mathematics

MDPI

Article

Inferring Complementary and Substitutable Products Based on Knowledge Graph Reasoning

Yan Fang [1], Jiayin Yu [1], Yumei Ding [1] and Xiaohua Lin [2,*]

[1] School of Maritime Economics and Management, Dalian Maritime University, Dalian 116026, China; yfang@dlmu.edu.cn (Y.F.); jiayinyu@dlmu.edu.cn (J.Y.); ymeiding@163.com (Y.D.)
[2] School of Economics and Management, Xiamen University of Technology, Xiamen 364021, China
* Correspondence: linxiaohua@xmut.edu.cn

Abstract: Complementarity and substitutability between products are essential concepts in retail and marketing. To achieve this, existing approaches take advantage of knowledge graphs to learn more evidence for inference. However, they often omit the knowledge that lies in the unstructured data. In this research, we concentrate on inferring complementary and substitutable products in e-commerce from mass structured and unstructured data. An improved knowledge-graph-based reasoning model has been proposed which cannot only derive related products but also provide interpretable paths to explain the relationship. The methodology employed in our study unfolds through several stages. First, a knowledge graph refining entities and relationships from data was constructed. Second, we developed a two-stage knowledge representation learning method to better represent the structured and unstructured knowledge based on TransE and SBERT. Then, the relationship inferring problem was converted into a path reasoning problem under the Markov decision process environment by learning a dynamic policy network. We also applied a soft pruning strategy and a modified reward function to improve the effectiveness of the policy network training. We demonstrate the effectiveness of the proposed method on standard Amazon datasets, and it gives about 5–15% relative improvement over the state-of-the-art models in terms of NDCG@10, Recall@10, Precision @10, and HR@10.

Keywords: product relationship reference; knowledge graph; knowledge representation learning; Markov decision progress

MSC: 68T30

Citation: Fang, Y.; Yu, J.; Ding, Y.; Lin, X. Inferring Complementary and Substitutable Products Based on Knowledge Graph Reasoning. *Mathematics* **2023**, *11*, 4709. https://doi.org/10.3390/math11224709

Academic Editor: Basilis Boutsinas

Received: 7 October 2023
Revised: 13 November 2023
Accepted: 17 November 2023
Published: 20 November 2023

1. Introduction

Understanding the complementary and substitutable relationships between products can determine numerous core applications in decision-making processes for retailers. [1]. For example, retailers determine which product to offer in each category [2]. Brick-and-mortar retailers seek to identify the best way to arrange the product layout in aisles and stocking shelves [3], and e-tailers also strive to recommend a list of products or promote more attractive product bundles for consumers [4]. These decisions have a significant influence on customers' choices, sales of products, and finally, profits [5].

Previous studies have formulated the relationship inference problem as an unsupervised or supervised learning problem. Unsupervised methods primarily involve models based on consumer behavioral data and implementing multi-task learning mechanisms for training, such as BB2Vec [3]. On the other hand, supervised learning methods can utilize deep learning and other techniques to predict complex relationships, as exemplified by the word embedding-based RSC framework [6], the graph neural network-based DecGCN model [4], and the graph embedding Cleora algorithm [7]. Although these methods can identify related products, they often lack interpretability and do not explain the reasons for their interconnections.

Recently, more and more attention has been given to model interpretability since it provides more evidence and reliability for the results. A knowledge graph is a semantic network, which can establish comprehensive semantic relationships and make inference results interpretable. However, when inferring product relationships, the following shortcomings still exist: (1) When the quantity of commodities increases, the inference of relationships between them becomes increasingly intricate. (2) The rapid development of e-commerce platforms have brought about diverse sets of structured and unstructured data, including commodity descriptions, consumer purchases, online reviews, as well as co-purchase and co-browsing data. Gaining information from these heterogeneous data amplifies the complexity of relationship analysis. (3) Revealing the original relationships is even more important for the analysis of product relationships.

To address these challenges, we present an improved knowledge-graph-based reasoning model for product relationship inferring. First, a knowledge graph was constructed using multi-source data. Then, we propose a knowledge representation learning model that integrates structured and unstructured knowledge. Finally, the task of inferring relationships is transformed into a Markov Decision Process (MDP) environment, where an agent searches for paths consisting of products potentially related to a given product.

