A Project Report

on

EduEmbed - Embeddings for Education

Web Science Lab (WSL)

Masters In Technology

COMPUTER SCIENCE AND ENGINEERING

BY

Sahil Khatri(MT2022095)
Sheetal Agarwal(MT2022109)

Under the Guidance of

Prof. Srinath Srinivas
Anurag Mohanty

International Institute Of Information Technology
Bangalore

Jan - May 2023

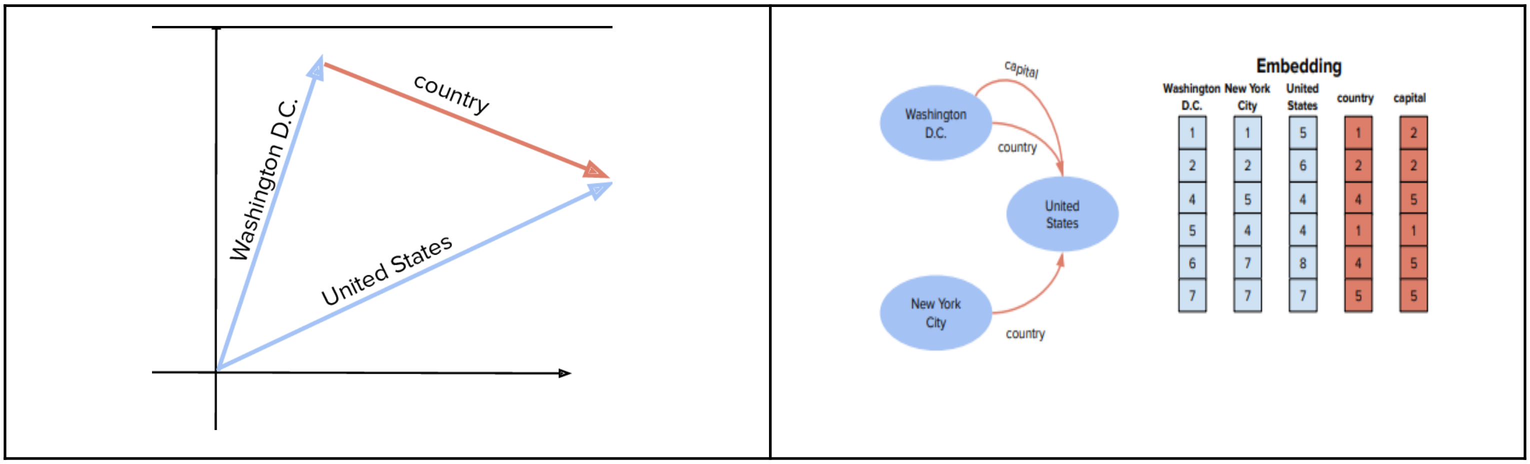
1. Overall objective of the project

The object of the EduEmbed project is to generate embeddings for the Education Domain using Knowledge Graph Models such that they can understand the underlying semantics of how the triples are correlated with each other. Once this has been done, the embeddings can be used for various education domain related tasks. Such as, curriculum generation, course sequencing, difficulty analysis, etc. To achieve this it is of utmost importance that the embeddings generated are very much effective in understanding the context of the domain. For these are training and analyzing various models such as TransE, HolE, and TransH and their generated embeddings.

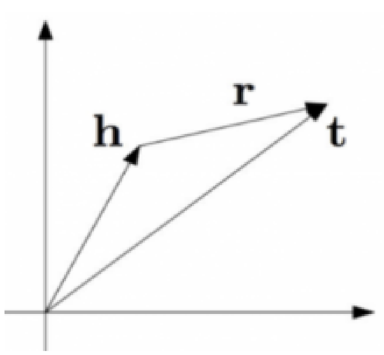
Knowledge Graph Embeddings:
KG represents diverse types of information in the form of different types of entities connected via different types of relations. Information extracted from KGs in the form of embeddings is used to improve search, recommend products, and infer missing domain specific context. Popular KGE models are TransE, TransH, etc. which define different score functions to learn entity and relation embeddings. Input data for KGE is in the form of triplets (head, relation, tail).

