Data extraction from Wine Reviews

Self Project
Time period: Aug 2021 – WIP

Agenda

- Overview
- Data chosen
- Approach
- User Interface – Screenshots
  - New data input
  - Querying the database
- Further improvements

Overview

- Process unstructured data and graph a network
- Allow query running via user interface
- Created two python scripts:
  - Extract data from original source into text files
  - Process the text files to load graph and show GUI

Data used

- As unstructured data decided to use wine reviews as I have already worked in the alco-bev industry earlier:
  - Searched for “wine reviews data” and found kaggle data
  - Link: https://www.kaggle.com/zynicide/wine-reviews
- About the data:
  - CSV file with 130k rows and 14 columns
  - Used only “Description column” as unstructured data

Data Extraction

- Loaded to Pandas
- Accessed first 10k rows
- Content of “Description” cell written to individual text file
  - Files named as fxxxx.txt, where ‘xxxx’ is from 0000 to 9999
  - First 5 files kept aside for “new user data” input
- Script name: 01_create_data_1.py

- Contents of some random file (f0028.txt):
  - *Aromas recall ripe dark berry, toast and a whiff of cake spice. The soft, informal palate offers sour cherry, vanilla and a hint of espresso alongside round tannins. Drink soon.*

Approach

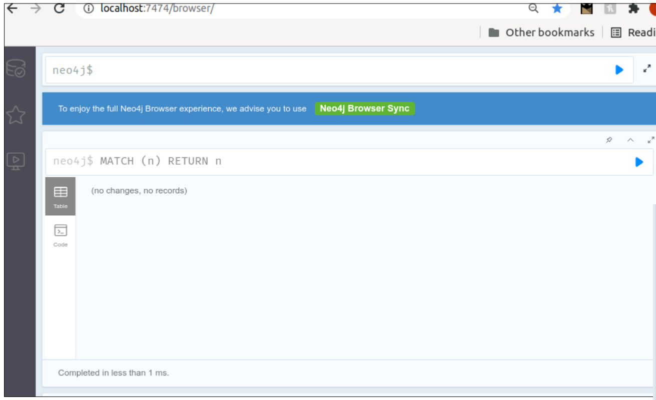
- Performed data extraction (covered earlier)
- Feature extraction with Spacy (version 3.1.1) large model
- Saved features in custom data structure to intermediate file - .json type
- Features Extracted:
  - Word count, sentence count, sentiment score
  - Raw text and processed text
    - ✓ Lemmatization, removal of stop-words and punctuations
- Named-entity-recognition (NER) extraction

Approach

- Inserted data to Neo4j graph using the intermediate json file
  - Flag (RELOAD_TO_NEO) to allow processing of specified input files and load to graph (LIMIT_UPLOAD_TO_NEO)
- GUI implemented with Tkinter
  - 3 pre-set queries with user-defined input parameters
    - Query 1: Find count of nodes of a certain Label type
    - Query 2: Find count of Review type nodes whose raw text is longer than minimum specified word count, and the sentiment score is greater than minimum specified score
    - Query 3: Show Review nodes “that have flavors” as specified by user input
  - New input file path for processing new data and Neo4j load

GUI Screenshots

Empty graph before initial insertion



```
neo4j$
```

To enjoy the full Neo4j Browser experience, we advise you to use Neo4j Browser Sync

