Predicting Interactions in the Weapons of Mass Destruction Knowledge Graphs

Abhigya Agrawal, Md Saidul Hoque Anik, and Ariful Azad

Department of Intelligent Systems Engineering
Indiana University, Bloomington, USA
abagra@iu.edu, mdshoque@iu.edu, azad@iu.edu

Abstract. In this paper, we apply graph machine learning methods to predict unseen interactions within the Weapons of Mass Destruction (WMD) dataset, developed by DARPA and IARPA. This dataset captures complex online activities, including sales, purchases, and forum discussions, with a focus on topics such as weapons, explosives, and other sensitive subjects. We represent the data as a knowledge graph, where nodes correspond to entities and edges denote relationships between them. Among various knowledge graph embedding techniques and graph neural networks, semantic matching models like DistMult demonstrate the ability to accurately predict 84% of relations, particularly due to their strength in capturing the one-to-many relationships common in the WMD data. To streamline the analysis, we implement an automated pipeline that stores the knowledge graph in a Neo4j database, extracts subgraphs using Cypher queries, trains knowledge graph embedding models on these subgraphs, predicts links, and reintegrates high-confidence edges back into the main graph.

Keywords: Knowledge graph, embedding, online forum

1 Introduction

In today's digital landscape, online forums play an important role in facilitating social and economic transactions [12, 7]. Predicting future interactions based on historical forum activities has significant applications in areas such as advertising, recommendation systems, and cybersecurity [20, 21]. For instance, security agencies monitor and analyze forum posts, comments, and interactions to detect suspicious activities. In line with these objectives, the U.S. Defense Advanced Research Projects Agency (DARPA) compiled an extensive dataset with online sales, purchases, and forum discussions, including those related to weapons, explosives, and other sensitive topics[1]. This dataset is referred to as the Weapons of Mass Destruction (WMDs) dataset. This paper explores machine learning (ML) algorithms to predict future interactions including potential threats from these multi-modal WMD datasets. Security agencies can take proactive measures to prevent harm and save lives by accurately identifying future activities.

[1] https://www.darpa.mil/program/modeling-adversarial-activity

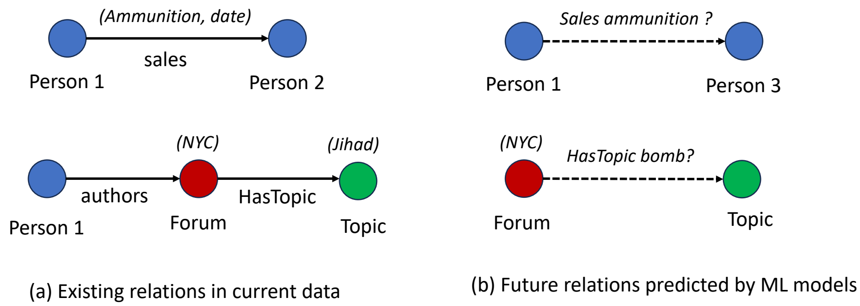

Fig. 1. (a) Examples of sale, author, and HasTopic relations in the WMD data. (b) The top right prediction task might assess whether Person 1 is likely to sell ammunition to Person 3, based on the fact that Person 1 previously sold ammunition to Person 2. Similarly, the bottom right prediction might evaluate whether a forum that discusses topics related to New York City will eventually discuss the topic of bombs. These predictions leverage patterns observed in historical data to anticipate future interactions.

One of the key challenges in predicting future activities from the WMD data is its multi-modal and unstructured nature. The data captures entities, including individuals, products, topics, publications, forums, and their interactions. It also includes both text and numerical descriptions with temporal aspects. To address this complexity, we use a knowledge graph (KG) to model the data. A KG organizes information into a network of nodes (representing entities) and edges (representing relationships), with entities and relations having additional features [6]. Modeling the WMD data as a knowledge graph offers two main advantages. First, it provides a structured representation of the data, which is stored in a graph database, allowing for easy retrieval and augmentation with additional information. Second, it enables us to approach future interaction predictions as a link prediction problem, a well-established technique in the literature.