KGE Models:

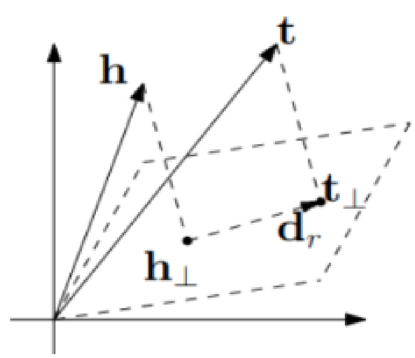
TransE



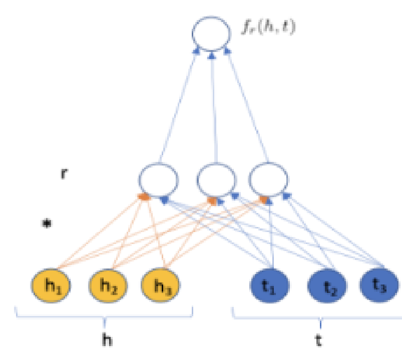
$$h + r \text{ and } t \text{, or } f = -\|h + r - t\|_{\frac{1}{2}}$$



TransE



TransH



HolE

2. Your responsibility in the project

Our responsibility in the project begins with understanding the objective and the existing work that was done till then. After that we had to understand the underlying technology i.e. Knowledge Graph Embeddings and its applications. Perform some basic tasks to get hold of the concepts that will be used across the project. After that we worked on creating and training the triples on TransE. Later included the weights for relations. Then trained the models and analyzed the embeddings at a very initial stage. With the detailed understanding of the working of the models we moved ahead on creating automation scripts which can be used to train multiple models (TransE, HolE, TransH) with multiple hyperparameters. Updated the package module to get some desired output as per our requirement. Filtered and extracted subset of the original data which can be later used for qualitative analysis of the embeddings. And summarize the results obtained.

3. Sprint report

Sheetal Agarwal

Sprint Title	Sprint Description	Start Date of Sprint	End Date of Sprint	Major Outcomes
Sprint 1	Understanding of existing code	9/1/2023	15/1/2023	Understanding of existing code
Sprint-2	Software installation and technology understanding	16/1/2023	22/1/2023	Software installed and Knowledge graph understanding
Sprint 3	Implementation understanding and POC	23/1/2023	29/1/2023	POC and preprocessing
Sprint-4	TransE with Dummy weights	30/1/2023	5/2/2023	TransE with Dummy weights
Sprint-5	TransE with actual weight	6/2/2023	13/2/2023	TransE with actual weights
Sprint-6	Embedding similarity, HolE	14/2/2023	20/2/2023	Embedding Similarity and HolE
Sprint-7	Scaling the weights and splitting data for each relation	20/2/2023	27/2/2023	scaling the weights and splitting data for each relation
Sprint-8	Run models for more epochs and compare evaluation	28/2/2023	6/3/2023	Finding best hyperparameter based on evaluation matrix
Sprint-9	Use tfidf score to generate weghts for concept vocab index	6/3/2023	20/3/2023	Use tfidf score to generate weghts for concept vocab index
Sprint-10	Remove duplicates from concept vocab index and Tfidf as weight for concept vocab index.	20/3/2023	26/3/2023	Remove duplicates from concept vocab index to get better embeddings and used TFIDF score as weight for concept vocab index.

Sprint-11	Vectorization issue faced due to certain discrepancy in training and test set and Evaluation on data with weights for TransE and HolE.	27/3/2023	2/4/2023	Quantitative result analysis and consistent train and test set.
Sprint-12	-Cosine similarity of H+R=T for transE and TransH. -Gather concept vocab index list	3/4/2023	9/4/2023	-Cosine similarity of H+R=T for transE and TransH. -Gather concept vocab index list
Sprint-13	Custom setup to fetch loss values and storing it for generating loss vs epoch graphs of TransE, TransH and HolE for each combination of hyperparameters. This was done for two sets of data one which contain the topic and concept-vocab relation while other didn't have that relation.	10/4/2023	16/4/2023	Tentatively finalized hyperparameter values based on graph, score metrics and embedding quality. This was done for two sets of data one which contain the topic and concept-vocab relation while other didn't have that relation. After analysis data with concept-vocab and topic relation performed better than the other set.
Sprint-14	Evaluated cosine similarity: 1. For two different entities, 2. head+ relation and	17/4/2023	26/4/2023	Qualitative analysis of the transH generated embeddings.

	tail , 3. head + relation(l_text_topic) of entity1 and head +relation(l_text_topic) of entity2 (Both entity having same topic as tail) 4. head + relation(concept_vocab_index) of entity1 and head +relation(concept_vocab_index) of entity2 (Both entity having same concept_vocab as tail)			