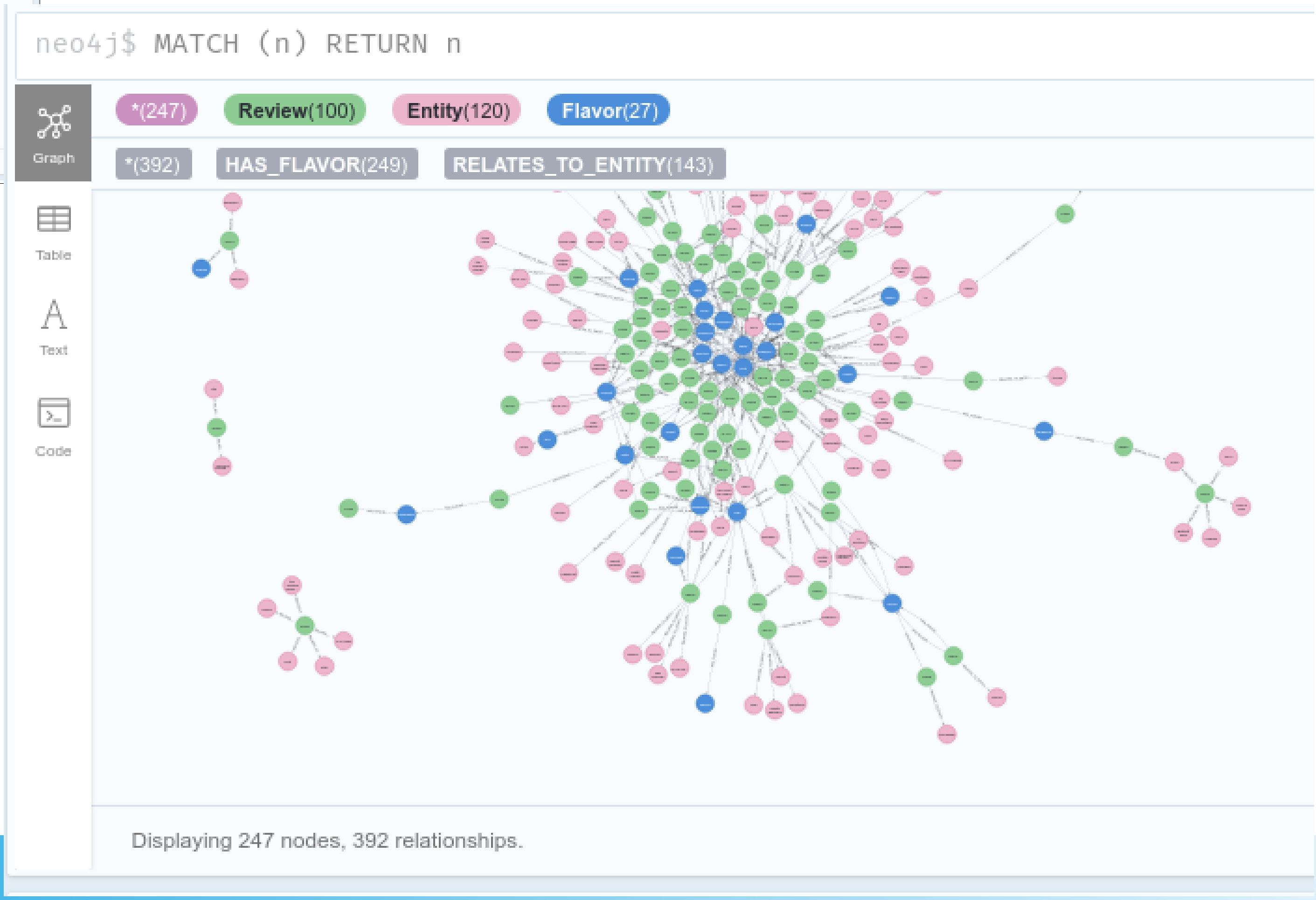
```
neo4j$ MATCH (n) RETURN n
```

(no changes, no records)

Completed in less than 1 ms.

Neo4j graph after initial insertion from 100 files.

247 nodes, 392 relationships.



```
neo4j$ MATCH (n) RETURN n
```

Displaying 247 nodes, 392 relationships.

Ran script:
02_read_process_for_neo_3.py

with RELOAD_TO_NEO = True

LIMIT_UPLOAD_TO_NEO = 100

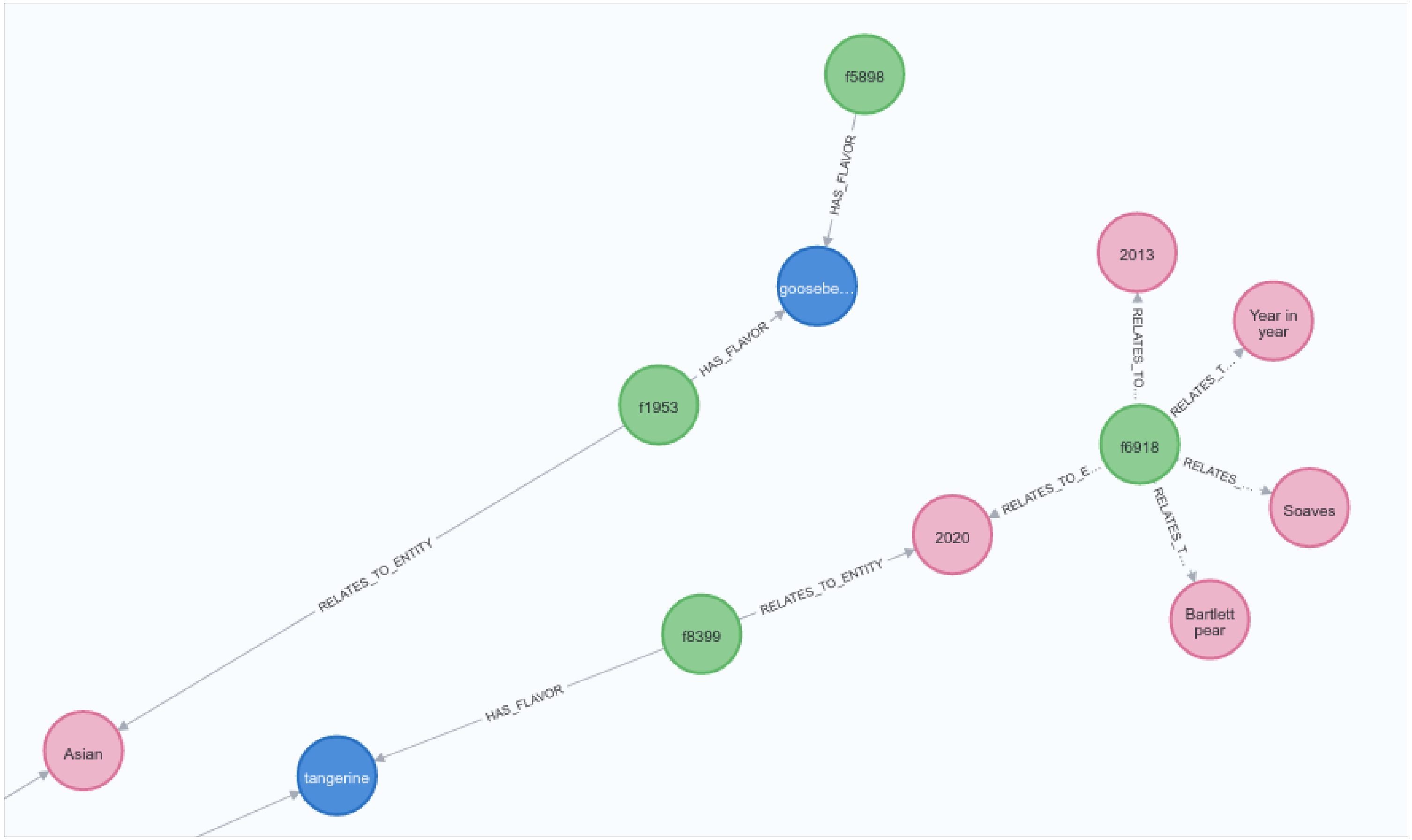
GUI Screenshots

Neo4j graph nodes and relationships:

