Complex-Event Representation Learning

Jennifer Lu

Dec 20th, 2023

CS333 Natural Language Processing

Professor Carolyn Anderson

I. Abstract

This project investigates whether instilling inductive biases into large language models (LLMs) can improve their ability to represent and reason about complex historical events. We argue that effective event embeddings—capturing both semantic content and relational structure—are essential for advancing model understanding. Complex events, such as The Rebellion of 1923, consist of interdependent subevents, entities, and concepts, and are situated within broader causal chains (e.g., the assassination of Archduke Franz Ferdinand leading to World War I and subsequently the Paris Peace Conference). Capturing such semantic detail and causal dependency is critical for enabling socially grounded reasoning in LLMs.

To address this challenge, we propose a hybrid architecture that integrates language and graph modalities. Semantic information is derived from dimensionality-reduced DistilBERT embeddings, while relational dependencies are modeled using Graph Neural Networks (GNNs) and Knowledge Graph Embeddings (KGE). Through ablation experiments across seven representation settings (KGE, GNN, DistilBERT, KGE+DistilBERT, GNN+DistilBERT, KGE+GNN+DistilBERT, and random+DistilBERT), we evaluate performance on binary classification using an HGB classifier and on link prediction. Results demonstrate consistent improvements, with classification robustness across random seeds and a ~26% accuracy increase in link prediction.

This work points toward a deeper integration of computational methods with social science inquiry. By instilling inductive biases grounded in knowledge graphs, it offers a pathway to models that not only process language but also support the exploration of historical hypotheses and theories of societal change.

II. Introduction

Questions about how Language Models (LLMs) encode world knowledge are currently prevalent. Growing evidence indicates that LLMs acquire linear representations of space and time across various scales, showcasing their ability to structurally store temporal-spatial information. However, when prompted with factual knowledge, these models encounter challenges, such as hallucinations. To enhance LLMs' understanding of complex events and improve their reasoning skills, this research aims to focus on the fundamental level: event embeddings.

Two key characteristics are considered in the context of complex events: semantic information and relationship information. First, events such as "The Rebellion of 1923" comprise numerous subevents, entities, and concepts that are interconnected, forming a crucial part of the event embedding. Second, events exhibit correlation and causal connections with other events, emphasizing the importance of capturing dependency relationships in a suitable representation. For example, the event "*Assassination of Archduke Franz Ferdinand*" caused "*WWI*", which in turn caused "*The Paris Peace Conference*".

This research proposes a novel architecture that integrates language and graph modalities. Semantic information is stored in a dimensionality-reduced pre-trained DistilBERT embedding for sentences containing event keywords, while relational information between events is captured using Graph Neural Networks (GNN) and Knowledge Graph Embeddings (KGE) trained on the dataset.

I conducted ablation experiments to test which among the 7 representations (KGE, GNN, DistilBERT, KGE+DistilBERT, GNN+DistilBERT, KGE+GNN+DistilBERT, random+DistilBERT) perform with the best accuracy for two evaluation tests – one binary classification task using HGB classifier and a Link Prediction task. As a result, the proposed architecture that combines rich semantic information and interrelated causal relations demonstrates significant and consistent improvements in complex event representation. Classification based on the dataset that I constructed demonstrates a consistent improvement across each random seed test; the second evaluation through link prediction based on the synthetic negative edges sampled from the provided adjacency demonstrates a significant increase of around 26% accuracy score.

