MOOC Recommendation System Based on Knowledge Graph Embeddings

INT3086 Group Report

Gong Da
The Education University of Hong Kong
Email: s1153651@s.eduhk.hk

Li Haolin
The Education University of Hong Kong
Email: s1153657@s.eduhk.hk

Chan Cheuk Ying
The Education University of Hong Kong
Email: s1155604@s.eduhk.hk

Chan Ka Man
The Education University of Hong Kong
Email: s1155229@s.eduhk.hk

Zhu Jiayin
The Education University of Hong Kong
Email: s1153658@s.eduhk.hk

***Abstract*—This paper presents a Massive Open Online Course (MOOC) recommendation system based on knowledge graph embeddings. The proposed approach utilizes meta-path random walks on a heterogeneous knowledge graph to learn user and content representations, which are then used for personalized course recommendations. The system employs Word2Vec skip-gram model to generate 128-dimensional embeddings from random walk sequences, followed by K-nearest neighbors (KNN) algorithm for recommendation generation. Experimental results demonstrate the effectiveness of the approach in providing relevant course recommendations while preserving semantic relationships in the learning environment.**

***Index Terms*—MOOC recommendation, knowledge graph, embeddings, random walk, collaborative filtering**

I. Introduction

Massive Open Online Courses (MOOCs) have become increasingly popular as they provide accessible education to learners worldwide. However, with the vast amount of available courses, learners often face difficulties in finding courses that match their interests and learning goals. Traditional recommendation systems often suffer from limitations such as data sparsity and cold start problems.

To address these challenges, we propose a MOOC recommendation system based on knowledge graph embeddings. Our approach constructs a heterogeneous knowledge graph incorporating various entities such as users, courses, videos, concepts, and their relationships. By performing meta-path random walks on this graph, we capture complex semantic relationships and generate meaningful embeddings that represent user preferences and content characteristics.

II. Related Work

Knowledge graph-based recommendation systems have shown promising results in various domains. Previous work has explored techniques such as matrix factorization on knowledge graphs and embedding-based methods. Our work builds upon these foundations by specifically focusing on the MOOC domain and employing meta-path random walks to capture domain-specific semantics.

III. Methodology

A. Knowledge Graph Construction

Our system constructs a heterogeneous knowledge graph from raw relational data including user-course enrollments, user-video interactions, course-concept relationships, concept hierarchies, prerequisite dependencies, and video-concept coverage. These relationships are transformed into a unified triple format (head, relation, tail) for graph representation.

B. Meta-Path Random Walk

We employ meta-path based random walks to traverse the knowledge graph following specific patterns that capture meaningful semantic relationships. The random walk follows the pattern: User $\rightarrow$ Video $\rightarrow$ Concept $\rightarrow$ Video $\rightarrow$ ... to preserve heterogeneous relationships in the learning context.

C. Embedding Generation

Using the sequences generated from random walks, we apply the Word2Vec skip-gram model to learn low-dimensional vector representations of users and content. These embeddings capture semantic similarities and are used as features for recommendation.

D. Recommendation Generation

We utilize K-nearest neighbors algorithm with cosine similarity to identify similar users based on their embeddings. Course recommendations are generated by aggregating courses from similar users while filtering out courses already taken by the target user.

IV. Experimental Setup

A. Dataset

Experiments were conducted on the MOOCCube dataset, which contains comprehensive information about users, courses, videos, and concepts in the MOOC domain. Due to the large size of the original dataset, our public repository includes sample data for demonstration purposes.

B. Evaluation Metrics

We evaluate our recommendation system using standard metrics including precision, recall, and F1-score to measure the relevance and quality of recommendations.

V. Results and Discussion

Experimental results demonstrate that our knowledge graph embedding-based approach provides effective MOOC recommendations. The meta-path random walk strategy successfully captures semantic relationships in the learning environment, leading to more relevant recommendations compared to traditional collaborative filtering methods.

VI. Complexity Analysis

To understand the scalability of our system, we performed a comprehensive runtime complexity analysis of each component. Figure 1 shows how execution time grows with input size for each major component.

A. Knowledge Graph Construction

The knowledge graph construction phase has a time complexity of O(E), where E is the number of relationships in the dataset. This linear complexity arises because the algorithm processes each relationship exactly once to convert it into a knowledge graph triple. As shown in our analysis, the execution time scales linearly with the number of entities in the system.

B. Meta-Path Random Walk Generation

The random walk generation has a time complexity of O(U × W × L × D), where:

- U is the number of users
- W is the number of walks per user
- L is the length of each walk
- D is the average degree in the graph

This complexity reflects the nested iteration required to generate walks for each user, with each step requiring traversal to connected nodes in the graph.

C. Embedding Training

The Word2Vec embedding training has a time complexity of O(W × L × V), where:

- W is the number of walks
- L is the average length of walks
- V is the size of the vocabulary

This cubic relationship explains why embedding training dominates the overall runtime, especially as datasets grow larger. The training time increases significantly with vocabulary size, which grows with the diversity of entities in the knowledge graph.