- ✔ (REVIEW Node) - HAS_FLAVOR -> (FLAVOR node)
- ✔ (REVIEW Node) - RELATES_TO_ENTITY -> (ENTITY node)

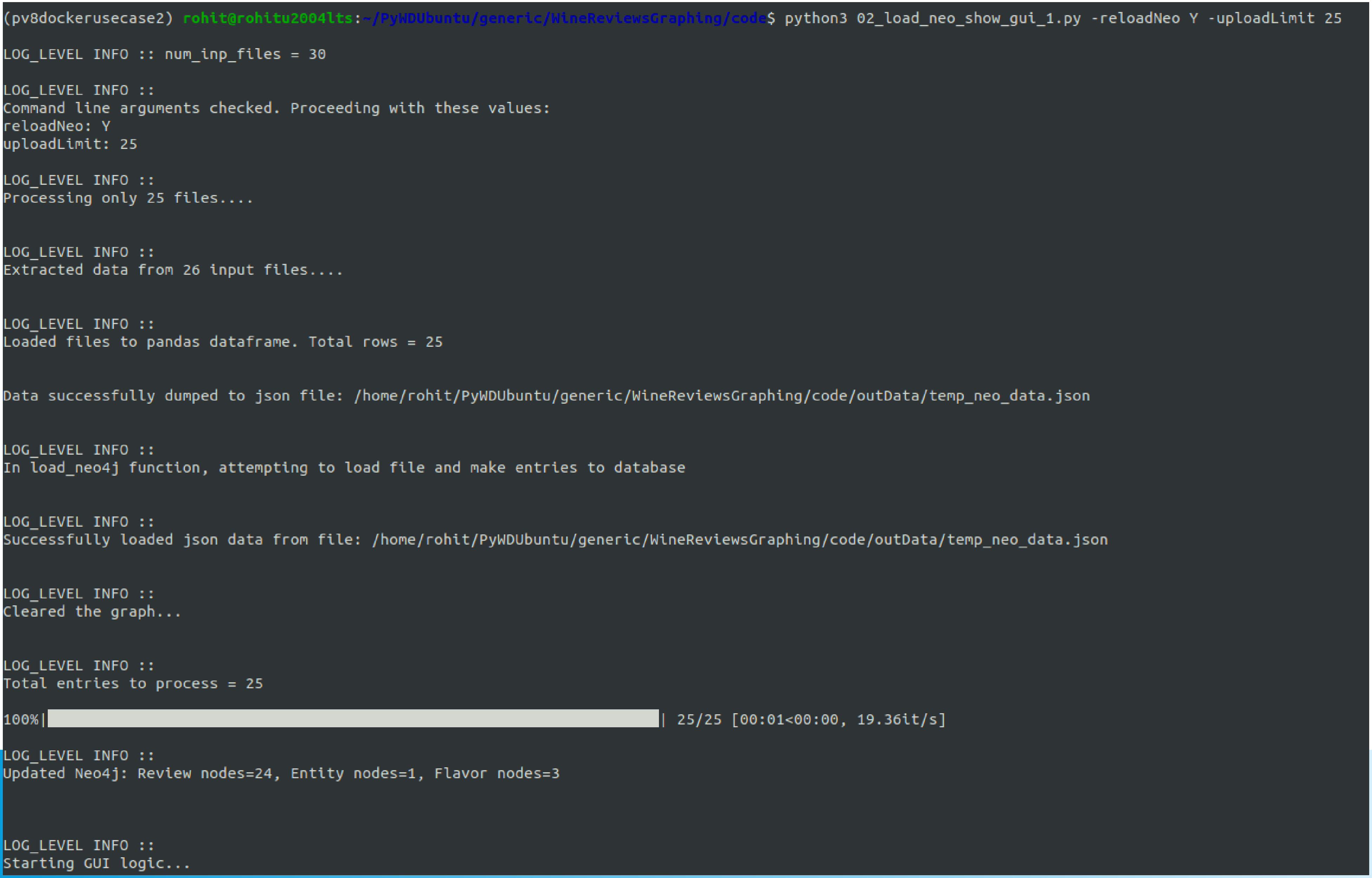
Properties of Graph:

- Review Node (*green*): filename, sentiment score, word count, sentence count
- Entity Node (*pink*): text, label code, label name. E.g. name=2020, label=391, label_=DATE
- Flavor Node (*blue*): name. E.g. name=cherry



