Guide

ER project

1. Clone the code

```
git clone https://github.com/Mzhongwei/er_embedding_streaming.git
```

you will get code with the following structure:

```
|----config/
|    |------default/     # Default configuration settings
|    |------examples/    # Example configuration files
|----dataprocessing/     # All data processing methods, receive data by
kafka consumer
|----dynamic_embedding/  # Core methods for dynamic embedding
|----Data_example/       # Example datasets for testing
|----utils/              # Utility functions and helper scripts
|----main.py
|----requirements.txt
|----README.md
```

2. Create a virtual environment, activate and install dependencies:

```
python3 -m venv venv
source venv/bin/activate

pip install -r requirements.txt
pip install git+https://github.com/dpkp/kafka-python.git
```

3. Execute the Pre-training process:

3.1. Set configuration variables

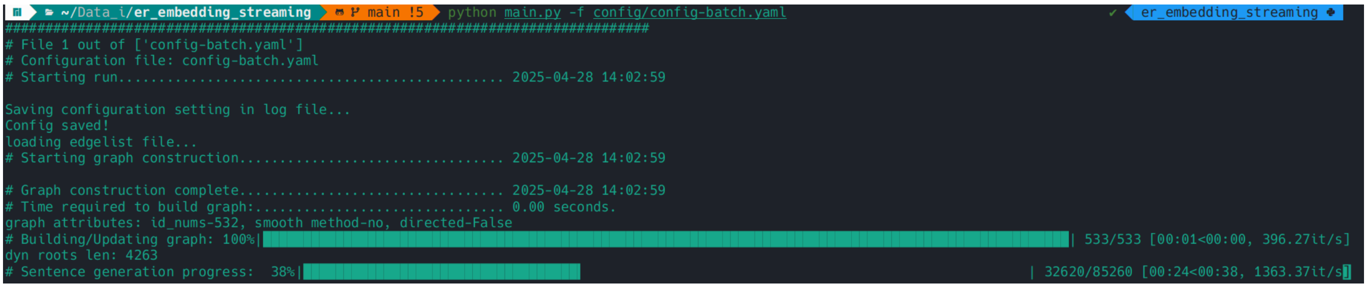
- For testing, do not need to change anything
- During experiments, set your own variables. You can find some examples in `config/examples/`

> more config settings and corresponding explication can be found in files under the directory `config/default/`, but do not modify parameters in these files directly.

3.2. Execute the pre-training script:

```
python main.py -f config/examples/config-batch.yaml
```

In termial, you'll get:



```
~/Data_i/er_embedding_streaming  main !5  python main.py -f config/config-batch.yaml            ✔  er_embedding_streaming
################################################################################
# File 1 out of ['config-batch.yaml']
# Configuration file: config-batch.yaml
# Starting run............................................. 2025-04-28 14:02:59

Saving configuration setting in log file...
Config saved!
loading edgelist file...
# Starting graph construction............................... 2025-04-28 14:02:59

# Graph construction complete............................... 2025-04-28 14:02:59
# Time required to build graph:............................. 0.00 seconds.
graph attributes: id_nums-532, smooth method-no, directed-False
# Building/Updating graph: 100%|███████████████████████████████████████████████████████████████████████████████████████████████████| 533/533 [00:01<00:00, 396.27it/s]
dyn roots len: 4263
# Sentence generation progress:  38%|██████████████████████████████████                                                                 | 32620/85260 [00:24<00:38, 1363.37it/s]
```

This may take a few seconds or minutes.

3.3. Verify the output in the following folders:

```
pipeline/graph/<output_file_name>.graphml   # xml
pipeline/embeddings/<output_file_name>.emb  # binary file
```

`output_file_name` is set in configuration file, by default, `output_file_name: fodors_zagats`

4. Start streaming process

4.1. Set configuration

- Nothing to do when testing
- During experiments, keep in mind the following variables.

```
graph_file: pipeline/graph/<output_file_name>.graphml
embeddings_file: pipeline/embeddings/<output_file_name>.emb
kafka:
    topicid: <user_name>
    groupid: <user_name>_consumer_group
```