The main contributions of our research can be summarized as follows:

(1) A knowledge representation learning method that integrates structured triplet knowledge and unstructured entity description knowledge has been proposed through a two-stage training process using TransE and SBERT. It provides an efficient way for multi-structured knowledge fusion and can extract more comprehensive evidence for better relationship inferring.

(2) A dynamic policy network is trained for path searching using a proper pruning strategy and a modified reward function that incorporates the cosine similarity between the inference path and query relationship. It guarantees a more precise and valid reasoning direction for the agent to find the accurate relationship paths.

(3) An extensive experimental evaluation of real-world datasets shows that our model can successfully address substitutable and complementary relationship inference with interpretable path reasoning, surpassing state-of-the-art approaches.

The rest of our paper is organized as follows. Section 2 presents a literature review. In Section 3, we introduce methods for analyzing complementary and substitutable relationships based on the knowledge graph. Experimental results and analysis are conducted in Section 4, and Section 5 concludes the research.

2. Literature Review

In the following section, we will present an overview of three streams in the literature that are relevant to our study: relationship inference, knowledge reasoning based on knowledge graphs and e-commerce recommendation systems.

2.1. Product Relationship Inference

Previous studies formulate the task of relationship inference as a link prediction problem on a graph, where products can be regarded as nodes and links represent relationships. Then the relationship inference involves estimating whether there is a potential link between two nodes. There are numerous approaches to tackle the problem such as the similarity-based approach [8], probabilistic approach [9], and, within these, machine learning techniques are the most advanced for solving the problem [10]. It works on various learning techniques which help in gaining superior performance by using the efficient dataset. Generally, two groupings of methods can be made, namely unsupervised and supervised learning.

A. Unsupervised learning

Unsupervised learning means that no training data is provided, and the method has to learn by itself by finding patterns among the input data. In most unsupervised methods, commodity relationship models are built based on consumer behavior data, followed by

the construction of multi-task learning mechanisms and ranking training. The first method applied to basket analysis to explore product relationships is the association rules [11,12]. However, association rules are sensitive to the choice of parameters and often result in a large number of unrelated connections. On the other hand, association rules are not suitable for large datasets with complex relationships. A new bipartite network has been proposed to automatically identify complements and substitutes from sales transaction data. They quantify the degree of complementarity and substitutability using weighted cosine similarity between vectors to find groups of similar products for different types of relationships [2].

With recent developments in feature representation supported by machine learning techniques, the capabilities of statistical models based on unsupervised learning are expected to be improved. For example, the Node2Vec [13] model can learn continuous feature representations for network nodes. The Struct2Vec [14] model uses a weighted multi-layer network to build context while evaluating the node pair's structural similarity based on the node's degree order. The GraphGAN (Graph Representation Learning with Generative Adversarial Nets) [15] model intends each node in a graph to be embedded by GraphGAN into a small dimensional vector space. And the BB2Vec [3] (Baskets and Browsing to Vector), a combination of several prod2vec [16] models, learns vector representations of products by jointly analyzing baskets and browsing data to make complementary product recommendations.

Generally, the unsupervised learning method utilizes unlabeled data, which can be used for more flexible and challenging problems. These methods require complicated computations. The results of the unsupervised learning method may be less precise because the input data is not labeled, and the algorithms do not know the exact result beforehand.

B. Supervised learning

Supervised learning methods, leveraging techniques including deep learning, can effectively model highly complex relationships and become one of the preferred solutions for inferring relationships in recent years. One of the most commonly used supervised learning-based link prediction methods is the support vector machine (SVM) algorithm [17], which uses a set of pre-classified data to create a classification model that classifies links into positive and negative links, thus inferring the relationships.

With the widespread adoption of textual information processing technology, the LDA [18] model can infer relationship by constructing product representations from textual information. However, it neglects relationship information. The PMSC model adopts relation constraints to distinguish relationships [19], and LVAE links two variational auto-encoders to learn latent features from product reviews [20]. SPEM considers both textual information and relational constraints [21].