Sahil Khatri

Sprint No.	Sprint Description	Start Date of Sprint	End Date of Sprint	Major Outcomes
Sprint 1	Understanding of existing code	9/1/2023	15/1/2023	Understanding of existing code
Sprint 2	Software package installation and technology understanding	16/1/2023	22/1/2023	Software package installation and knowledge graph understanding
Sprint 3	Implementation understanding and POC	23/1/2023	29/1/2023	POC and preprocessing
Sprint 4	TransE with dummy weights	30/1/2023	5/2/2023	TransE with dummy weights
Sprint - 5	TransE with actual weights	6/2/2023	13/2/2023	TransE with actual weights
Sprint - 6	Embedding similarity and HolE,	13/2/2023	20/2/2022	cosine similarity and holE
Sprint - 7	Scaling the weights and splitting data for each relation.	20/2/2023	27/2/2023	Scaling the weights and splitting data for each relation.
sprint 8	Run for more epochs and compare evaluation results.	27/2/2023	6/3/2023	Working on finding hyper parameters based on the evaluation results.
Sprint - 9	Use tf-idf score to generate weights for concept vocab index	6/3/2023	20/3/2023	Working on generating tf-idf score for the word corresponding to concept vocab index
Sprint - 10	structural weight as hyper parameter for further training	20/3/2023	26/3/2023	trained model with various hyperparameters giving more importance to weights
Sprint - 11	Evaluate model	27/3/2023	2/4/2023	Quantitative result analysis and consistent

	performance on new data with triple weights. Vectorization issue faced due to certain discrepancy in training and test set.			train and test set.
Sprint - 12	TransH model setup to support our custom dataset and triples weights	10/4/2023	16/4/2023	Successful setup of transH and Quantitative analysis for transH, transE, holE
Sprint - 13	Automation to test multiple models and evaluate their results based on hyperparameter values for TransE, HolE, TransH. Also, this was done for 2 sets of data, one which contained the relation between topic and concept vocab while the other didn't contain these relations.	10/4/2023	16/4/2023	Automated generation of separate directory for each category of model for each combination of hyperparameters. And use the corresponding generated embeddings and results for cosine similarity and loss analysis using graphs. The data with relation between topic and concept vocab performed better comparatively.
sprint -14	Find common percentage of concept_vocab_index among 2 courses which have similar l_text_topics	17/4/2023	26/4/2023	Used the file with same ltt and cv for cosine similarity tasks

4. Final summary of sprints

Sheetal Agarwal

First I had the understanding of the existing code, and the technology used in this project, which is Knowledge graph embedding, models used in it etc.
Installed the required softwares and ran that code on my own system.
Implemented TransE with dummy and actual weights. Did the same for the holE.
To generate weights used tf idf score for concept vocab index.
Split the data in such a way all the relation related triples include in the Train, Test and Validation set. And generate the embeddings from it for each Model (TransE, TransH and HolE).
Run each model (TransE, TransH and HolE) for different Hyperparameters to get the best hyperparameter out of it.
Removed the duplicates from concept vocab index to get better embeddings.
Find the cosine similarity of embedding of head + embedding of relation and embedding of Tail entity (H+R=T) to check how accurate embedding are that are generated from transE and TransH.
Gather concept vocab index list related to this domain to have better triples.
Finalized hyperparameter values based on graph, score metrics and embedding quality. This was done for two sets of data one which contain the topic and concept-vocab relation while the other didn't have that relation.
After analysis data with concept-vocab and topic relation performed better than the other set.

Sahil Khatri

After getting the understanding of the existing code, I learnt what knowledge graph is and its related models etc.
Installed the required softwares and ran that code on my own system.
Implemented TransE with dummy and actual weights. Did the same for the holE.
To generate weights used tf idf score for concept vocab index.
Split the data in such a way all the relation related triples include in the Train, Test and Validation set. And generate the embeddings from it for each Model.
Did set up of transH and Quantitative analysis of it.
Run each model (TransE, TransH and HolE) for different Hyperparameters to get the best hyperparameter out of it.
Removed the duplicates from concept vocab index to get better embeddings.
Finalized hyperparameter values based on graph, score metrics and embedding quality. This was done for two sets of data one which contain the topic and concept-vocab relation while the other didn't have that relation.
To do the above task create an automation script which generates a separate directory for each category of model for each combination of hyperparameters.
And use the corresponding generated embeddings and results for cosine similarity and loss analysis using graphs.
The data with relation between topic and concept vocab performed better comparatively.

