Assignment-4

GMLFA (AI60007) - Autumn,2024 - IIT Kharagpur

Release Date: [17 /10/2024]
Submission Date: [31 /10/2024]
Total Marks: 21

General Instructions:

- All graded questions are compulsory to solve, and non-graded questions are optional.
- ***Negative marking*** will be there as per our ***plagiarism policy*** given in the course webpage.
- You can use any language for coding questions, but ***'python'*** is preferred.
- Frameworks like Pytorch and Tensorflow are encouraged to construct deeper neural network architectures.
- Any required help will be provided to you in the code notebook regarding data or any specific library.

Submission Instructions:

Following are the Deliverables and submission instructions for the assignment:

1. **Code Notebook (.ipynb):** A notebook containing all the code, including the implementation and execution of experiments. Notebook Format: <***group_number*>*_assignment4.ipynb,*** replace <***group_number***> with your assigned group number.
2. Install the necessary libraries for loading the specific datasets for this assignment.
3. Ensure that the notebook runs smoothly considering the aforementioned dataset loaded from the pykeen library to yield the expected results.
4. **Report (.pdf):** A comprehensive report documenting all findings from the experiments conducted. Report Format: ***group_number_assignment4.pdf***
5. Use the device as **GPU** for this assignment, select **change runtime type** in google collab and select **T4 GPU**.

Problem Statement:

1-hop question-answering and Similar Fact Retrieval in Knowledge Graph:

The objective of this assignment is to develop a custom implementation of two knowledge graph embedding techniques—**TransE** and **TransR**— for the task of 1-hop question-answering and design a model for similar fact retrieval to identify and rank the top-5 most similar facts to a given fact. You will:

1. **Implement TransE and TransR models** for obtaining the embedding entities and relations in a knowledge graph.
2. **Perform a 1-hop question-answering** to infer missing entities in the knowledge graph.
3. **Evaluate and compare** the performance of both models using appropriate metrics.
4. **Perform a similar fact retrieval** to extract top-5 most similar facts for a given fact.

Dataset Description:

Use the **Nations and Kinships** datasets for the 1-hop question-answering and **Nations** dataset for the similar fact retrieval task.

Nations: The Nations dataset is a small knowledge graph with 14 entities, 55 relations, and 1992 triples describing countries and their political relationships.

Kinships: The Kinships dataset describes relationships between members of the Australian tribe Alyawarra and consists of 10,686 triples. It contains 104 entities representing members of the tribe and 26 relationship types that represent kinship terms such as Adiadya or Umbaidya.

Tasks:

Setup and Data Preprocessing (2 marks)

- Install the pyKEEN library.
- Load the datasets from the **pyKEEN** library extracting triples for training, validation and testing. (https://pykeen.readthedocs.io/en/stable/reference/datasets.html)

TransE and TransR Implementation Specifications(6 marks)

- **Embedding Dimension**: 100
- **Margin**: Vary 1.0 to 5.0 with step size 1.0 for Nations dataset and take 1.0 for Kinships dataset.

- **Optimizer**: Adam, learning rate 0.001, train for 50 epochs.
- Use **margin-based ranking loss.**
- Use **Bernoulli Negative sampling** (without using the built-in library function) for generating the negative samples.
  Refer:https://pykeen.readthedocs.io/en/stable/reference/negative_sampling.html

1-hop question-answering (6 marks)

- For each triplet in the test set, perform head and tail prediction:
  - **Head Prediction:** Replace the head entity with all possible entities and rank the scores. [Example: Given the triple **(?, hasCapital, France)**, if we replace **?**(head) with possible entities: **London**, **Berlin**, **Paris**, **Madrid**. The model should predict the correct head as **Paris**.]
  - **Tail Prediction:** Replace the tail entity with all possible entities and rank the scores. [Example: Given the triple **(Paris, hasCapital, ?)**, if we replace ? (tail) with possible entities: **France**, **Germany**, **Spain**, **UK**. The model should predict the correct tail as **France**.]
- Compute ranking metrics for each prediction.
- Overall Evaluation:
  - Use the RankBasedEvaluator from https://pykeen.readthedocs.io/en/latest/api/pykeen.evaluation.RankBasedEvaluator.html to obtain the metrics as follows:-
    - **Mean Rank (MR):** Average rank of the correct entity.
    - **Hits@10:** Proportion of correct entities ranked in the top 10
    - Compare the performance of TransE and TransR based on these metrics.

Similar Fact Retrieval (7 marks)

Design an unsupervised model to compute the similarity between triples in the **Nations** dataset. We have provided a 5 facts validation set, for which you have to retrieve 5 similar facts for each fact.

```
Triple1: ['brazil', 'commonbloc1', 'india']
Triple2: ['burma', 'intergovorgs3', 'indonesia']
Triple3: ['china', 'accusation', 'uk']
Triple4: ['cuba', 'reldiplomacy', 'china']
Triple5: ['egypt', 'embassy', 'uk']
```

1. Use the TransE and TransR embeddings (dimension=30) of the elements of the triples to derive the embedding of the input triples.
2. Implement a dot-product based similarity score function to evaluate how similar a validation triple embedding is to others in the dataset. Based on this similarity, you will rank and retrieve the top 5 most similar triples for each given validation triple for both the models.

Reporting:

- Present the evaluation results for both the models (TransE and TransR).
- Discuss the differences in performance between TransE and TransR.
- Report the results of the top 5 most similar triples for each given validation triple for both the models.