Over the last decade, the link prediction problem has been solved by various graph embedding algorithms [5, 13, 17], graph neural networks [8, 18], and knowledge graph embedding (KGE) models [1, 9]. Given the extensive research in this area, we do not aim to develop new ML methods for link prediction on the WMD knowledge graph. Instead, we focus on identifying the most suitable models for predicting future transactions and understanding the underlying reasons for their observed performance. To this end, we employed a classic node embedding algorithm, node2vec [5], the Graph Convolutional Network (GCN) [8], and several KGE models. We found that certain classes of KGE models, particularly those capable of capturing one-to-many relations, performed well in predicting new relationships within the WMD knowledge graph. This is likely due to the prevalence of one-to-many relations in the dataset, such as sale (a person sells multiple products), author (a person authors several topics), etc. In contrast, node2vec and GCN did not perform as well as the top KGE models, primarily because of their limitations in capturing the types of relations in our dataset.

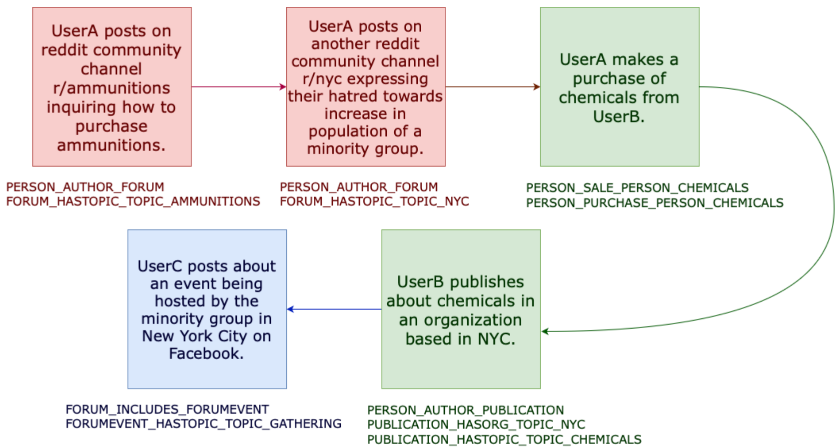

Fig. 2. Examples in WMD dataset and their corresponding triples.

Based on our experiments, this paper makes the following contributions. (1) We utilize a knowledge graph model to represent multi-modal WMD data from DARPA. (2) We compare various supervised and unsupervised embedding algorithms to predict new interactions within the WMD knowledge graph. Based on these experiments, we identified specific KGE models that effectively capture the unique relations in the data. (3) To automate the analysis, we develop a comprehensive pipeline that stores the knowledge graph in a Neo4j database, extracts subgraphs via Cypher queries, trains KGE models on subgraphs, predicts links, and reintegrates high-confidence edges back into the original graph.

2 Method

2.1 The Weapon of Mass Destruction (WMD) Social Data

The WMD data generated by the Modeling Adversarial Activity (MAA) Program in DARPA is prepared to help develop mathematical and computational techniques for modeling adversarial activity. This multi-modal data captures online sales/purchase activities including weapons, explosives, and fertilizers as well as forum discussions on various topics including bombing, jihad, and specific locations. Fig. 2 provides several examples of the data content and illustrates how it can be structured into knowledge graph triples.

2.2 Representing Online Transactions as Knowledge Graphs

A knowledge graph representation of the data is the best way to achieve our goal of predicting forthcoming interactions. WMD data contains five types of entities: Person, Forum Events, Forum, Publication, and Topic. These entities are linked to each other through different types of relations as shown in Fig. 3. A ‘Sale’ or