GUI Screenshots

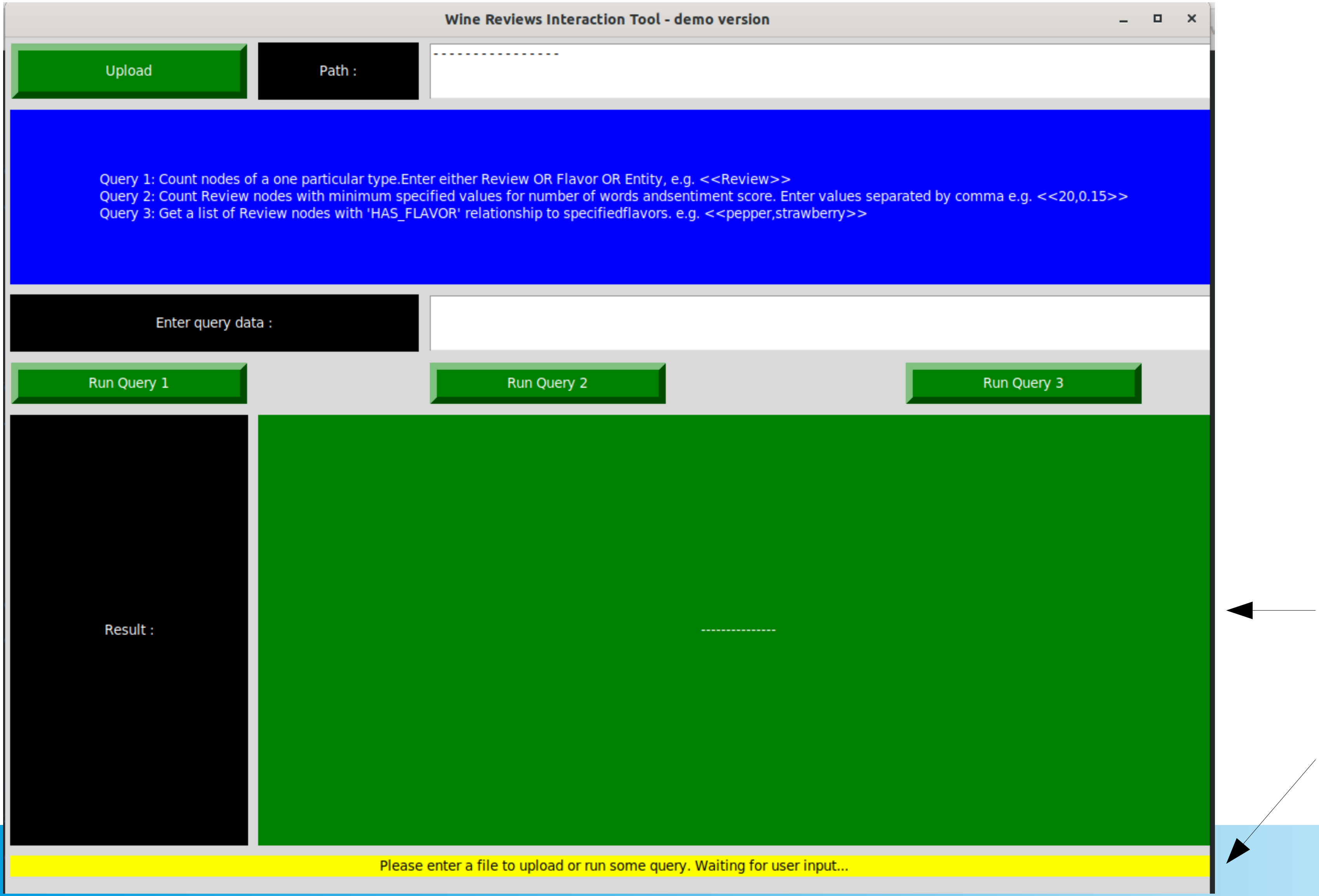
Neo4j graph after initial insertion:
Console ouput



```
(pv8dockerusecase2) rohit@rohitu2004lts:~/PyWDUbuntu/generic/WineReviewsGraphing/code$ python3 02_load_neo_show_gui_1.py -reloadNeo Y -uploadLimit 25

LOG_LEVEL INFO :: num_inp_files = 30

LOG_LEVEL INFO ::
Command line arguments checked. Proceeding with these values:
reloadNeo: Y
uploadLimit: 25

LOG_LEVEL INFO ::
Processing only 25 files....


LOG_LEVEL INFO ::
Extracted data from 26 input files....


LOG_LEVEL INFO ::
Loaded files to pandas dataframe. Total rows = 25


Data successfully dumped to json file: /home/rohit/PyWDUbuntu/generic/WineReviewsGraphing/code/outData/temp_neo_data.json


LOG_LEVEL INFO ::
In load_neo4j function, attempting to load file and make entries to database


LOG_LEVEL INFO ::
Successfully loaded json data from file: /home/rohit/PyWDUbuntu/generic/WineReviewsGraphing/code/outData/temp_neo_data.json


LOG_LEVEL INFO ::
Cleared the graph...


LOG_LEVEL INFO ::
Total entries to process = 25

100%|██████████████████████████████████████████████████████████████████| 25/25 [00:01<00:00, 19.36it/s]

LOG_LEVEL INFO ::
Updated Neo4j: Review nodes=24, Entity nodes=1, Flavor nodes=3



LOG_LEVEL INFO ::
Starting GUI logic...
```

GUI Screenshots

Initial Window



Wine Reviews Interaction Tool - demo version

Query 1: Count nodes of a one particular type.Enter either Review OR Flavor OR Entity, e.g. <<Review>>
Query 2: Count Review nodes with minimum specified values for number of words andsentiment score. Enter values separated by comma e.g. <<20,0.15>>
Query 3: Get a list of Review nodes with 'HAS_FLAVOR' relationship to specifiedflavors. e.g. <<pepper,strawberry>>

Please enter a file to upload or run some query. Waiting for user input...