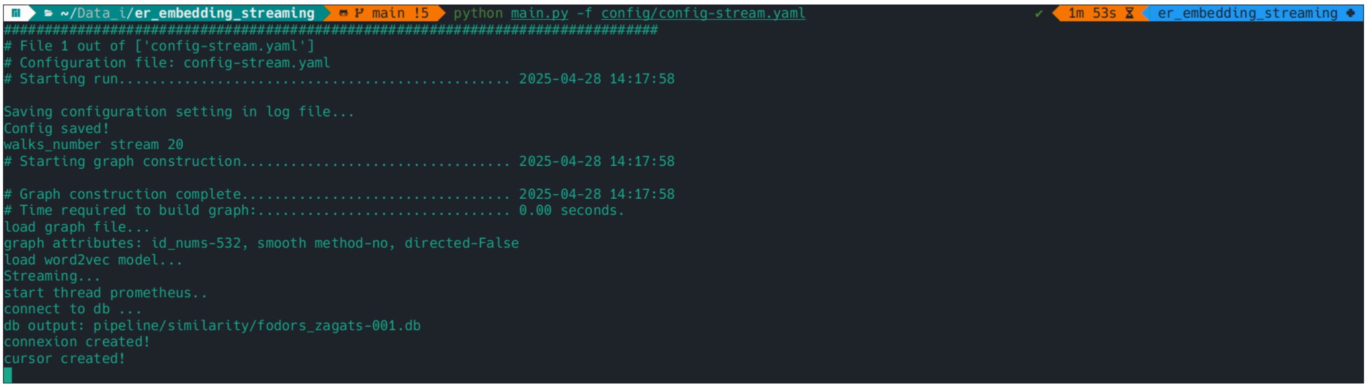
> ⚠ Attention 1: `kafka_topic_id` needs to be the same as topic ID of producer, here the producer is created by `Simulator`.

> ⚠ Attention 2: Each user should set `topicid` by his or her own `user_name` to prevent mixing data from different producer-consumer applications when multiple applications were running at the same time

4.2. Execute the kafka consumer

```
python main.py -f config/examples/config-stream.yaml
```

You'll get..



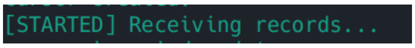
```
 ~/Data_i/er_embedding_streaming  main !5  python main.py -f config/config-stream.yaml          ✔  1m 53s ⧗  er_embedding_streaming
################################################################################
# File 1 out of ['config-stream.yaml']
# Configuration file: config-stream.yaml
# Starting run................................................ 2025-04-28 14:17:58

Saving configuration setting in log file...
Config saved!
walks_number stream 20
# Starting graph construction................................ 2025-04-28 14:17:58

# Graph construction complete................................ 2025-04-28 14:17:58
# Time required to build graph:.............................. 0.00 seconds.
load graph file...
graph attributes: id_nums-532, smooth method-no, directed-False
load word2vec model...
Streaming...
start thread prometheus..
connect to db ...
db output: pipeline/similarity/fodors_zagats-001.db
connexion created!
cursor created!
```

Kafka consumer service is waiting for the producer to send messages to broker.

4.3. Run the simulator

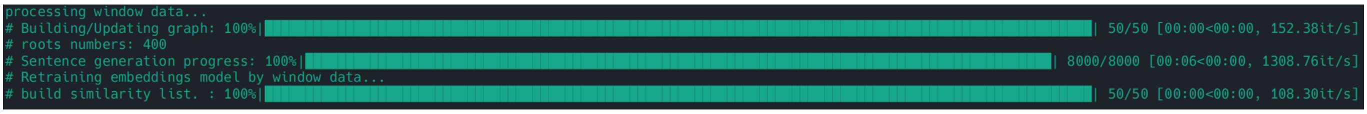
In another terminal, run the simulator, start Kafka producer service, details in section #Simulator

Messages are received when you see the following prompts:



```
[STARTED] Receiving records...
```