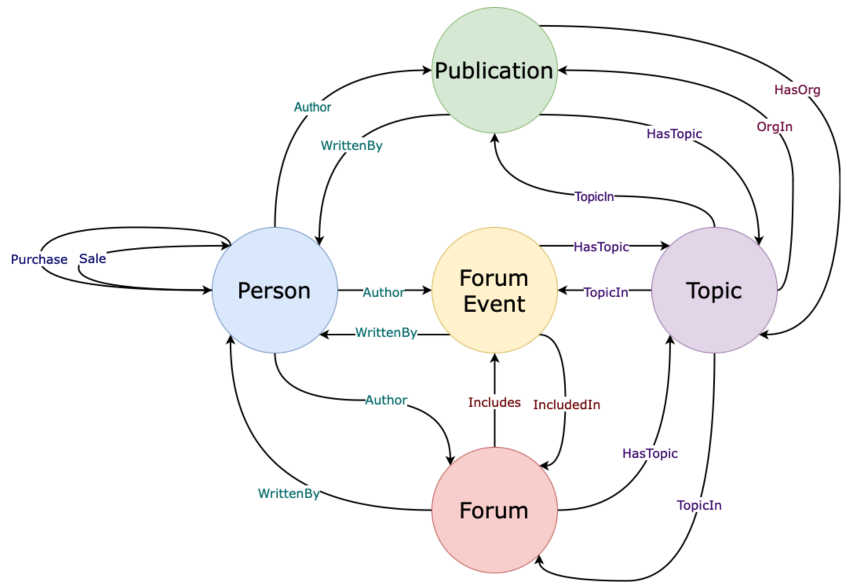

Fig. 3. Visualization of the WMD dataset entities and relations.

a 'Purchase' occurs between two persons, and these relations are inverse of one another. and are of the type 1-to-many. A person can be an 'Author' of Publications, Forums, and Forum Events. Publications, Forums, and Forum Events are 'WrittenBy' a Person. 'Author' and 'WrittenBy' relations are also inverse relations. A person can be an author for multiple Publications, Forums, and Events and vice versa. Hence both 'Author' and 'WrittenBy' relations are 1-to-many relations. Publications, Forums, and Forum Events will have certain Topics they discuss and are linked to Topics via the relation 'HasTopic'. 'HasTopic' and 'TopicIn' are inverse to each other and are 1-to-many relations. A Forum Event is 'IncludedIn' a Forum and a Forum 'Includes' a Forum Event. These relations are inverse of each other and one Forum can include multiple Forum Events and one Forum Event can be present in multiple Forums, making them 1-to-many relations. A Publication has an organization in Topic that is represented by the relation 'HasOrg'. 'OrgIn is the inverse of 'HasOrg' relation and they both follow the 1-to-many relation type. The original data file contains five types of relations, their inverse relations are added to the data while pre-processing. These five relations are: 'Sale', 'Author', 'Includes', 'HasTopic', and 'HasOrg'. These relations will be referred to as the main five relations in the following sections.

2.3 The Overall Software Framework

Fig. 4 shows the software framework used to train knowledge graphs retrieved from the Neo4j database. We rely on the TorchKGE library for the training and testing of KGE models. Our framework has configurable training and inference

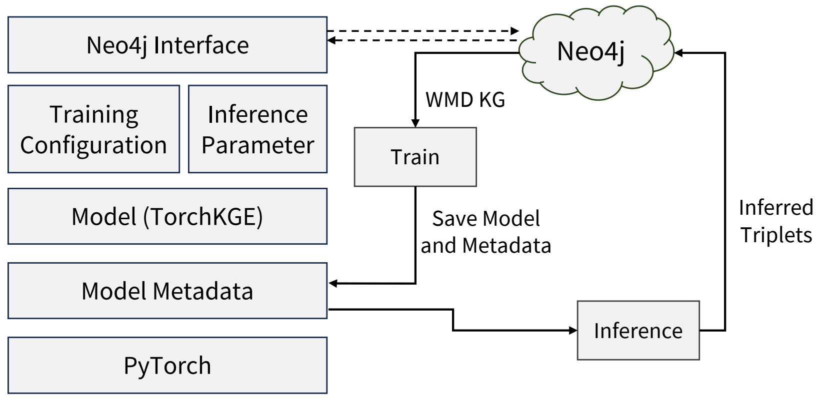

Fig. 4. Software framework on the left and the training and inference pipelines on the right.

pipelines. The training component can fetch triplets from the Neo4j database and perform model training to generate embeddings for entities and relations. After the training is finished, it stores the generated embeddings and model metadata locally. The inference component uses these model artifacts to create new facts and assign scores to them. The inferred triplets are stored locally. The user can upload any number of these triplets back to the Neo4j database. We deployed these pipelines on a compute cluster, allowing users to apply them to any knowledge graph for both training and inference purposes.