5. Source code details

Repository Link : https://github.com/anmohy/EduEmbedd

The code base can be downloaded from the repository link.
The codes are written to accomplish various tasks such as,

1. Preparing the data
2. Creating triples for different types of relations
3. Assigning each triple with different a weight value
4. Combining the triples and splitting it into train, val, test set
5. Training different types of Knowledge Graph Models (such as TransE, TransH, HolE)
6. Automation to support multiple model training with various combinations of hyper-parameter and to store their results in a properly defined directory hierarchy
7. Files to find the cosine similarity among different types of inputs for the qualitative analysis of the embeddings (entity-entity similarity, head+relation & tail similarity, head+relation & head+relation)

Further detail of the code is mentioned in the README file of the Repository.
The input output files required and expected by the script are mentioned in the respective python code.
The function description is also mentioned in detail along with commented example wherever needed

6. Structure of the code

The code structure is quite complex, it is not feasible to describe it completely here, so please refer to the README file in the repository for detailed structure of the code.
Here is the overview of the code structure.

- code
    - data
    - eduTransE_HolE
    - eduTransH
    - embeddings_final
        - transE
            - transE_50_5_40_0.1_0.1
            - transE_50_5_50_0.1_0.1

              .
              .
              .
        - holE
            - holE_50_5_40_0.1_0.1
            - holE_50_5_50_0.1_0.1

              .
              .
              .
        - transH
            - transH_50_5_40_0.1_0.1
            - transH_50_5_50_0.1_0.1

              .
              .
              .
        - input (contains multiple .csv files which are used as input to other files)

        - output (contains the output files generated by various code modules)

- conceptvocab_percentage.py
- cosine_similarity.py
- embedding_generation.py
- graphs.py
- hr-hr_cosine_similarity.py
- hr-t_cosine_similarity.py
- hr-t_e1-e2_cosine_similarity.py
- Manual_generated_data.xlsx
- README.txt

Screenshots

Data Preprocessing and Feature Extraction

file_name	text	course_name	temp	week	section	lesson	course_title	week_no	section_no	lesson_no	text1	text_topics	l_text_topics	l_text_prob	join_text	concept_vocab
Course1_W1-S1-L1_Introduction_Part_1_11-17	okay welcome natural language processing name ...	Course1	W1-S1-L1	W1	S1	L1	Introduction	1	1	1	okay welcom natur languag process name michael...	[0 11 10 12 7]	[1, 2, 6, 7, 9, 11, 14]	[13.13, 29.09, 2.45, 36.34, 12.84, 4.22, 1.85]	okay welcome natural language processing name ...	[vi43, vi106 vi1063, vi43,
Course1_W1-S1-L2_Introduction_Part_2_10-28	next want talk key challenges nlp answering qu...	Course1	W1-S1-L2	W1	S1	L2	Introduction	1	1	2	next want talk key challeng nlp answer questio...	[10 11 0 13 12]	[1, 2, 7, 9, 10]	[24.44, 52.35, 15.23, 3.47, 4.42]	next want talk key challenges nlp answering qu...	[vi70, vi1068 vi136, vi43,
Course1_W1-S2-L1_Introduction_to_the_Language_...	okay first topic going cover course problem la...	Course1	W1-S2-L1	W1	S2	L1	Introduction	1	2	1	okay first topic go cover cours problem langua...	[11 6 0 10 3]	[5, 7, 9, 13]	[8.0, 5.07, 35.97, 50.77]	okay first topic going cover course problem la...	[vi43, vi106 vi1575, vi828,
Course1_W1-S2-L2_Introduction_to_the_Language_...	soon start talk techniques solve precisely pro...	Course1	W1-S2-L2	W1	S2	L2	Introduction	1	2	2	soon start talk techniqu solv precis problem p...	[11 0 6 10 3]	[1, 6, 9, 13]	[43.18, 1.01, 7.11, 48.54]	soon start talk techniques solve precisely pro...	[vi149, vi199 vi1419, vi1
Course1_W1-S2-L3_Markov_Processes_Part_1_8-56	okay previous segments lecture gave basic defi...	Course1	W1-S2-L3	W1	S2	L3	Markov	1	2	3	okay previou segment lectur gave basic definit...	[3 11 1 13 4]	[12, 13]	[13.18, 86.63]	okay previous segments lecture gave basic defi...	[vi1575 vi1575, vi888 vi1