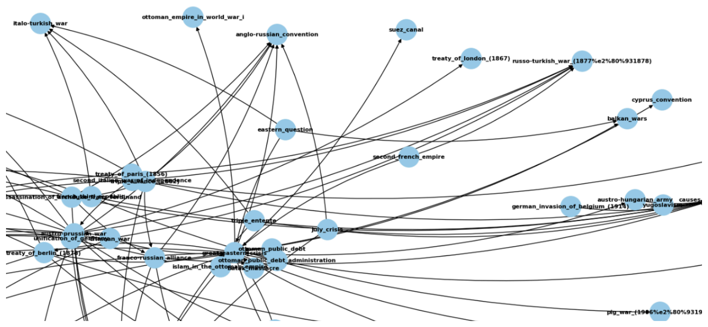
III. Data

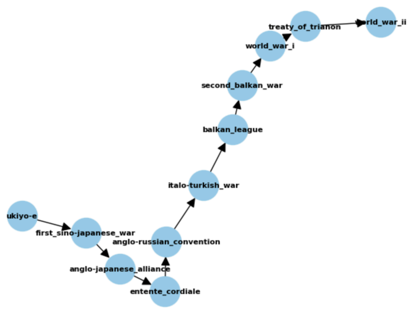
Wikipedia Knowledge Base: Despite the impossibility of quantitatively establishing real causal relationships, a vast amount of professional annotations is available online. Wikipedia was chosen as the foundation for knowledge graph construction. On a Wikipedia page for a specific event, denoted as A, sections such as "Introduction," "Cause" (or equivalents like "Background"), and "Consequences" (or equivalents such as "Aftermath") are considered highly relevant. Specifically, the "Introduction" contains many keywords for enriching the semantic meanings of an event. Furthermore, the "cause" section contains events that caused A, and the "consequence" section contains events that are caused by A. Therefore, causally-linked events centered around event A can be easily extracted in this manner.

Event Filtering through DistilBERT model: A complication arises when extracting events from the "Cause" and "Consequence sections." An example of a complex event is typically associated with hyperlinks on Wikipedia pages. However, not all terms with hyperlinks are events. For example, the "Assassination of Archduke Franz Ferdinand" includes real complex events like the "Secret Treaty of 1892" and "Austro-Hungarian ultimatum (23 July)," but also contains non-events like geographical terms such as "Danube" and country names such as "Germany." For the knowledge graph, only complex events as nodes are needed. Filtering out non-events is accomplished through a neural network for a classification task using DistilBERT pre-trained embeddings for inputs. The constructed neural network consists of 4 hidden layers with ReLU activation functions and an output softmax layer. The loss is calculated using negative-log-likelihood. After a small amount of labeling consisting of 150 events and 150 nonevents and feeding them in as training and testing data, the network produced a test accuracy of 96.6%.

Recursive Web-Scraping: The recursive web-scraping algorithm navigates each hyperlink, determining whether it represents a complex event through input into the neural network model. If classified as an "Event," the algorithm saves the keywords in its introduction into a corresponding CSV file and continues extracting more events until reaching a recursion depth of 4. With two starting events – WWI and WWII –, the algorithm obtained 20,165 intercorrelated events. I conducted several measures for data cleaning: I used RE to filter our irrelevant concepts and applied a second-time Event Classification network filtering. Furthermore, due to the noisy nature of web-scraped data, I detected a small amount of cycles in the adjacency list. Since cycle means invalid data in the context of event relations – if A causes B, B causes C, then C can never cause A – I cleaned up the nodes that are involved in the cycles by implementing cycle detection and removal algorithms. As a result, there are 19,221 intercorrelated events prepared for knowledge graph construction.

During the recursive web scraping process, keywords of each complex event from the event's introduction paragraphs were scraped and stored in CSV files to enrich the semantic meanings of the events. These keywords are later used as inputs for the language module and GNN node value.





III. Model

There are three major components in the architecture: DistilBERT Pretrained Sentence Embedding, trained KGE embeddings, and trained GNN embeddings.

Lower-Dimensional DistilBERT Sentence Embedding: I organized relevant "keywords" stored in each event corresponding CSV file as inputs and obtained the sentence embedding. The sentence embedding has a shape of (300,768) for each event because I have max_length set to 300 in the tokenizer 'sentence-transformers/stsb-distilbert-base'. Therefore, dim_reduction.py reduces the dimension of each event's DilstilBERT sentence embedding to (256,). The dimensionality reduction is important for increasing computation and retrieval speed; it is also particularly important for this architecture: the trained KGE embedding has a shape of (200,) and the GNN has a size of (64,). In order to prevent the semantic meaning module from being overly predominant in dictating the embeddings and balance out the weights that the model has for the three embeddings, dimensionality reduction is a crucial step.

Specifically, dim_reduction.py reproduces the "whitening" process developed by Su and etc. to reduce dimension. The author of the paper experimented on DistilBERT base & DistilBERT large and demonstrated that this methodology improved model performance on seven semantic textual similarity tasks, demonstrating that it greatly preserves semantic meanings while removing unnecessary correlations in the data, constructing semantically meaningful but denser vectors.

To make the data less redundant, the whitening algorithm obtains the average between the first and last layers of the pre-trained DistilBERT model to acquire relevant information syntactically and semantically. The compute_kernel_bias function processes vectors, calculates mean and covariance, applies SVD, and returns a transformation matrix and bias. Then, the transform_and_normalize function takes input vectors, a transformation matrix, and a bias, applying a linear transformation and L2 normalization to return normalized vectors. The normalization function independently standardizes input vectors. As a result of the transformation applied on the embeddings, the embeddings are reduced to shapes of (256,), which is more suitable for the architecture.