2.4 Predicting Interactions by KG Embedding

One of the key components of our data analysis pipeline is knowledge graph embedding (KGE). Most KGE models represent the head, relation, and tail of a triple by embedding vectors in Euclidean space such that a combination of head and relation embedding vectors results in the tail embedding vector. The score function of a knowledge graph embedding model measures how distant are two nodes compared to the relation between them. KGE models are trained to reduce the score between the combination of head and relation embedding and tail embedding. These embeddings are evaluated by testing their performance on knowledge graph completion tasks. Knowledge graph completion task includes predicting the unseen relations r between two existing entities: (h,?,t). This task is also known as a link prediction task or interaction prediction task. Since KGE is a well-studied problem in the literature, we do not aim to develop new models. Instead, we compare two classes of KGE models for WMD data.

The first class of models are called translational models that aim to reduce the Euclidean distance between head + relation embeddings and tail embeddings. Different types of translational models opt for different representations of head, relation, and tail embeddings. TransE [1] represents head, relation, and tail in the same semantic space of $\mathbb{R}^d$. Since both relations and entities are in

Table 1. Types of relations supported by different KGE models.

Model	Symmetric	Anti-Symmetric	Inversion	Composition	1-to-N
TransE	No	Yes	Yes	Yes	No
TransH	Yes	Yes	Yes	Yes	Yes
TransR	Yes	Yes	Yes	Yes	Yes
DistMult	Yes	No	No	No	Yes
RESCAL	Yes	Yes	Yes	Yes	Yes
ComplEx	Yes	Yes	Yes	Yes	Yes

the same dimensional space, TransE is unable to capture symmetric and 1-to-many relations. TransR [9] proposes a solution for TransE's drawbacks by having two different spaces for entities and relations. The entities in space $\mathbb{R}^k$ are projected to the relation space $\mathbb{R}^d$ by a projection matrix $M_r \in \mathbb{R}^{kxd}$. TransR can capture different kinds of relations at the cost of performance. TransH [19] proposes a method to perform translation on the hyperplane. Every relation has a relation-specific hyperplane represented by w_r. The head and tail vectors h, t are projected to the relation-specific hyperplane by w_r using $h_\perp = h - w_r^\top h w_r$ and $t_\perp = t - w_r^\top t w_r$. These projected head and tail vectors are then connected via the relation-specific translation vector d_r or just r.

The second model class are called semantic matching models that use a bilinear score function. DistMult [22], RESCAL [10] and ComplEx [16] are examples of semantic matching models. RESCAL represents each entity with a vector and each relation as a matrix to capture the latent semantics of a knowledge graph. The score function (h,r,t) is represented by the vectors h and t in R^d and the relation matrix M_r in $\mathbb{R}^{d\times d}$. The score function captures pairwise interaction between all instances of entities h and t through M_r. RESCAL uses tensor decomposition to solve the equation. RESCAL can be computationally complex and expensive as each relation requires $O(d^2)$ parameters. DistMult [22] proposes a solution to reduce the computational complexity of RESCAL by adding a constraint to the relation matrix M_r to be a diagonal matrix. This constraint also adds a limit to DistMult that the model can only capture symmetric and 1-to-many relations efficiently. The ComplEx model aims to perform embedding with lesser parameters while capturing anti-symmetric relations.

Table 1 shows the types of relations each KGE model can effectively capture. Different models excel at modeling various relation types. For example, TransE [1] performs well with anti-symmetric, inversion, and composition relations, while DistMult [22] is effective at modeling symmetric and 1-to-N relations. Our experiment examines these capabilities, evaluating how well each KGE model captures the diverse relationships present in the WMD knowledge graph.

2.5 Predicting Interactions by GNNs

Graph neural networks (GNNs) [15] are deep learning models that are designed to work with graphs. A knowledge graph can be viewed as a heterogeneous graph

Table 2. Dataset statistics

Dataset	Entities	Relations	Triples
WMD Small	105932	10	344946
WMD Large	189164	10	699828
WMD Small Mod	105932	6	189598
WMD Large Mod	189164	6	384155

with different types of entities and relations. Graph neural networks work on the principle of message passing between two neighboring nodes and combining this information received from their neighbors. Graph convolution networks (GCNs) [8] are a popular type of graph neural networks. Similarly to traditional convolution neural networks, GCNs also operate by convoluting or aggregating the information from the neighboring nodes of the target node. GNNs are widely used to analyze data in a graph structure. They are most often used for node classification and link prediction tasks. In this paper, we focus on the link prediction capabilities of GCN and compare its performance with knowledge graph embedding models' link prediction. Additionally, we run experiments with the embeddings generated by knowledge graph embedding models as features for graph neural networks.

