Dual-view Enhanced Knowledge Contrastive Learning for Recommendation

Anonymous Authors

1 Model Time Complexity Analysis

The time cost of DKCL mainly comes from two parts. 1) For the relation-aware attentive aggregator, the time cost of representation learning on relation- and semantic-aware knowledge views are $O(|\mathcal{G}_k^{rv}| \times L \times d)$ and $O(|\mathcal{G}_k^{sv}| \times L \times d)$, respectively. This module takes additional $O(m \times n \times K \times d)$ time for *K-means* algorithm to generate the semantic entities, where m and n separately denote the iteration times and the number of long-tail entities. 2) For the user and item propagation layer, the graph-based message passing mechanism has computational complexity $O(|\mathcal{G}_i| \times L \times d)$. The time complexity to calculate contrastive learning loss is $O(B_u \times |\mathcal{I}| \times d)$, where B_u is the number of unique items within a batch. In conclusion, our DKCL could achieve comparable time complexity with the KG-based recommendations [2,3].

2 Case Study (RQ4)

In this section, we illustrate the interpretability of our DKCL in the generation of relation- and semantic-aware knowledge views through case studies.

$$s_r = \frac{1}{N_r} \sum_{(h,r,t) \in \mathcal{G}_k} LeakyReLU\left(\mathbf{e}_r^{\top}\left(\mathbf{e}_h + \mathbf{e}_t\right)\right), \tag{1}$$

For the relation-aware knowledge view, we analyze whether the relation score learned from Eq. (1) is reasonable for the view's generation. We save the relation scores learned from each epoch during training and calculate their average as the final relation scores. As shown in Table 1, we list the top-3 relations with the highest and lowest relation scores on the MovieLens dataset. In the movie recommendation scenario, we intuitively consider that the three highest-scoring relations play important roles in movie recommendation, while the three lowest-scoring relations are relatively irrelevant to the recommendation task. For example, for a user, he/she is more concerned with the genre of the movie rather than the company that produces it. Thus, we argue that the generation of relation-aware knowledge view successfully suppresses task-irrelevant knowledge noise and possesses interpretability.

For the semantic-aware knowledge view, we speculate that there may be hidden semantic relevance among long-tail entities and generate them into semantic entities employing a clustering algorithm. We visualize the embeddings

Table 1: Top-3 relations with the highest and lowest learned relation scores on the MovieLens dataset.

Relation ID	**Relation**	**Relation Score**
1	genre	5.5443
0	rating	5.4146
20	actor	5.2355
16	film_casting_director	1.5478
7	production_companies	1.3696
18	executive_produced_by	1.3265

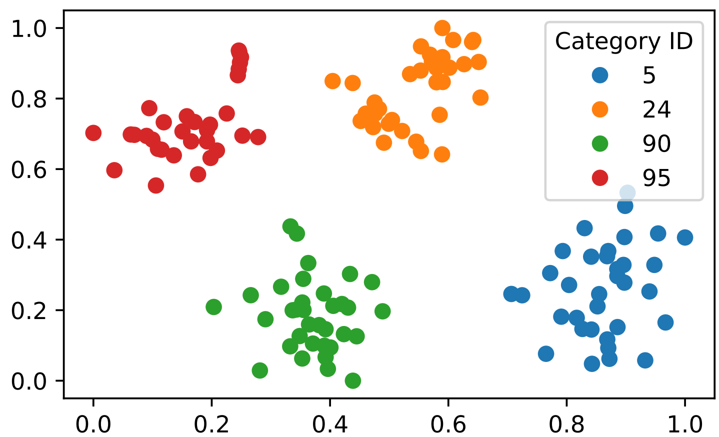

Fig. 1: Visualization of partial long-tail entity embeddings on the MovieLens dataset.

of the long-tail entities to verify their clustering effectiveness. On the MovieLens dataset, we select four semantic entities and pick the embedding of the corresponding original long-tail entities. As shown in Fig. 1, we utilize t-SNE [1] to visualize these embeddings on a two-dimensional space, and every category represents a semantic entity. We observe that the clustering algorithm achieves remarkable success and the long-tail entities are well clustered into semantic entities. This result indicates that replacing long-tail entities with semantic entities is reasonable and feasible, and the semantic-aware knowledge view successfully mitigates the knowledge sparsity issue.

References

1. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)

2. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.S.: Kgat: Knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 950–958 (2019)
3. Yang, Y., Huang, C., Xia, L., Li, C.: Knowledge graph contrastive learning for recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1434–1443 (2022)