Triples

head	variable	value
Course3_W9-S1-L3_-_Summarizatio...	l_text_topics	topic_11
Course3_W9-S1-L4_-_Summarizatio...	l_text_topics	topic_4
Course3_W9-S1-L4_-_Summarizatio...	l_text_topics	topic_6
Course3_W9-S1-L4_-_Summarizatio...	l_text_topics	topic_11
Course3_W9-S1-L4_-_Summarizatio...	l_text_topics	topic_12
Course3_W9-S1-L5_-_Summarizatio...	l_text_topics	topic_6
Course3_W9-S1-L6_-_Sentence_Sim...	l_text_topics	topic_1
Course3_W9-S1-L6_-_Sentence_Sim...	l_text_topics	topic_6
Course3_W9-S1-L6_-_Sentence_Sim...	l_text_topics	topic_9
Course3_W9-S1-L6_-_Sentence_Sim...	l_text_topics	topic_12
Course1_W1-S1-L1_Introduction_P...	concept_vocab_index	vi912
Course1_W1-S1-L1_Introduction_P...	concept_vocab_index	vi96

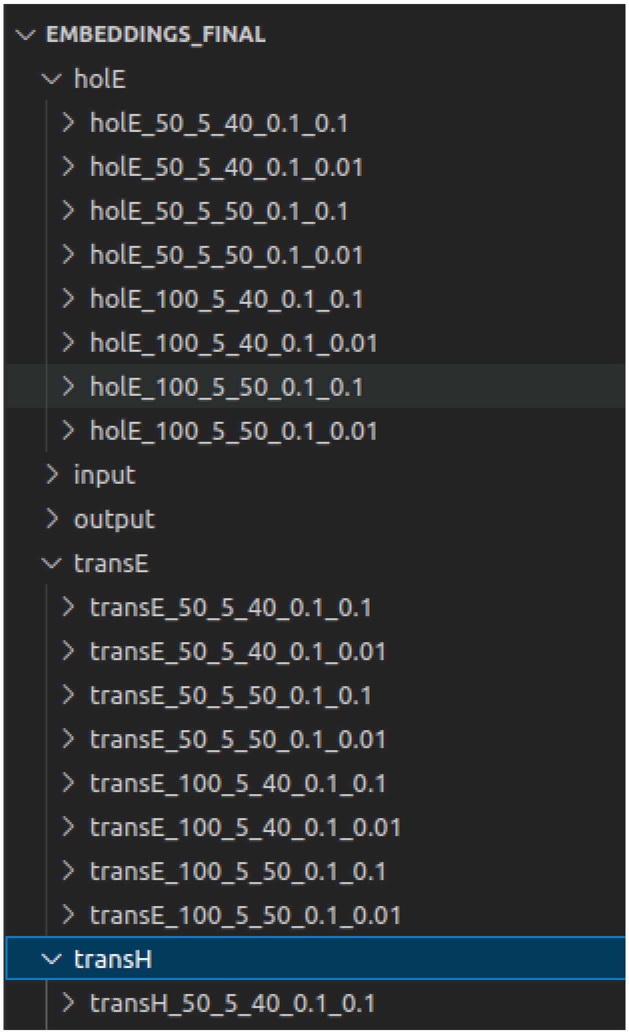
Concept-Vocab with weights using Tf-Idf

Course1_W10-S2-L3_The_Dependency_Parsing_Problem_Part_2_13-53	concept_vocab_index	vi946	0.01
Course1_W10-S2-L3_The_Dependency_Parsing_Problem_Part_2_13-53	concept_vocab_index	vi1178	0.05
Course1_W10-S2-L3_The_Dependency_Parsing_Problem_Part_2_13-53	concept_vocab_index	vi442	0.05
Course1_W10-S2-L3_The_Dependency_Parsing_Problem_Part_2_13-53	concept_vocab_index	vi47	0.01
Course1_W10-S2-L3_The_Dependency_Parsing_Problem_Part_2_13-53	concept_vocab_index	vi235	0.02
Course1_W10-S2-L3_The_Dependency_Parsing_Problem_Part_2_13-53	concept_vocab_index	vi262	0.03
Course1_W10-S2-L3_The_Dependency_Parsing_Problem_Part_2_13-53	concept_vocab_index	vi134	0.01
Course1_W10-S2-L3_The_Dependency_Parsing_Problem_Part_2_13-53	concept_vocab_index	vi74	0.02
Course1_W10-S2-L3_The_Dependency_Parsing_Problem_Part_2_13-53	concept_vocab_index	vi980	0.01
Course1_W10-S2-L4_GLMs_for_Dependency_Parsing_Part_1_11-59	concept_vocab_index	vi984	0.02
Course1_W10-S2-L4_GLMs_for_Dependency_Parsing_Part_1_11-59	concept_vocab_index	vi1206	0.11
Course1_W10-S2-L4_GLMs_for_Dependency_Parsing_Part_1_11-59	concept_vocab_index	vi873	0.03
Course1_W10-S2-L4_GLMs_for_Dependency_Parsing_Part_1_11-59	concept_vocab_index	vi1434	0.2
Course1_W10-S2-L4_GLMs_for_Dependency_Parsing_Part_1_11-59	concept_vocab_index	vi351	0.02
Course1_W10-S2-L4_GLMs_for_Dependency_Parsing_Part_1_11-59	concept_vocab_index	vi47	0.04
Course1_W10-S2-L4_GLMs_for_Dependency_Parsing_Part_1_11-59	concept_vocab_index	vi442	0.02
Course1_W10-S2-L4_GLMs_for_Dependency_Parsing_Part_1_11-59	concept_vocab_index	vi512	0.19