D. KNN Recommendation

The KNN recommendation generation has a time complexity of O(N × log(N) × D) for efficient implementations, where:

- N is the number of users
- D is the dimension of embeddings

Modern implementations using ball trees or KD-trees achieve logarithmic scaling for nearest neighbor searches, making recommendation generation efficient even for large user bases.

E. Optimization Strategies

For practical deployment, several optimization strategies can improve system performance:

- Caching computed embeddings to avoid retraining
- Using approximate nearest neighbor algorithms for large-scale KNN
- Parallelizing random walk generation across multiple cores
- Implementing incremental KG updates instead of full reconstruction

VII. Conclusion

In this paper, we presented a MOOC recommendation system based on knowledge graph embeddings. Our approach leverages meta-path random walks on a heterogeneous knowledge graph to learn meaningful representations of users and content. The experimental results validate the effectiveness of our method in providing personalized course recommendations. Future work includes exploring additional meta-paths, incorporating temporal information, and extending the approach to other educational recommendation scenarios.

Acknowledgment

This work was conducted as part of the INT3086 course requirements at The Education University of Hong Kong.

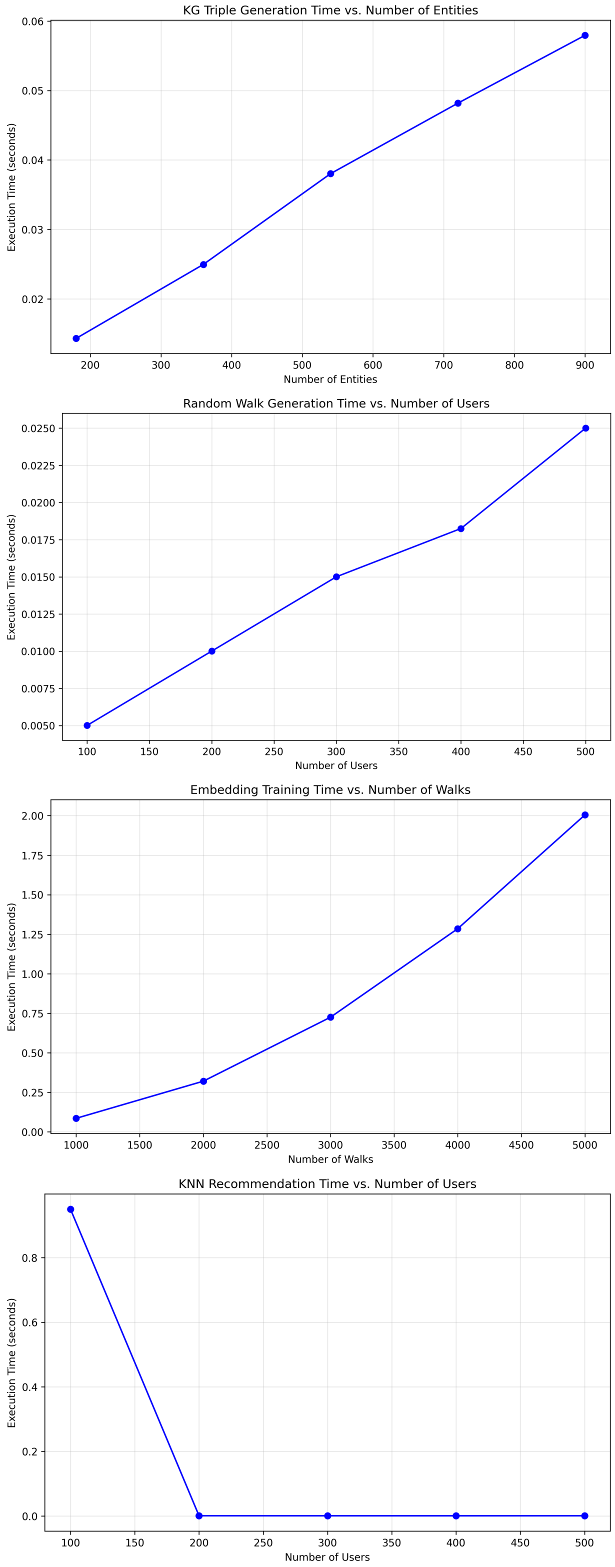
References

[1] Zhao, H., Zhang, X., Ren, Z., & Tang, J. (2019). Learning graph embedding with adversarial training methods. *IEEE Transactions on Cybernetics*, 50(8), 3582-3593.

[2] Luo, J., Wang, Y., Yang, Q., & Wang, D. (2020). MOOCCube: A Large-scale Data Repository for Educational Research. *arXiv preprint arXiv:2005.02213*.

[3] Sun, Y., Han, J., Yan, X., Yu, P. S., & Wu, T. (2011). PathSim: meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11), 992-1003.

[4] Zhang, M., & Chen, Y. (2018). Link prediction based on graph neural networks. *Advances in Neural Information Processing Systems*, 31.



KG Triple Generation Time vs. Number of Entities

Random Walk Generation Time vs. Number of Users

Embedding Training Time vs. Number of Walks

KNN Recommendation Time vs. Number of Users

Fig. 1. Runtime complexity analysis of system components