Upload new data with file path

Instructions for Queries

Query data input area and submit buttons

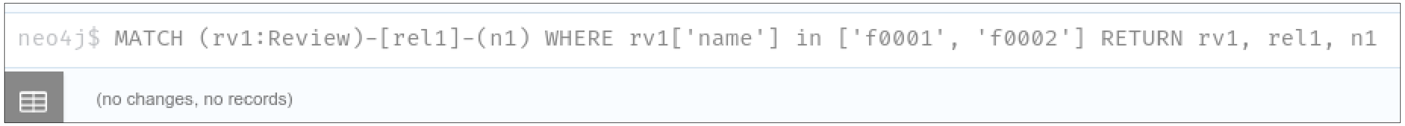
Result area

Status message

GUI Screenshots

Adding new file for processing

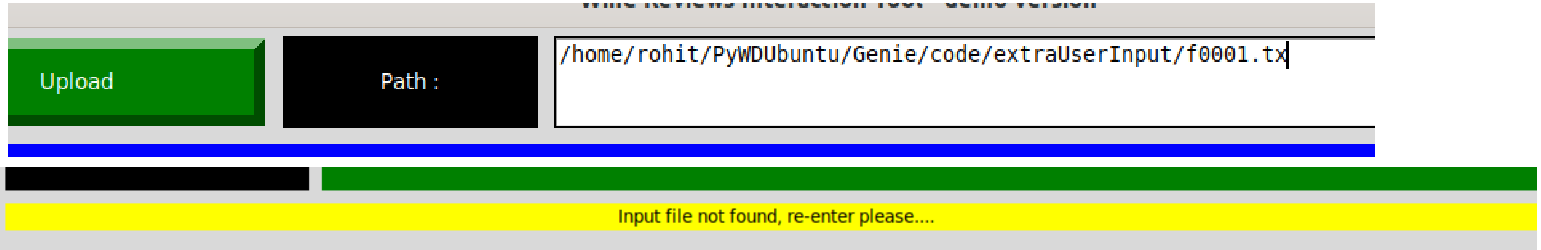
- Initially this query returns no hits: as files f0001.txt and f0002.txt are NOT yet processed



```
neo4j$ MATCH (rv1:Review)-[rel1]-(n1) WHERE rv1['name'] in ['f0001', 'f0002'] RETURN rv1, rel1, n1
```

(no changes, no records)

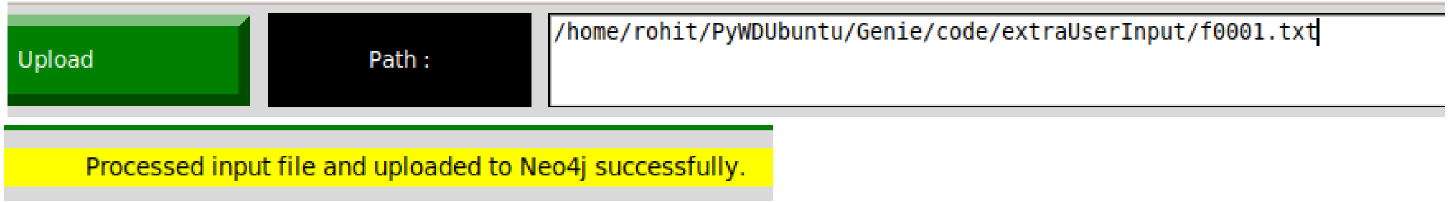
- Incorrect file entered: status message shows file not found



/home/rohit/PyWDUbuntu/Genie/code/extraUserInput/f0001.tx

Input file not found, re-enter please....

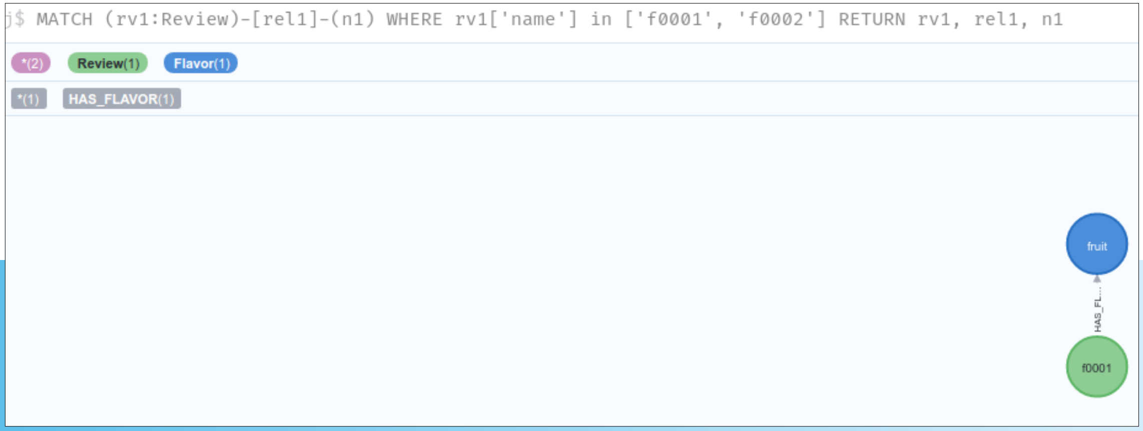
- Correct file specified now – processed successfully



/home/rohit/PyWDUbuntu/Genie/code/extraUserInput/f0001.txt

Processed input file and uploaded to Neo4j successfully.

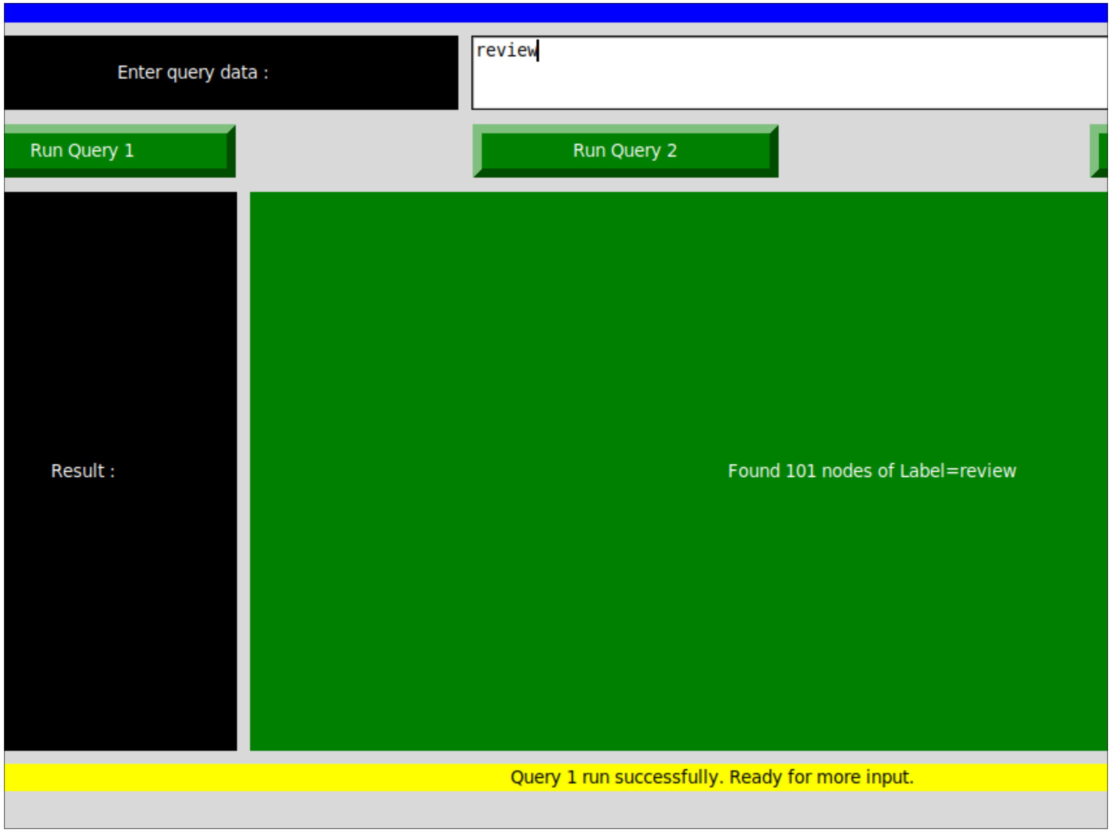
- Same query now returns hit in Neo4j



```
j$ MATCH (rv1:Review)-[rel1]-(n1) WHERE rv1['name'] in ['f0001', 'f0002'] RETURN rv1, rel1, n1
```