3 Results

3.1 Experimental Settings

Datasets. In this paper, we use four versions of the WMD dataset, each generated using a simulator that creates realistic data based on the actual ontology shown in Fig. 3. The simulator was made available by the IARPA AGILE program [2] to mimic real-world scenarios while ensuring data security and privacy. We used the WMD Small dataset to train KGE models and tune their parameters. The WMD Large dataset is almost double the size of the WMD Small dataset in terms of the number of triples. WMD Small Mod dataset is a modified version of the WMD Small dataset with the five main relations: 'Sale', 'Author', 'Includes', 'HasTopic', and 'HasOrg' and an additional 'Purchase' relation. Similarly, the WMD Large Mod dataset is a modified version of the WMD Large dataset. Table 2 summarizes these datasets.

KGE models. We used six KGE models to predict new relations from the WMD data. These models can be categorized into two groups: Translational models and semantic matching models. Translational models include TransE [1], TransR [9], and TransH [19], which aim to represent relationships as translations in the embedding space. On the other hand, semantic matching models such as RESCAL, DistMult, and ComplEx focus on learning entity and relation embeddings based on similarity in the latent space.

[2] https://www.iarpa.gov/research-programs/agile

Performance metrics. Knowledge graph embedding models calculate scores for positive and negative triples using the model's scoring function. A positive triple represents a fact that exists in the knowledge graph, while a negative triple represents a hypothetical fact that does not exist in the graph. To generate the negative triples, either the head entity or the tail entity of a given triple is corrupted and a link prediction task is performed. The metrics used to evaluate these results are Hits@k, MRR, and MR. Hits@k denotes the fraction of positive triples that rank in the top-k among their negative triples. A higher Hits@k score is better. Hits@1 score can be equated to accuracy, where accuracy is defined as the number of positive triples predicted. KGE models assign higher scores to positive triples and lower scores to negative ones, thereby distinguishing between valid and invalid relationships within the knowledge graph. Triples can be ranked according to their scores, where positive triples should ideally receive higher scores than negative triples Marginal Rank or MR is an evaluation metric that marginalizes over the rank of all possible positive triples for the (h, r) or (r, t) pair. For a given pair (h, r) and a corresponding set of tails (t) that represent all tails that form positive triples and (t_{-}) that represent all the tails that form negative triples with (h, r) in the dataset, Marginal Rank will be calculated by taking an average of the ranks for all (h, r, t). MRR or Mean Reciprocal Rank is calculated by taking the mean of the reciprocal of the ranks. Ideally, a lower Marginal Rank and a higher Mean Reciprocal Rank is desired.

Experiment platforms. In this paper, all the knowledge graph embedding models were trained using TorchKGE python library [2]. The graph neural networks were implemented using the PyTorch-geometric python library [4]. Both of these libraries build upon the PyTorch framework [11] that provides capabilities to define, train, test, and evaluate complex deep learning models. One GPU node instance of NVIDIA A100 GPU [3] is used to run all the code.

3.2 Predicating new relations in the WMD knowledge graphs

In this experiment, our goal is to predict missing relationships (edges) between entities (nodes) in the WMD knowledge graph. As discussed in the methods section, we formulate this as a recommendation problem $(h, r, ?)$ where we aim to predict the tail entity given a head h and relation r. Table 3 compares the results from six KGE models coupled with two loss functions.

We used two loss functions for training these models. The margin loss function [1] works by maximizing the distance between positive and negative triples, ensuring that positive triples are scored significantly higher. On the other hand, the Binary Cross Entropy (BCE) loss [14] outputs a probability, aiming to increase the likelihood of positive triples while reducing the probability of negative ones.

Table 3 shows that these loss functions give drastically different results when used with the same model. DistMult and RESCAL are both semantic matching models and give their best performance when combined with the BCE Loss function. TransE and TransH are both translational models and give their best performance when combined with Margin loss. TransR requires $O(dk)$ parameters

Table 3. Knowledge graph embedding model performance comparison on the WMD Small dataset.