Trained KGE embeddings: The goal of KGE models is to embed the entities and relations including symmetric, antisymmetric, inversion, and composition into a continuous and low-dimensional vector space. Specifically, for event relationships, antisymmetric and composition relations are important to learn. For example, if eventA causes eventB, eventB causes eventC, then there are two conclusions one can draw: (1) eventB cannot cause eventA(antisymmetric); (2) eventA indirectly causes eventC (composition). The major component in KG training is the scoring layer: the scoring function assigns a score to the triple (event1, relation, event2), and the score should be higher if it is an existing statement. There are many developed score functions, including TransE, TransR, DistMult, ComplEx, RESCAL, and RotatE. Specifically, for the purpose of complex event representation learning, I applied ComplEx, which demonstrates better performance in capturing antisymmetric relations while maintaining a linear complexity. ComplEx uses the following scoring function: $h.T\ Re(diag(r)t)$. Given the scoring layer outputs, during training, KG samples synthetic fake statements and uses the loss function of NLL and the logistic loss returns -1 for negative samples and +1 for the positive samples.

I created a training dataset for KG training using ComplEx Model statement triples: I first wrote a script to extract chains from the cleaned adjacency list and constructed the triplets from the chains. After training for 300 epochs, reached a training loss of. The loss was at its minimum at around 300 epochs. The model achieves an 0.43 MRR, indicating that, on average, the reciprocal of the rank of the first correct prediction is 0.43. The average rank (MR) of the first correct prediction is 2177.93. The model achieves a Hits@10 score of 0.50, meaning that the correct prediction is included in the top 10 predictions 50% of the time. Hits@3: The Hits@3 score is 0.45, indicating that the correct prediction is included in the top 3 predictions 45% of the time. Hits@1: The Hits@1 score is 0.39, signifying that the correct prediction is included in the top-ranked prediction 39% of the time.

Trained GNN embeddings: KGE techniques encode the interactions between entities and relations through models that are not natively built for encoding graph structures. However, a novel family of neural architectures has been proposed to address this limitation and improve upon the performance. Therefore, I applied Graph Neural Networks (GNNs) to attempt to learn the latent representation of graph-structured data and enrich the graph modality. Furthermore, KG models only allow input data to be in the format of entity and relations as strings On the other hand, GNN allows additional information to be put into the training and learning process. Specifically, GNN also requires the dataset to have additional node features, for which I inputted the 265-dimensional BERT-sentence embedding that captured the keyword information of the complex event. This is an attempt to let the GNN learn the patterns given potentially correlated keywords and semantic information between the nodes so that it can leverage both the graph structure and node features for capturing relationships and patterns in the data.

I implemented an encoder-decoder structure for the GNN embedding learning. Specifically, the encoder is a neural network that maps nodes in a graph to low-dimensional vectors; the decoder maps the vectors back to the nodes. The encoder and decoder are trained jointly so that the vectors produced by the encoder can be used by the decoder to reconstruct the original graph. The loss function used to train the encoder-decoder is based on the idea that similar nodes should have similar embeddings. This loss function encourages the embeddings for similar nodes to be close together and the embeddings for dissimilar nodes to be far apart. The GNN undergoes optimization via iterative parameter updates to minimize the loss function, refining the neural network to produce embeddings that capture the semantic meanings. I trained for 200 epochs and decided to use the embeddings learned at epoch 30, where the loss stopped decreasing and started oscillating around 0.4338.

IV. Evaluation Metric

1. Classification on Hand-Constructed Dataset

Test Dataset Preparation: From the chains that I extracted from the adjacency graph, I created three distinct datasets: close pairs, middle pairs, and distant pairs. Close pairs are direct neighbors; middle pairs are indirect neighbors with a directed path <= 6 edges; distant pairs are indirect neighbors that are extracted from the two ends of one chain. There are 8438 close pairs, 8442 middle pairs, and 8442 distant pairs. To create negative examples, based on the asymmetry relationship of the events, I sampled from each of the three datasets and reversed the event ordering. To enhance the negative data, I also sampled unrelated events, which are events not from a consecutive chain, as the negative examples and have in total of 1777 pairs of unrelated events. I merged the unrelated events into the close_pair, distant_pair, and middle_pair and made them into JSON files.