Course1_W10-S2-L4_GLMs_for_Dependency_Parsing_Part_1_11-59	concept_vocab_index	vi1422	0.02
Course1_W10-S2-L4_GLMs_for_Dependency_Parsing_Part_1_11-59	concept_vocab_index	vi55	0.05
Course1_W10-S2-L4_GLMs_for_Dependency_Parsing_Part_1_11-59	concept_vocab_index	vi98	0.03
Course1_W10-S2-L4_GLMs_for_Dependency_Parsing_Part_1_11-59	concept_vocab_index	vi122	0.02
Course1_W10-S2-L4_GLMs_for_Dependency_Parsing_Part_1_11-59	concept_vocab_index	vi988	0.1
Course1_W10-S2-L4_GLMs_for_Dependency_Parsing_Part_1_11-59	concept_vocab_index	vi1341	0.11
Course1_W10-S2-L4_GLMs_for_Dependency_Parsing_Part_1_11-59	concept_vocab_index	vi1338	0.05

Cosine similarity (Embedding of TransH)

head	relation	tail	cos_sim_score
Course1_W1-S2-L2_Introduction_to_the_Language_Modeling_Problem_Part_2_7-12	l_text_topics	topic_12	0.998957668603049
Course1_W1-S2-L3_Markov_Processes_Part_1_8-56	l_text_topics	topic_12	0.950749169979532
Course1_W1-S2-L4_Markov_Processes_Part_2_6-28	l_text_topics	topic_1	0.999668315237906
Course1_W1-S2-L4_Markov_Processes_Part_2_6-28	l_text_topics	topic_12	0.998995795113438
Course1_W1-S2-L5_Trigram_Language_Models_9-40	l_text_topics	topic_7	0.999854714837523
Course1_W1-S2-L5_Trigram_Language_Models_9-40	l_text_topics	topic_11	0.999100864631429
Course1_W1-S2-L5_Trigram_Language_Models_9-40	l_text_topics	topic_12	0.998965324818128
Course1_W1-S2-L6_Evaluating_Language_Models-_Perplexity_12-36	l_text_topics	topic_7	0.99820199170814
Course1_W1-S2-L6_Evaluating_Language_Models-_Perplexity_12-36	l_text_topics	topic_11	0.997567515183758
Course1_W1-S2-L6_Evaluating_Language_Models-_Perplexity_12-36	l_text_topics	topic_12	0.997636153669363
Course1_W1-S3-L1_Linear_Interpolation_Part_1_7-46	l_text_topics	topic_7	0.999848392190715
Course1_W1-S3-L1_Linear_Interpolation_Part_1_7-46	l_text_topics	topic_12	0.998951500896143
Course1_W1-S3-L2_Linear_Interpolation_Part_2_11-35	l_text_topics	topic_7	0.979916300031045
Course1_W1-S3-L2_Linear_Interpolation_Part_2_11-35	l_text_topics	topic_11	0.979582630834873
Course1_W1-S3-L3_Discounting_Methods_Part_1_9-26	l_text_topics	topic_7	0.996274158196031
Course1_W1-S3-L3_Discounting_Methods_Part_1_9-26	l_text_topics	topic_11	0.995690466993966
Course1_W1-S3-L4_Discounting_Methods_Part_2_3-34	l_text_topics	topic_7	0.940899408374497
Course1_W1-S3-L4_Discounting_Methods_Part_2_3-34	l_text_topics	topic_11	0.940876298894193