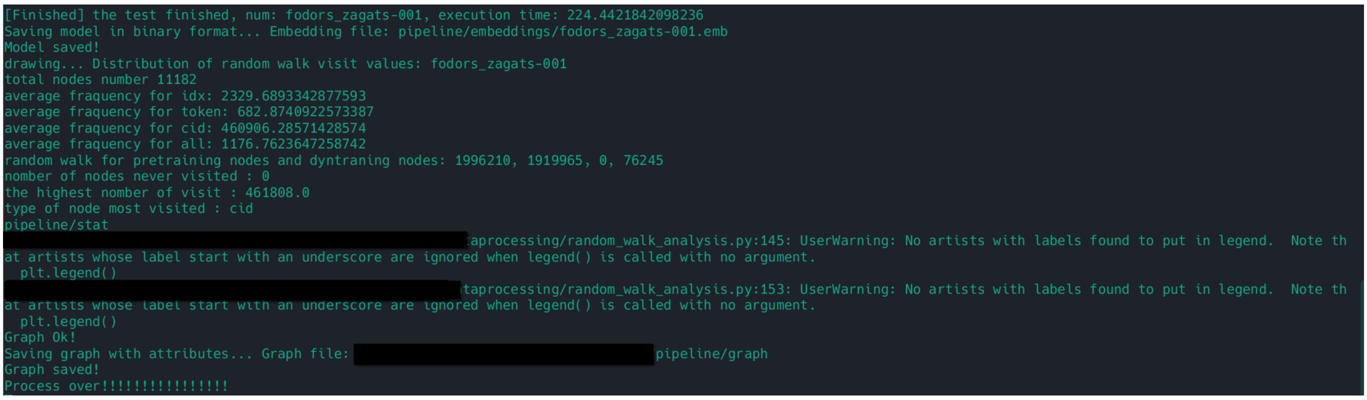
The embedding model is being trained if you see..



```
processing window data...
# Building/Updating graph: 100%|██████████████████████████████████████████| 50/50 [00:00<00:00, 152.38it/s]
# roots numbers: 400
# Sentence generation progress: 100%|█████████████████████████████████████| 8000/8000 [00:06<00:00, 1308.76it/s]
# Retraining embeddings model by window data...
# build similarity list. : 100%|██████████████████████████████████████████| 50/50 [00:00<00:00, 108.30it/s]
```

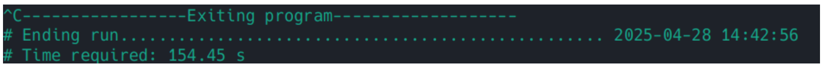
4.4. Stop the program

All data is processed when you see `process over!!!!` Like this:



```
[Finished] the test finished, num: fodors_zagats-001, execution time: 224.4421842098236
Saving model in binary format... Embedding file: pipeline/embeddings/fodors_zagats-001.emb
Model saved!
drawing... Distribution of random walk visit values: fodors_zagats-001
total nodes number 11182
average fraquency for idx: 2329.6893342877593
average fraquency for token: 682.8740922573387
average fraquency for cid: 460906.28571428574
average fraquency for all: 1176.7623647258742
random walk for pretraining nodes and dyntraning nodes: 1996210, 1919965, 0, 76245
nomber of nodes never visited : 0
the highest nomber of visit : 461808.0
type of node most visited : cid
pipeline/stat
[redacted]aprocessing/random_walk_analysis.py:145: UserWarning: No artists with labels found to put in legend.  Note th
at artists whose label start with an underscore are ignored when legend() is called with no argument.
  plt.legend()
[redacted]taprocessing/random_walk_analysis.py:153: UserWarning: No artists with labels found to put in legend.  Note th
at artists whose label start with an underscore are ignored when legend() is called with no argument.
  plt.legend()
Graph Ok!
Saving graph with attributes... Graph file: [redacted]pipeline/graph
Graph saved!
Process over!!!!!!!!!!!!!!!!
```

In terminal, stop the program with `ctrl + c`



```
^C-----------------Exiting program------------------
# Ending run................................................ 2025-04-28 14:42:56
# Time required: 154.45 s
```

4.5. Verify output

You can find similarity file in

```
pipeline/similarity/<output_file_name>.db
```

5. Evaluate

5.1. Set config

- nothing to do when testing
- remember to modify the following variables during experiments. `Similarity_file` is the output result in step 4.5, `match_file` is the ground truth for these dataset