Benchmarking against Llama and GPT3-turbo: I used the constructed dataset of positive and negative examples and conducted prompt engineering to test the performance of GPT3.5-turbo and Llama to use their results as a general benchmark against the new representation.

Specifically, I prompted GPT3-turbo with the following prompt: 'Did Event 1 directly or indirectly cause Event 2?' 'Event 1: {cause}, Event 2: {effect}. What is the Answer? Give me ONE WORD YES/NO RESPONSE!' GPT3-turbo achieves an accuracy of 51.2% across 2000 sampled data from the merged dataset of close pairs, middle pairs, and distant pairs. For Llama, due to Llama's inability to take from the instruction "give me a one-word response" and tendency for random generation. The prompt is as follows: 'I will give you some examples. Event1: Second World War, Event2: battle of the kerch peninsula \n Answer: Yes. \n. Event1: battle of the Kerch peninsula, Event2: second world war\nAnswer: No. \n. Event1: invasion of

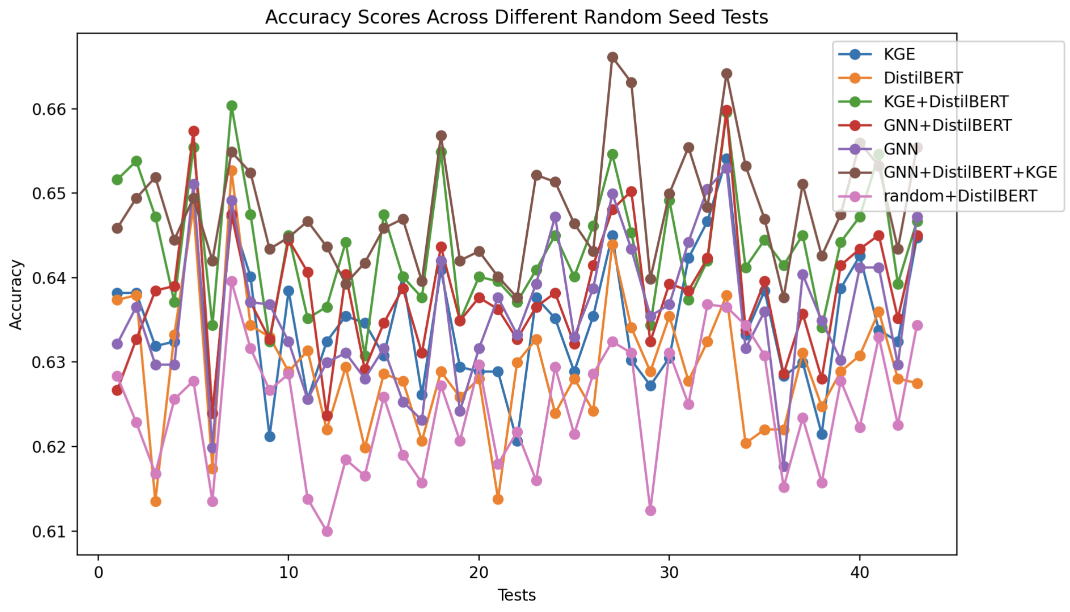
Sicily, Event2: palestine war\nAnswer: No. \n'.'Event1: second world war, Event2: battle of moscow\nAnswer: Yes. \n'. Llama achieved an accuracy of 0.49 in 2000 promptings.

Ablation Experiments: I am using the classifier HistGradientBoostingClassifier because HGB classifiers apply the concept of binning that is most relevant to Decision Tree classification algorithms by grouping with histograms; it is easy and very efficient to train. During test time, I split the dataset that is merged from close pair, middle pair, and distant pair using a test_size of 0.2, as I evaluated that this will generate around 50% of negative samples and 50% of positive samples. Across 100 tests, the average accuracy scores are as follows:

KGE(200): 0.6339331689947959
DistilBERTsentence(256): 0.6282936181867981
KGE + DistilBERTsentence(200+256): 0.642925225965489
GNN+DistilBERTsentence(64+256): 0.637518488085456
GNN(64): 0.6352999178307314
KGE+GNN+DistilBERT(520): 0.646296904957546
random264+DistilBERT(520): 0.6243577102163791

KGE+GNN+DistilBERT achieves the best average performance, KGE+DistilBERT is the second best and GNN+DistilBERT achieves the third. Across all test distributions of the accuracy score, the performance peaked with an accuracy of 0.6686, 0.6673, and 0.6652 using KGE+GNN+DistilBERT. Furthermore, KGE+DistilBERT also demonstrates a good performance with two peaks at 0.6605 and 0.6598 accuracy. GNN+DistilBERT also archives some good outcomes with one peak at 0.658. The results demonstrate DistilBERT enriched with graph embeddings, whether it is KGE, GNN, or KGE+GNN because at every single test, these three variations consistently outperform the classification given only the DistilBERT embeddings.