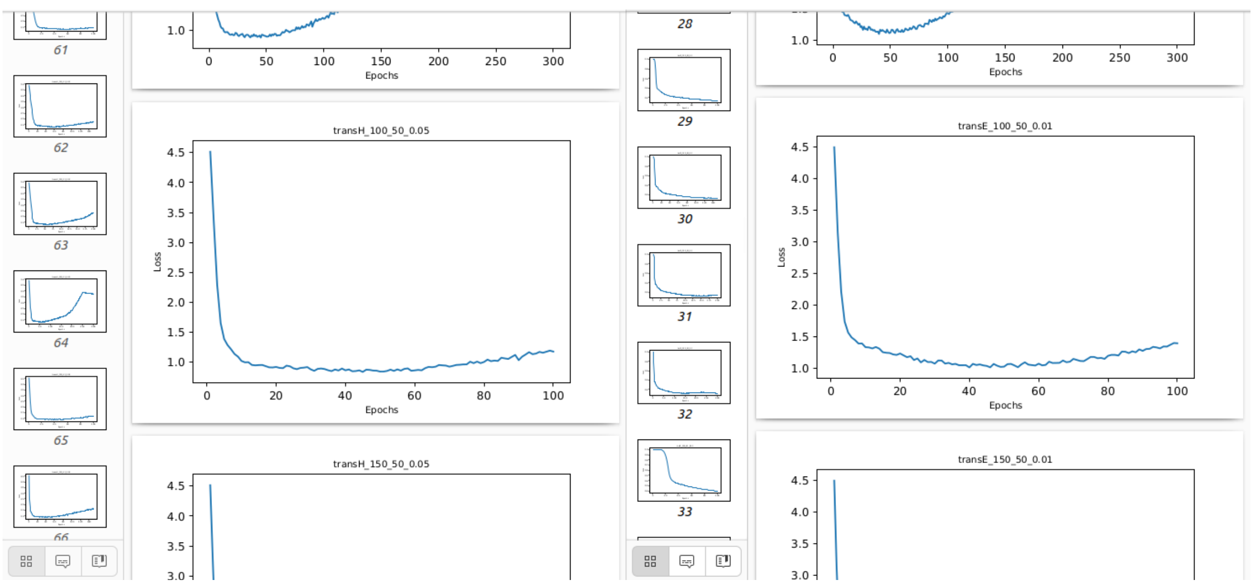
Directory Structure



Hyperparameter Metrices

model_nan	epochs	batches_coun	k	structural_	lr	start_loss	end_loss	mrr	mr	hits_10	hits_5	hits_3
holE	100	5	50	0.1	0.1	1.003445855	0.13282739	0.136809785	77.36415929	0.238938053	0.180088496	0.14292035
holE	100	5	40	0.1	0.01	1.000874222	0.177928071	0.135836614	70.40353982	0.260176991	0.18539823	0.13716814
transH	100	5	40	0.1	0.1	4.311885308	1.131924913	0.131325459	84.93918919	0.261711712	0.184684685	0.1396396
transH	100	5	50	0.1	0.1	4.290151367	1.136793778	0.127743085	85.18918919	0.254954955	0.173873874	0.13108108
holE	150	5	40	0.1	0.01	1.000874222	0.145108493	0.12464887	64.92168142	0.263274336	0.181858407	0.13185840
transH	100	5	50	0.1	0.05	4.515290618	1.176531349	0.112728842	102.0216216	0.213963964	0.146846847	0.11576576
holE	200	5	40	0.1	0.01	1.000874222	0.142629618	0.110368248	68.72168142	0.257522124	0.165929204	0.11150442
transE	150	5	40	0.1	0.05	2.95983568	1.875992047	0.110040697	94.97123894	0.223451327	0.148672566	0.11061946
transE	100	5	50	0.1	0.1	2.812907873	1.53011504	0.107742908	99.82389381	0.220353982	0.149557522	0.10707964
transE	150	5	40	0.1	0.1	2.928856111	2.250745307	0.106670958	86.20707965	0.241150442	0.152654867	0.10442477
transE	100	5	50	0.1	0.05	2.778270083	1.199851345	0.105732788	105.9137168	0.211504425	0.14380531	0.10973451
transE	100	5	40	0.1	0.1	2.928856111	1.381392054	0.105213873	98.43539823	0.221681416	0.147787611	0.10707964
transE	150	5	50	0.1	0.05	2.778270083	1.990873617	0.10503738	95.63230088	0.237610619	0.145575221	0.10663716
transE	100	5	50	0.1	0.01	4.231377044	1.130923959	0.104927972	117.4252212	0.210619469	0.144690265	0.1084070
transE	100	5	40	0.1	0.05	2.95983568	1.142853235	0.104887689	105.4584071	0.22699115	0.150442478	0.10486725
transE	150	5	50	0.1	0.1	2.812907873	2.466611328	0.104418895	91.69646018	0.233185841	0.149115044	0.10353982
transE	150	5	50	0.1	0.01	4.231377044	1.636973741	0.103042215	99.47566372	0.217256637	0.148230088	0.10176991
holE	100	5	40	0.1	0.1	1.002921441	0.11768559	0.101072042	85.63451327	0.213716814	0.148672566	0.10353982
transE	150	5	40	0.1	0.01	4.395874566	1.630313766	0.100276449	100.1765487	0.203097345	0.138495575	0.10884955