Method	Loss	MRR	MR	Hits@1	Hits@10
TransE	Margin	0.336	2604	0.089	0.654
TransE	BCE	0.236	2180	0.196	0.301
DistMult	Margin	0.392	2979	0.331	0.495
DistMult	BCE	0.79	1995	0.767	0.838
RESCAL	Margin	0.345	3159	0.284	0.45
RESCAL	BCE	0.489	2828	0.379	0.674
TransR	Margin	0.133	11967	0.008	0.31
TransR	BCE	0	30933	0	0
TransH	Margin	0.337	2645	0.087	0.66
TransH	BCE	0.229	2205	0.188	0.298
ComplEx	Margin	0.216	3549	0.15	0.361
ComplEx	BCE	0.549	3266	0.552	0.619

per relation, and this data has ten types of relations, making TransR computationally complex. This results in the worst performance among all the models, even reaching a Hits@k score of 0 with BCE loss. Nevertheless, we observed that DistMult combined with BCE Loss gives the best performance for Hits@k and reaches up to a 0.838 score for Hits@10 and a 0.767 score for Hits@1. Hits@1 can be considered equivalent to an accuracy metric. This will be useful in comparing KGE results with those obtained from graph neural networks. DistMult with BCE also outperforms all the other models in both MRR and MR metric with a 0.79 score for MRR and 1995 rank for MR.

3.3 Relation-wise performance of KGE models

To assess the performance of various KGE models, we analyzed their ability to predict different types of relations, as shown in Fig.5. DistMult outperforms other models for all relations except "Includes," largely because it excels at capturing 1-to-many relations like "Sale," "Author," and "HasTopic," as indicated in Table1. RESCAL performs best for the "Includes" relation due to its flexibility in modeling various relation types. Since 1-to-many relations dominate the dataset, models like DistMult and RESCAL perform the best. However, performance may vary depending on the dataset, so testing multiple models is recommended.

3.4 Comparison of KGE with node embedding and GNNs

When a knowledge graph is simplified by ignoring entity and relation types, traditional graph embedding methods and graph neural networks (GNNs) can be used for edge prediction. In our experiment, we applied the node2vec embedding algorithm [5] and trained a logistic regression classifier for link prediction. We also trained a Graph Convolutional Network (GCN) [8] for the same task.

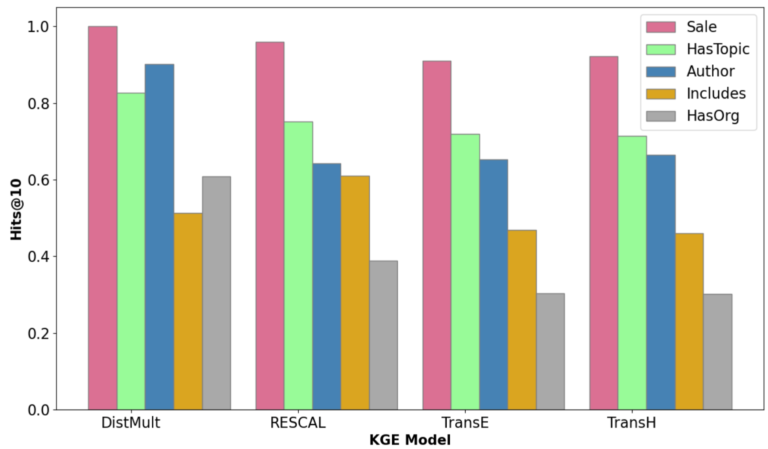

Fig. 5. Relation-wise performance of KGE models.

Table 4. Link Prediction with different methods on WMD Small dataset.

Method	Final Accuracy
DistMult Link Prediction	0.767
GCN Link Prediction	0.7029
Node2Vec + Logistic Regression Link Prediction	0.6884
DistMult Emb + GCN Link Prediction	0.6971

Table 5. Knowledge graph embedding model performance for different sizes of data

Dataset	MRR	MR	Hits@1	Hits@10
WMD Small	0.79	1995	0.767	0.838
WMD Large	0.763	2722	0.676	0.811
WMD Small mod	0.255	11907	0.252	0.307
WMD Large mod	0.251	19554	0.211	0.324

Accuracy was used for GCN and logistic regression, while Hits@1 was used for KGE models. As shown in Table 4, the DistMult KGE model outperformed both node2vec and GCN. Using DistMult embeddings as node features for the GCN did not improve its performance, indicating that KGE models are more effective in capturing entity and relation information that GCN models overlook.