I also added a random vector of 265 + DistilBERT to make sure that it is not solely due to the increasing dimensionality that the classification model is able to be more expressive to capture nonlinear patterns. However, random265+DistilBERT performs the worst on average, which means that KGE and GNN are encoding structural and relational information that boosted the performance.
The following diagram samples 40 tests from the random seed tests to showcase the performances and the peaks:
Throughout my undergraduate years, I have conducted an extensive amount of research in the field of CV and NLP. Currently, I am writing two first-author papers and preparing for submission during winter. The first one is Denoising Triadic-Interactions for Robot Interjection Analysis. I led the project at the MIT Media Lab Personal Robotics Group, aiming to improve the robot's response in a parent-child interaction with robot interjection. I led the project to develop a multimodal algorithm yoloDeepface and integrated algorithms including distance interpolation and memory loading to clearly differentiate parent and child from others in complex videos and achieve 95% accuracy. We are preparing for submission to ICMI 24'. The second first-author publication is BertKG-ComplexEvent Representation Learning. I proposed a novel architecture combining language and graph modalities to enhance LM with a comprehensive understanding and logical inference skills. I am preparing paper submissions to computational social science conferences and journals.

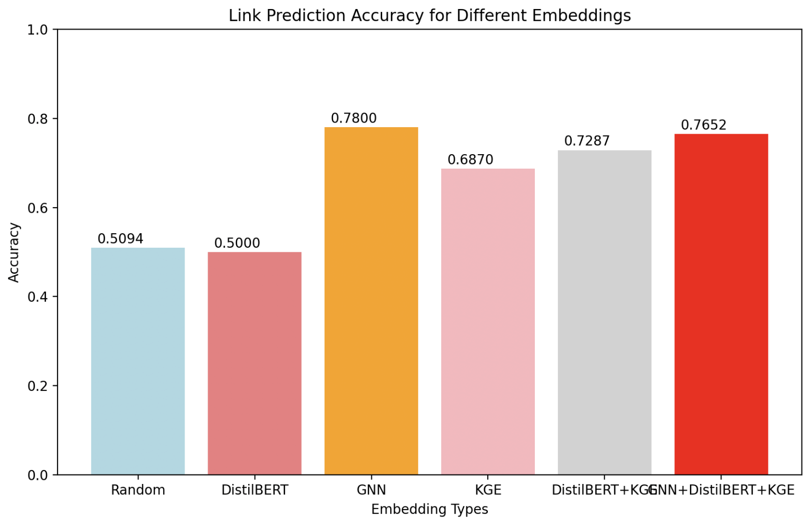


The classification task demonstrates that the graph-enhanced representation has consistent improvement. However, it does not demonstrate a significant improvement in terms of accuracy score difference.

2. Link Prediction Performance

Link Prediction asks the system whether there are potential linkages (edges) between nodes. Specifically, using the encodings generated through the training, the decoder component makes link predictions (i.e. binary classifications) on all the edges including the negative links using node embeddings. It calculates a dot product of the node embeddings from a pair of nodes on each edge. Then, it aggregates the values across the embedding dimension and creates a single value on every edge that represents the probability of edge existence.
Ablation Experiments: For the ablation experiments, I extracted the encodings trained by the GNN and replaced them with several variations before feeding them into the decoder module, specifically including BERT, GNN, KGE, BERT+KGE, BERT+GNN, GNN+KGE+DistilBERT. The results demonstrate significant improvement: the decoder that uses DistilBERT embeddings achieves around 50% accuracy, while the concatenated GNN+DistilBERT+KGE achieves an accuracy of 76.5%, which is a 26.5% increase that demonstrates significant improvement. Interestingly, GNN alone achieves 78%, which is roughly the same performance as the proposed GNN+DistilBERT+KGE representation by 1.5%. The competitive performance is reasonable considering that the GNN itself is enhanced with rich semantic meanings because the node features are the DistilBERT reduced sentence embeddings.
I also benchmarked the result with an embedding of random-(128,)-vector+DistilBERT to make sure that it is not solely due to the increasing dimensionality that the classification model is able to be more expressive to capture nonlinear patterns. However, random+DistilBERT obtains an

accuracy of around 50.9%, which means that the graph module encodes structural and relational information that boosts the performance.