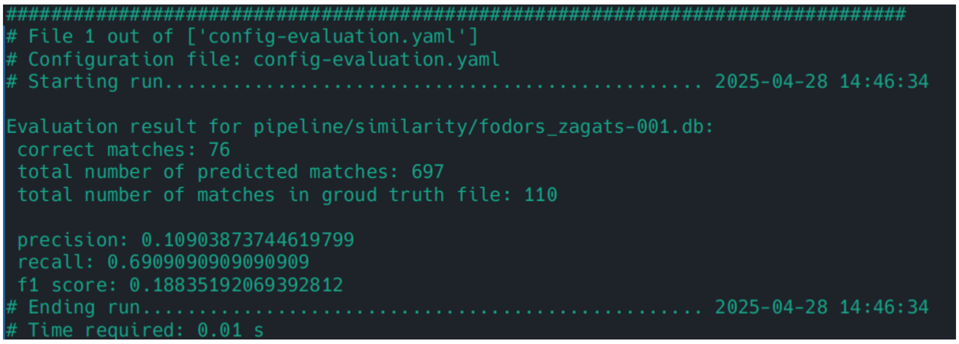
```
similarity_file:pipeline/similarity/<output_file_name>.db
match_file:<ground truth file>
```

5.2. Run the evaluation process

Execute the following command:

```
python main.py -f config/examples/config-evaluation.yaml
```

Results of evaluation are shown as follows:



```
################################################################################
# File 1 out of ['config-evaluation.yaml']
# Configuration file: config-evaluation.yaml
# Starting run............................................... 2025-04-28 14:46:34

Evaluation result for pipeline/similarity/fodors_zagats-001.db:
 correct matches: 76
 total number of predicted matches: 697
 total number of matches in groud truth file: 110

 precision: 0.10903873744619799
 recall: 0.6909090909090909
 f1 score: 0.18835192069392812
# Ending run................................................. 2025-04-28 14:46:34
# Time required: 0.01 s
```

> Repete 3-6 for more tests

Simulator

1. Clone the code

```
git clone https://github.com/Mzhongwei/dataStreamSimulator.git
```

2. Set configuration

2.1. Get config file

Config file path: `src/main/resources/application.properties.example`

Remove extension `.example`

2.2. Modify the file by your own settings

Uncomment the line `# csv.file.path=<your file path>`

Replace `<your file path>` by the file path to the dataset which will be sent as streams

- For initial test, replace the `csv.file.name` with the path where you cloned er_embedding_streaming in section ER step 1: `csv.file.path=<rootFolder>/er_embedding_streaming/Data_example/fodors_zagats-tableB.csv`
- During experiments, pay attention to the following variables:

```
spring.kafka.producer.topic-id=<user name>
spring.kafka.producer.group-id=<user name>_producer_group
csv.file.name=<location of dataset>
```

> Make sure kafka producer ID is your own ID, which is also the same as what you set for the consumer.

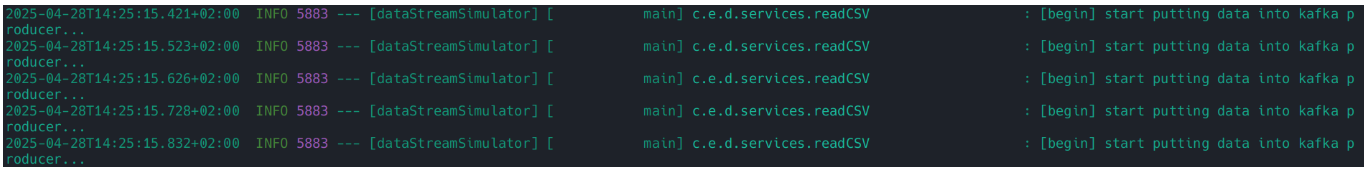
> Remember to modify the file path to dataset as well

3. Run simulator

In the project directory, execute the following command to run the simulator:

```
mvn spring-boot:run
```

The simulator is running correctly if you see the following messages:



```
2025-04-28T14:25:15.421+02:00  INFO 5883 --- [dataStreamSimulator] [           main] c.e.d.services.readCSV                   : [begin] start putting data into kafka p
roducer...
2025-04-28T14:25:15.523+02:00  INFO 5883 --- [dataStreamSimulator] [           main] c.e.d.services.readCSV                   : [begin] start putting data into kafka p
roducer...
2025-04-28T14:25:15.626+02:00  INFO 5883 --- [dataStreamSimulator] [           main] c.e.d.services.readCSV                   : [begin] start putting data into kafka p
roducer...
2025-04-28T14:25:15.728+02:00  INFO 5883 --- [dataStreamSimulator] [           main] c.e.d.services.readCSV                   : [begin] start putting data into kafka p
roducer...
2025-04-28T14:25:15.832+02:00  INFO 5883 --- [dataStreamSimulator] [           main] c.e.d.services.readCSV                   : [begin] start putting data into kafka p
roducer...
```