GUI Screenshots

Query 1: Count of particular node e.g. Review node



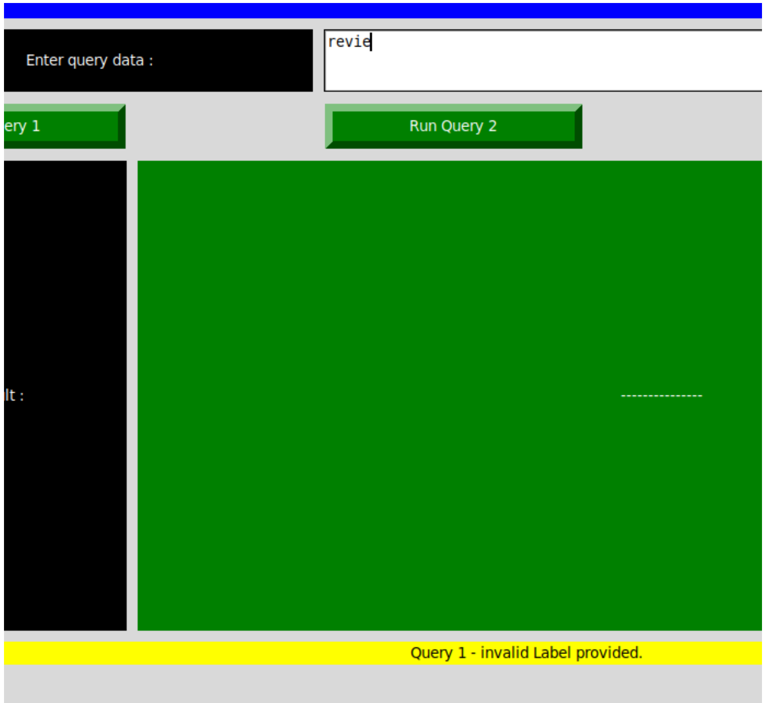
Enter query data : review

Result :

Found 101 nodes of Label=review

Query 1 run successfully. Ready for more input.