V. Conclusion

In conclusion, the proposed architecture that combines the rich semantic information and the interrelated causal relations demonstrates significant and consistent improvements in complex event representation. Classification based on the dataset that I constructed demonstrates a consistent improvement across each random seed test; the second evaluation through link prediction based on the synthetic negative edges sampled from the provided adjacency demonstrates a significant increase of around 26% accuracy score. The results showcased the significance of combining language and graph modules to obtain a better representation of complex events.

The discrepancies between the two evaluation schemes are worth further investigations; specifically, the first classification evaluation demonstrates a 2% accuracy increase, while the link prediction demonstrates a 26% accuracy increase. The only thing different was that my dataset consisted of a good portion of unrelated event pairs instead of merely "reversed edges" – B to A when A actually causes B. Therefore, it is plausible that GNN learns the antisymmetric relations well but is less clear when classifying the relations between unfamiliar nodes. I believe conducting more experiments using the close pair, middle pair, and distant pair datasets separately may be able to give interpretable outcomes to this discrepancy,

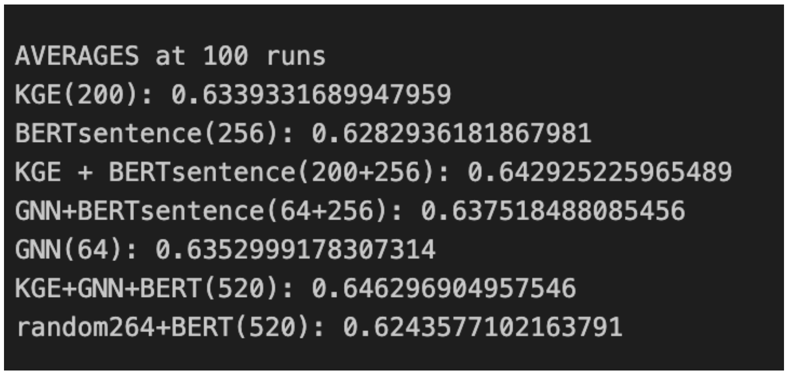
Furthermore, I attempted to conduct clustering experiments to uncover "how" these concatenated language and graph representations are structured in the high-dimensional space. I specifically wanted to investigate the underlying reason for how the structures allow improvement in the classification and link prediction performance because the benchmark tests against random+DistilBERT demonstrate that there is inherent structural information that has been encoded when enhanced with the graph module. Specifically, I implemented Hierarchical DBSCAN clustering and applied t-SNE to visualize the result. However, I was unable to capture meaningful clusters and the data seemed disproportionate. The absence of meaningful clusters in Hierarchical DBSCAN and t-SNE visualization may be due to the sensitivity of DBSCAN to noise and outliers, potential issues with data scaling affecting DBSCAN's performance, limitations in capturing meaningful relationships when dealing with high-dimensional data in t-SNE. Further research would need to be conducted to further investigate the embedded structure in the high-dimensional space.