3.5 Scaling and Privacy

The WMD dataset contains sensitive information and ideally, we want to be able to train on a small portion of the dataset and reproduce those results for a larger dataset. To test the scaling capability of KGE models, we ran experiments for WMD Small and WMD Large datasets, as well as for WMD Small mod and WMD Large mod datasets. The results for these experiments are shown in Table 5. The purpose of these experiments is to check if the performance is consistent across different dataset sizes. The DistMult model with BCE loss function is used to train all datasets. The different-sized datasets do show a consistent performance for the Hits@10 metric and Marginal Rank metric. The experiments conclude that to protect the privacy of the data, the knowledge graph embedding models can be trained on a smaller portion of the data and the results can be scaled to a larger dataset size.

4 Conclusions

We developed a software framework for storing and analyzing DARPA's WMD knowledge graph. This framework incorporates various KGE models, graph embedding algorithms, and GNNs to predict new relationships. The best-performing model, DistMult, accurately predicts 84% of relations, outperforming other KGE models, node2vec, and GCNs. DistMult's success is due to its ability to capture the one-to-many relationships prevalent in the WMD data. The analyzed WMD knowledge graph includes typical online activities like sales, purchases, and discussions, offering insights applicable to similar online forum datasets.

5 Acknowledgements

This research is based upon work supported by the Intelligence Advanced Research Projects Activity (IARPA), through the Advanced Graphical Intelligence Logical Computing Environment (AGILE) research program, under Army Research Office (ARO) contract number W911NF22C0084. This work is also supported by the NSF grant OAC-2339607.

References

1. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Weinberger (eds.) Advances in Neural Information Processing Systems, vol. 26. Curran Associates, Inc. (2013)
2. Boschin, A.: Torchkge: Knowledge graph embedding in python and pytorch. In: International Workshop on Knowledge Graph: Mining Knowledge Graph for Deep Insights (2020)
3. Choquette, J., Gandhi, W., Giroux, O., Stam, N., Krashinsky, R.: Nvidia a100 tensor core gpu: Performance and innovation. IEEE Micro **41**(2), 29–35 (2021)
4. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019)
5. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 855–864 (2016)

6. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., et al.: Knowledge graphs. ACM Computing Surveys (Csur) **54**(4), 1–37 (2021)
7. Jin, L., Chen, Y., Wang, T., Hui, P., Vasilakos, A.V.: Understanding user behavior in online social networks: A survey. IEEE communications magazine **51**(9), 144–150 (2013)
8. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017)
9. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. Proceedings of the AAAI Conference on Artificial Intelligence **29**(1) (2015)
10. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11, p. 809–816. Omnipress, Madison, WI, USA (2011)
11. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library (2019)
12. Pendry, L.F., Salvatore, J.: Individual and social benefits of online discussion forums. Computers in Human Behavior **50**, 211–220 (2015)
13. Rahman, M.K., Sujon, M.H., Azad, A.: Force2vec: Parallel force-directed graph embedding. In: 2020 IEEE International Conference on Data Mining (ICDM), pp. 442–451. IEEE (2020)
14. Ruffinelli, D., Broscheit, S., Gemulla, R.: You can teach an old dog new tricks! on training knowledge graph embeddings. In: International Conference on Learning Representations (2020)
15. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Transactions on Neural Networks **20**(1), 61–80 (2009). DOI 10.1109/TNN.2008.2005605
16. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, p. 2071–2080. JMLR.org (2016)
17. Tsitsulin, A., Mottin, D., Karras, P., Müller, E.: Verse: Versatile graph embeddings from similarity measures. In: Proceedings of the 2018 world wide web conference, pp. 539–548 (2018)
18. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
19. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. Proceedings of the AAAI Conference on Artificial Intelligence **28**(1) (2014)
20. White, R.W., Bailey, P., Chen, L.: Predicting user interests from contextual information. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pp. 363–370 (2009)
21. Wong, G.K., Li, Y.K., Lai, X.: Visualizing the learning patterns of topic-based social interaction in online discussion forums: An exploratory study. Educational Technology Research and Development **69**(5), 2813–2843 (2021)
22. Yang, B., tau Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases (2015)