Loss Evaluation for TransE, TransH, HolE



7. Libraries Used

For reference we have used a python library “ampligraph”. The details of the library as as below:

Name: ampligraph
Version: 1.4.0
Summary: A Python library for relational learning on knowledge graphs.
Home-page: https://github.com/Accenture/AmpliGraph/
Author: Accenture Dublin Labs
Author-email: about@ampligraph.org
License: Apache 2.0
Location: /home/user/anaconda3/envs/pe/lib/python3.7/site-packages
Requires: beautifultable, flake8, networkx, numpy, pandas, pytest, pyyaml, rdflib, recommonmark, scikit-learn, scipy, setuptools, sphinx, sphinx-rtd-theme, sphinxcontrib-bibtex, tqdm

Ampligraph provides the functionalities to train the different Knowledge Graph Embedding models such as TransE, HolE, ComplEx, etc.

8. System requirements

The code is system independent and can be executed across any OS platform.

The system environment used for the development of the code is as below:

OS : Ubuntu 22.04
Python : 3.7.16
IDE : Spyder, Jupyter Notebook, Google Colab

9. Challenges faced (Bugs detection and correction)

Few of the issues that we faced during the development of the projects are:

1. To get the embeddings of the entities and relations, we need to give the input entity which was given for the training of the KGE model.

To solve this issue, for every model that is being trained we are storing the train_triples, its entities and embeddings so that we can filter our input from these data in further stages (if required).

2. While preparing the triples from the original data, the l_text_topics and concept_vocab_index columns are in string format. Which is supposed to be a list. So we were not able to directly load the data as per the requirement.

To solve this, we incorporated the python code to convert the string data into list and then use it further.

3. For hyperparameter tuning and analysis, we had to go through a tedious task of entering values of each model execution manually into the excel sheet. It was not feasible when we had run the models of multiple parameters and that too for more than one model type.

To overcome these, we created an automated workflow, where based on the model type and its hyperparameter, a directory-subdirectory structure will be created and all the desired output files will be stored in the respective subdirectory of the model.

4. While training the KGE model we were not getting the loss value for each epoch. We had to keep an eye on the progress bar and observe manually how the loss is varying across the training of the model. Again this was not feasible for multiple models with multiple hyperparameters.

To solve these, we modified the package that was being used and made changes in the dependent files to store and return the list of losses for each epoch. And stored in a csv file. Later used these losses list to plot the loss vs epoch graphs, which helped us to identify which model hyperparameters are performing best.

9. Talks given

Review 1:

- Introduction about the project
- What is Knowledge Graph Embeddings
- What is TransE and how it works
- Preprocessing and Feature Extraction from the dataset
- Creating triples for Knowledge Graph
- Embedding generation using TransE

Review 2:

- Using dummy weights for all the triples
- Preparing triples with actual weight for l_text_topics (LDA probability)
- Exploring and Understanding the working of Knowledge Graph Embedding models such as TransE, HolE, TransH
- Comparing embeddings using cosine similarity to analyze the similarity/dissimilarity of the embeddings
- Scaling the weights to generalize the dataset
- Splitting the data uniformly across all the types of relations (l_text_topics, concept_vocab_index, prerequisite, level)
- Hyperparameter tuning
- Metric Evaluation
    - mr_score
    - mrr_score
    - rank_score
    - hits_at_n_score

Review 3:

- Concept-Vocab with weights using Tf-Idf
- TransH cosine similarity to verify the head+relation=tail equation
- Automation of the model training to support multiple models with multiple hyperparameters.
- Hyperparameter Evaluation Metric
- Loss Evaluation for TransE, TransH, HolE
- Qualitative Analysis on the manually selected data with similar l_text_topics
- Cosine Similarity Evaluation
    - Entity and Entity similarity
    - Head + Relation and Tail similarity

- Entity1 (head + relation) and Entity2 (head + relation) for l_text_topics relation
- Entity2 (head + relation) and Entity2 (head + relation) for concept_vocab_index