References

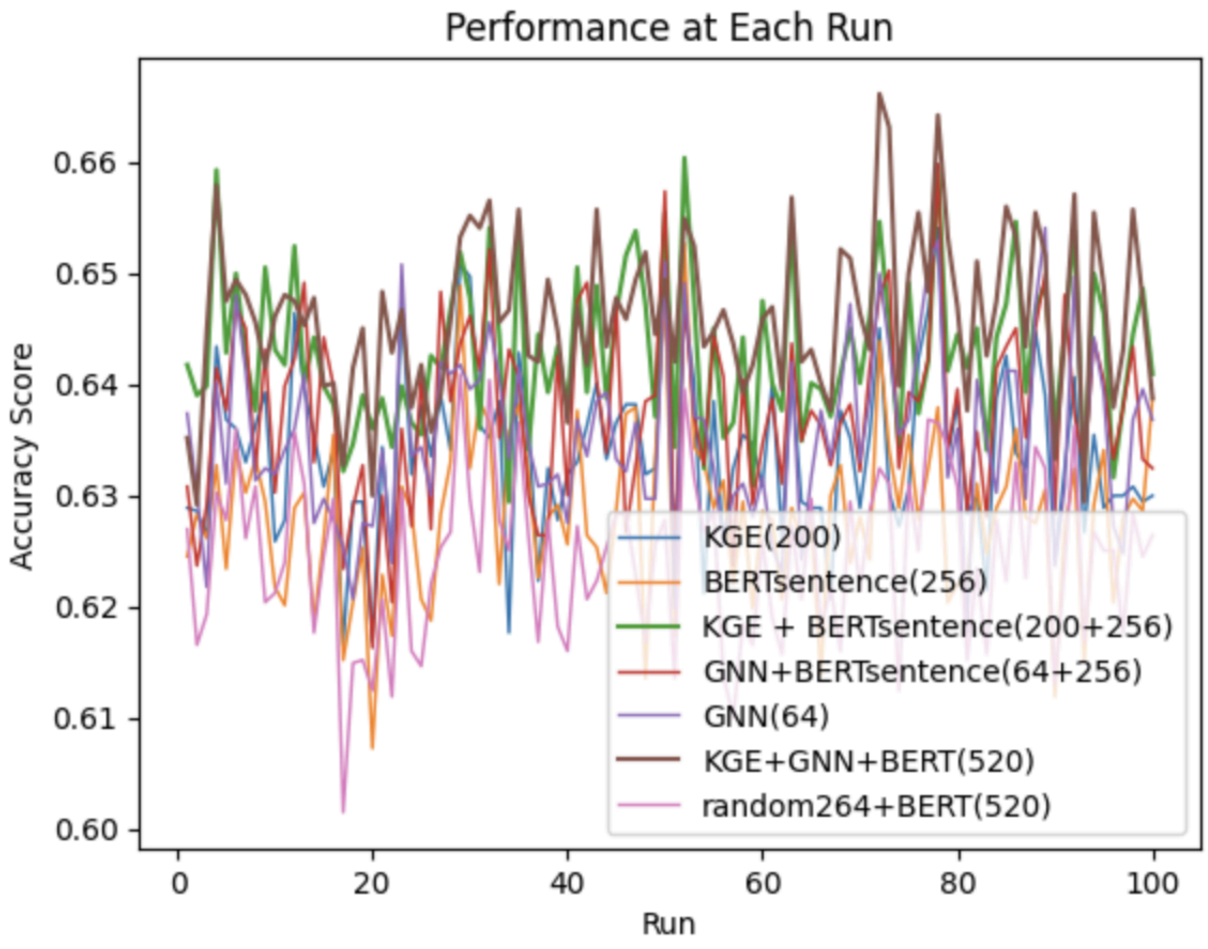
T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard. "Complex Embeddings for Simple Link Prediction." In Proceedings of the International Conference on Machine Learning (ICML), 2016. arXiv:1606.06357 [cs.AI]. DOI: 10.48550/arXiv.1606.06357

J. Su, J. Cao, W. Liu, and Y. Ou. "Whitening Sentence Representations for Better Semantics and Faster Retrieval." arXiv:2103.15316 [cs.CL], 2021. DOI: 10.48550/arXiv.2103.15316

W. Gurnee, M. Tegmark. "Language Models Represent Space and Time." arXiv:2310.02207 [cs.LG], 2023. DOI: 10.48550/arXiv.2310.02207



```
AVERAGES at 100 runs
KGE(200): 0.6339331689947959
BERTsentence(256): 0.6282936181867981
KGE + BERTsentence(200+256): 0.642925225965489
GNN+BERTsentence(64+256): 0.637518488085456
GNN(64): 0.6352999178307314
KGE+GNN+BERT(520): 0.646296904957546
random264+BERT(520): 0.6243577102163791
```



The peaks are never the random / BERTsentence embeddings.

Conclusion

- A small set of 43000 events. But this webscraping

Events in the human history are linked with casualty and correlation. For example, WWI leads to WWII. Essentially, domain-specific professional knolwge capture these relationships and the research aims to enhance LLMs the ability to store these causal relationships.
I constructed a causally-linked event graph through ML-backboned recursive web-scraping from Wikipedia. Contextualized on the dataset, CoT prompting demonstrated that Llama and GPT-3.5 have ~50% accuracy at causal prediction tasks. In order to mitigate such deficiency for LMs, I proposed a novel architecture combining language and graph modalities: the semantic information is stored in a dimensionality-reduced pretrained DistilBERT embedding for a sentence consisting the event's keywords; the relational information between events are captured by the GNN and KGE that I trained on the dataset. I used a graph link prediction task for evaluation: the multimodal representation achieved a 78% accuracy rate in differentiating true versus synthetic fake causal relations, outperforming the DistilBERT representation by 28%.