Invalid label – appropriate status message



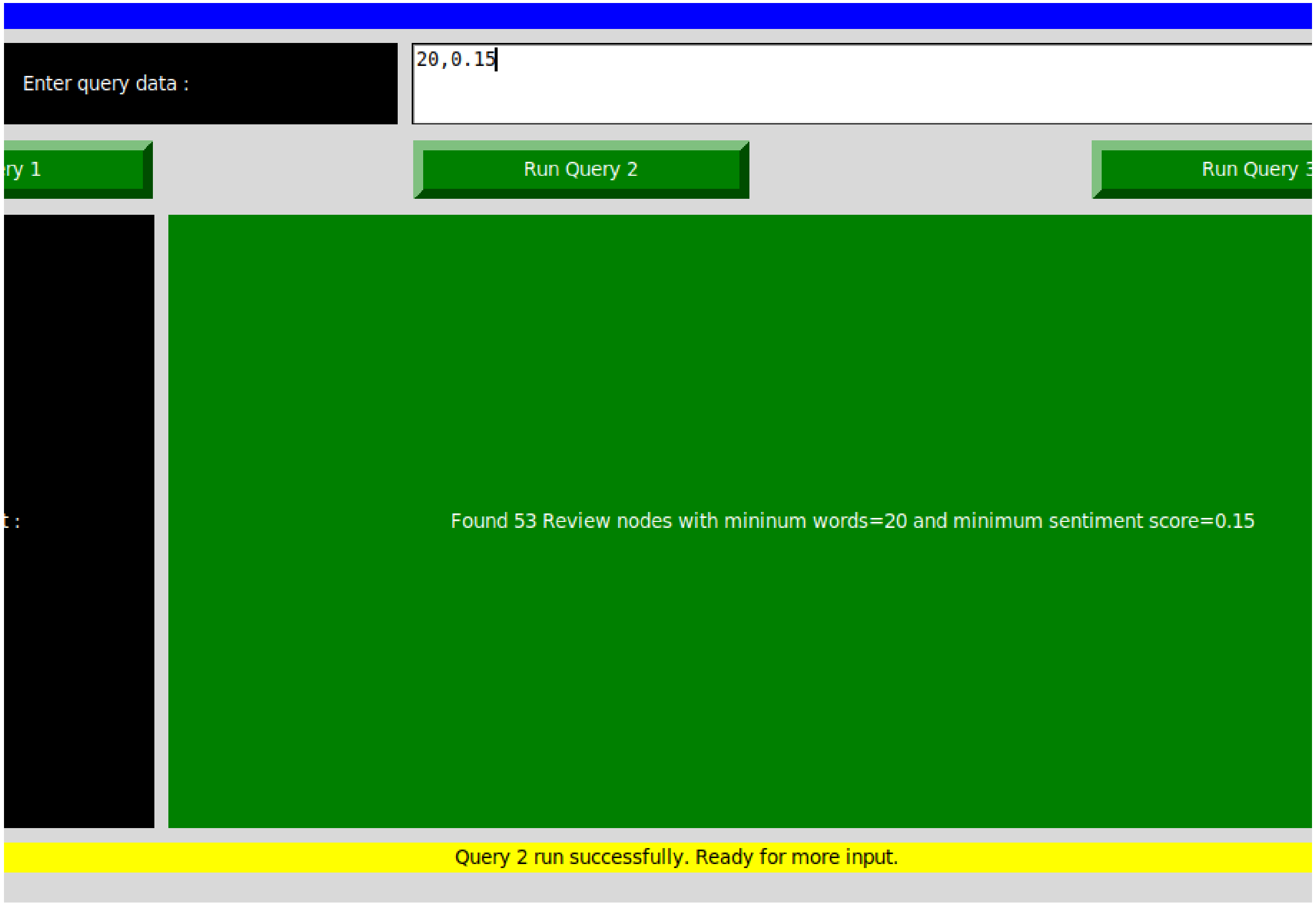
Enter query data : revie

lt :

Query 1 - invalid Label provided.

GUI Screenshots

Query 2: Count Review nodes with minimum 20 words and sentiment score of 0.13

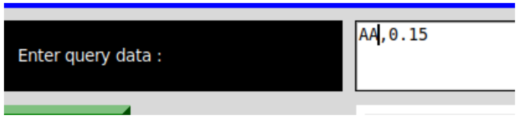


Enter query data : 20,0.15

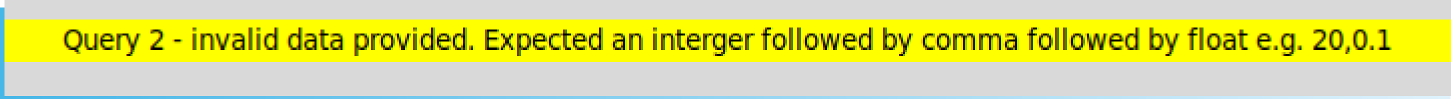
Found 53 Review nodes with mininum words=20 and minimum sentiment score=0.15

Query 2 run successfully. Ready for more input.

