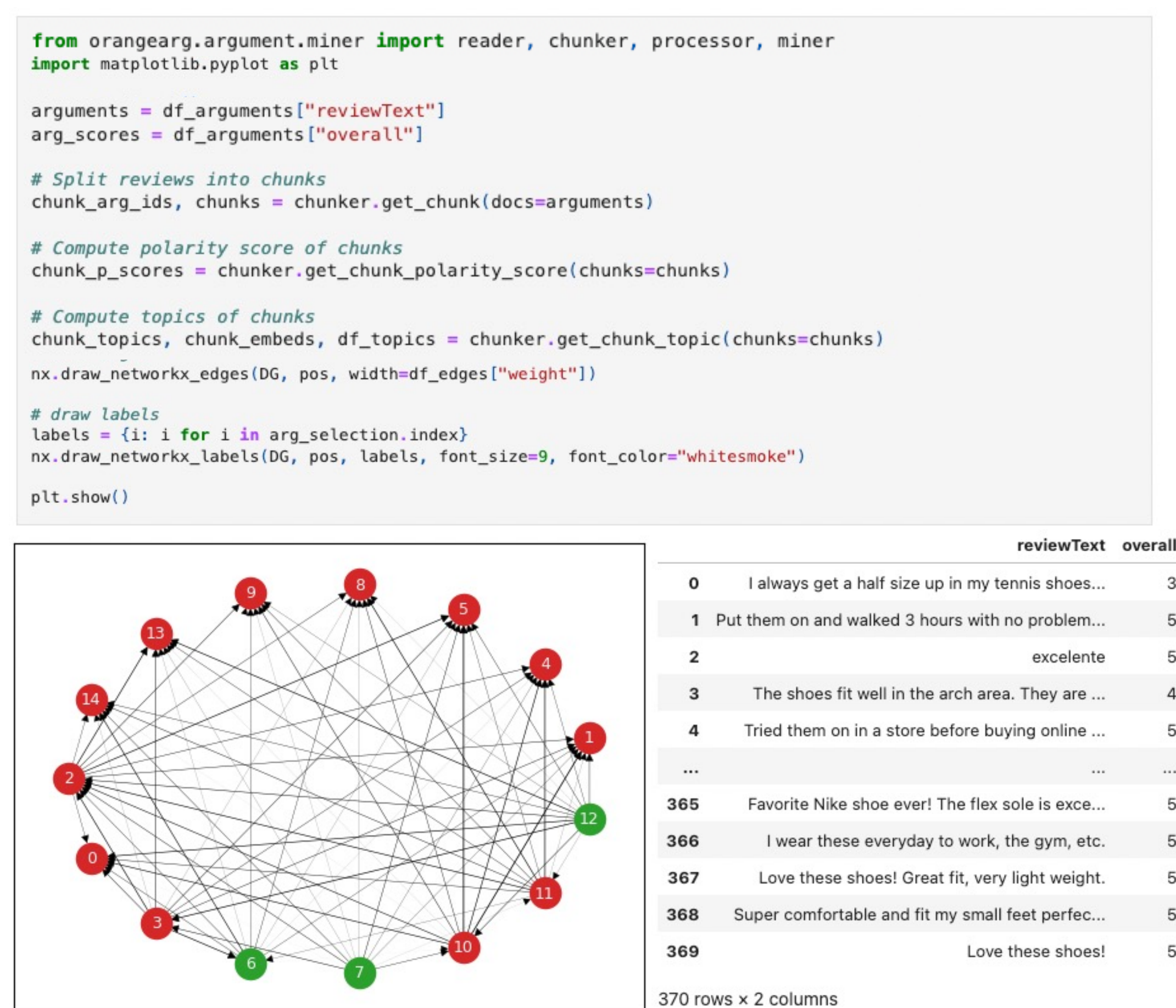
## Methods

We use **Abstract Argumentation Frameworks** to address the question in three steps:



### Review Segregation
- Parsing review into chunks
- Topic modeling of chunks
- Importance rank of chunks
- Sentiment of chunks

R1
→ C1-1 (…well built…) T="Quality" | S=0.8 | IR=0.5
→ C1-2 (…a bit pricey.) T="Price" | S=-0.4 | IR=0.5

### Review Rejoin
- Topics of reviews
- Sentiment of reviews
- Coherence of reviews

C1-1, C1-2 → R1 (…well built…a bit pricey) T=("Quality", "Price") | S=0.2 | C=0.4

### Attack mining
- Attack detection
- Review labeling

In terms of Quality
R2 (Reliable) → R1 (UnReliable)

## Implementation

A Python package is developed to implement the pipeline, as well as GUIs for user-friendly interaction.

```python
from orangearg.argument.miner import reader, chunker, processor, miner
import matplotlib.pyplot as plt

arguments = df_arguments["reviewText"]
arg_scores = df_arguments["overall"]

# Split reviews into chunks
chunk_arg_ids, chunks = chunker.get_chunk(docs=arguments)

# Compute polarity score of chunks
chunk_p_scores = chunker.get_chunk_polarity_score(chunks=chunks)

# Compute topics of chunks
chunk_topics, chunk_embeds, df_topics = chunker.get_chunk_topic(chunks=chunks)
nx.draw_networkx_edges(DG, pos, width=df_edges["weight"])

# draw labels
labels = {i: i for i in arg_selection.index}
nx.draw_networkx_labels(DG, pos, labels, font_size=9, font_color="whitesmoke")

plt.show()
```



|  | reviewText | overall |
|---|---|---|
| 0 | I always get a half size up in my tennis shoes… | 3 |
| 1 | Put them on and walked 3 hours with no problem… | 5 |
| 2 | excelente | 5 |
| 3 | The shoes fit well in the arch area. They are … | 4 |
| 4 | Tried them on in a store before buying online … | 5 |
| … | … | … |
| 365 | Favorite Nike shoe ever! The flex sole is exce… | 5 |
| 366 | I wear these everyday to work, the gym, etc. | 5 |
| 367 | Love these shoes! Great fit, very light weight. | 5 |
| 368 | Super comfortable and fit my small feet perfe… | 5 |
| 369 | Love these shoes! | 5 |

370 rows × 2 columns



Find our work on  GitHub
https://github.com/EyeofBeholder-NLeSC/orange3-argument

esciencecenter.nl        **Powered by pioneering software**