Data

I constructed a causally-linked event graph through ML-backboned recursive web-scraping from Wikipedia.

Model

Metric

Results

Conclusion

References

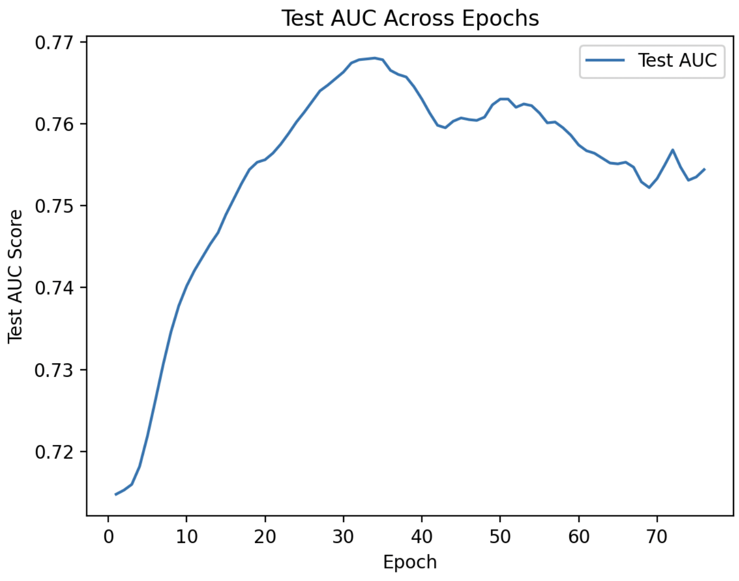
Future

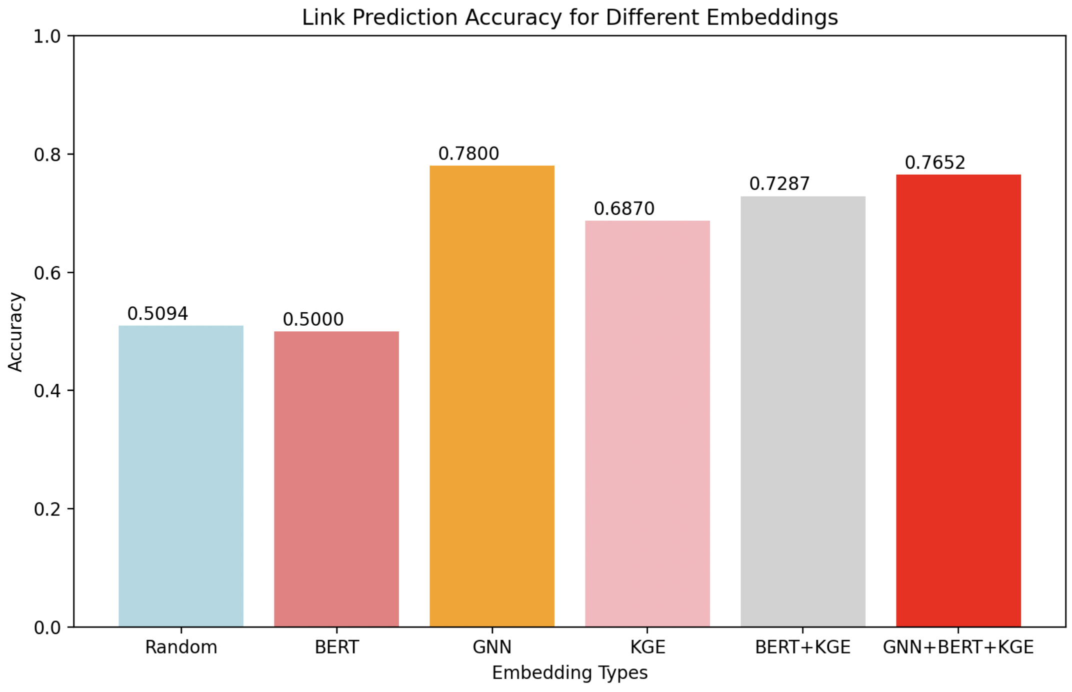
GRAPHS:

Formula for ComplX:

$$f_r(h,t) = Re(h^{\top} diag(r)\bar{t}) = Re(\sum_{i=0}^{d-1}[r]_i \,.\, [h]_i \,.\, [\bar{t}]_i)$$

GNN training graph:





Python Files Needed:
1.Prompt GPT and Llama
2.make_dataset.py: this IV. make test set

CSV files needed