Invalid input – appropriate status message: entered AA,0.15



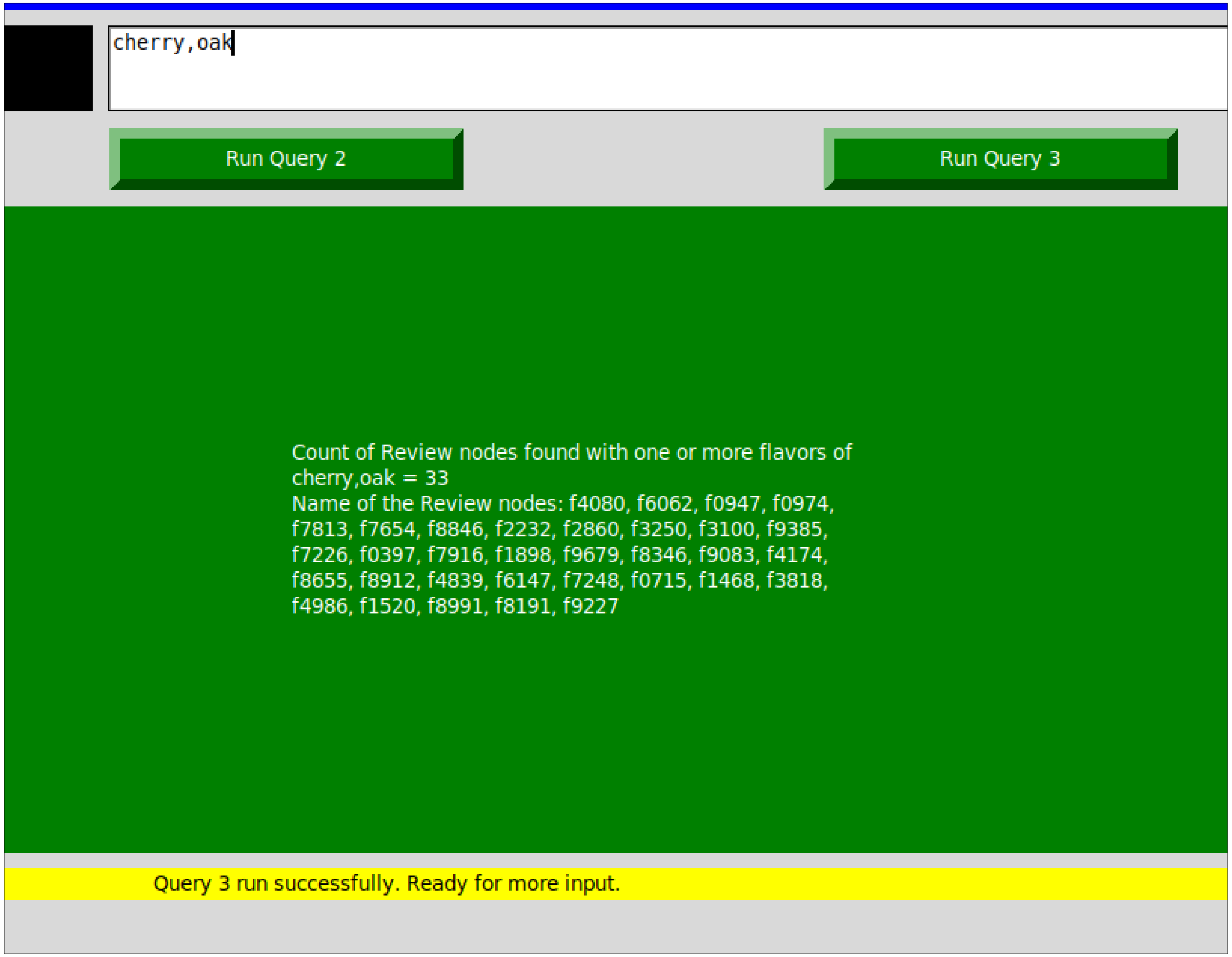
Enter query data : AA,0.15



Query 2 - invalid data provided. Expected an interger followed by comma followed by float e.g. 20,0.1

GUI Screenshots

Query 3: Count and show Review nodes with review having specified flavors



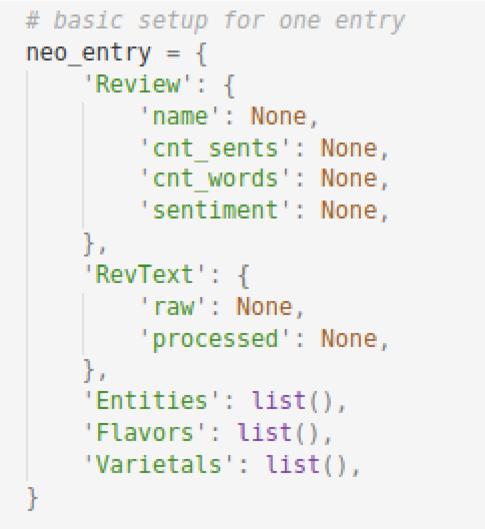
cherry,oak

Count of Review nodes found with one or more flavors of cherry,oak = 33
Name of the Review nodes: f4080, f6062, f0947, f0974, f7813, f7654, f8846, f2232, f2860, f3250, f3100, f9385, f7226, f0397, f7916, f1898, f9679, f8346, f9083, f4174, f8655, f8912, f4839, f6147, f7248, f0715, f1468, f3818, f4986, f1520, f8991, f8191, f9227

Query 3 run successfully. Ready for more input.

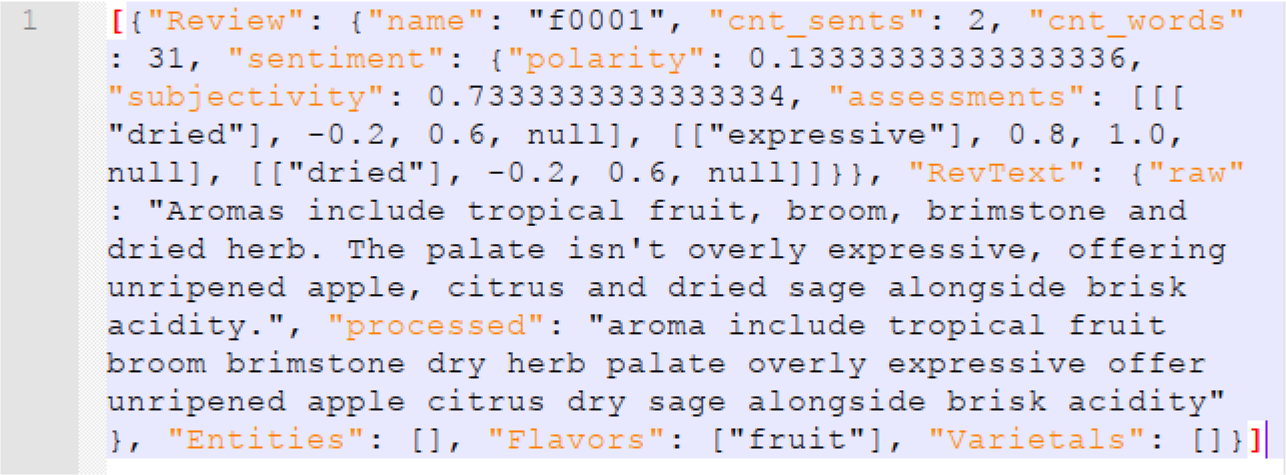
Code Snippet

Custom data
structure to store
features



```
# basic setup for one entry
neo_entry = {
    'Review': {
        'name': None,
        'cnt_sents': None,
        'cnt_words': None,
        'sentiment': None,
    },
    'RevText': {
        'raw': None,
        'processed': None,
    },
    'Entities': list(),
    'Flavors': list(),
    'Varietals': list(),
}
```

Intermediate Json
file contents after
processing f0001.txt



```
[{"Review": {"name": "f0001", "cnt_sents": 2, "cnt_words"
: 31, "sentiment": {"polarity": 0.13333333333333336,
"subjectivity": 0.7333333333333334, "assessments": [[[
"dried"], -0.2, 0.6, null], [["expressive"], 0.8, 1.0,
null], [["dried"], -0.2, 0.6, null]]}}, "RevText": {"raw"
: "Aromas include tropical fruit, broom, brimstone and
dried herb. The palate isn't overly expressive, offering
unripened apple, citrus and dried sage alongside brisk
acidity.", "processed": "aroma include tropical fruit
broom brimstone dry herb palate overly expressive offer
unripened apple citrus dry sage alongside brisk acidity"
}, "Entities": [], "Flavors": ["fruit"], "Varietals": []}]
```

Improvements

- For Web application - use Flask or Django instead of Tkinter
- Allow custom Cypher query instead of pre-set queries
- Add wine varietals as a new category in NER processing
  - Will allow Relationship like WINE_TYPE
- Topic modeling to find related reviews
- Allow free typing of input from user as new review to process