With the development of embedding algorithms, Zhao et al. [1] proposed an RSC framework, which can simultaneously address user ratings, substitutable networks and complementary networks. Recently, researchers have explored graph neural networks, such as the decoupled graph convolutional network (DecGCN) model [6] and the graph embedding Cleora algorithm [4], to learn product representations in different relationship spaces. Although these methods are highly applicable, they can only identify associated commodities without providing explanations for their mutual relationships.

2.2. Knowledge Graph Reasoning

In recent years, with the rapid development of big data, which contains a significant amount of valuable knowledge, the knowledge graph has emerged as an intuitive, concise, and flexible form of knowledge expression since it is a special heterogeneous network containing rich semantic and structural information. Knowledge reasoning aims to extract unknown and implicit knowledge from known information. Knowledge reasoning based on knowledge graphs is more diversified and has been explored in many research areas such as recommendation [22], question answering [23], and medicine treatment [24].

Generally speaking, knowledge graph reasoning can be classified into four main categories: logic-rule-based reasoning, representation-learning-based reasoning, neural-network-based reasoning, reinforcement-based reasoning, and hybrids.

A. Logic-Rule-Based Reasoning

Logic-rule-based reasoning methods refer to reasoning new relations based on rules of first-order predicate logic or description logic. For example, in the inductive logic programming model, entity relations in the knowledge graph are regarded as facts described by binary predicates, and the first-order logic rules are learned from the knowledge graph [25]. Wang et al. [26] proposed a first-order probabilistic language model for personalized web page ratings on a knowledge graph. Jang et al. [27] proposed a rule-based reasoning model for triplet quality assessment based on the hypothesis that more frequent patterns are more reliable. CAFE [28] uses path reasoning to deduce fine-grained models in neural symbolic reasoning methods from user behavior sketches.

The rule-based reasoning method is similar to human reasoning and possesses good deductive ability. However, the efficiency of logical reasoning on large-scale knowledge graphs is limited.

B. Representation-Learning-Based Reasoning

The representation-learning-based reasoning method is to define a mapping function that maps a symbolic representation to a vector space for knowledge representation. It starts with the feature representation of entities and relationships in the knowledge graph, and the data structure is then converted into vectors for knowledge reasoning [29]. One of the representative methods is the TransE (translating embedding) algorithm [30], which constructs a canonical model to interpret relationships as translation operations on the low-dimensional embeddings of the entities. However, it cannot handle the one-to-many relationships properly, which decreases its robustness and flexibility. Many scholars have made improvements. For example, the TransAt model [31] adopts the attention mechanism to enhance focus on relation-specific attributes. The TransA model [32] uses Mahalanobis distance to increase flexibility. The CTransR [33] clusters different entities of the same relation to learn a separate relation vector representation for each entity.

Generally, the representation-learning-based method is direct and quick, reducing the dimensionality of disaster and capturing the implicit relationships between entities and relations. However, it lacks explicit reasoning processes, limiting their persuasiveness in their results.

C. Neural-Network-Based Reasoning

Neural network-based reasoning methods exhibit strong expressive capabilities in relation to link, prediction and other tasks. They leverage feature learning and nonlinear transformations to convert the original feature representations into another feature space Generally speaking, knowledge reasoning methods based on neural networks mainly include convolutional neural networks (CNN), recurrent neural networks (RNN), graph neural networks (GNN), and graph convolutional network (GCN).

CNN-based reasoning is capable of using convolution operations to extract local features from a single triple for predicting missing relationships. It is one of the earliest neural network approaches applied for knowledge reasoning. For example, Zhao et al. proposed a novel representation learning method that can extract feature information from text data by using CNN [34]. The MCKRL model [35] employs graph attention mechanisms and convolutional neural networks to construct an encoder-decoder model based on TransE, enabling knowledge representation learning from multi-source information. CNN can enhance the representation of the entity relationship structure vector in the existing knowledge graph and obtain rich semantic information.

Many studies focus on using RNN to analyze the knowledge paths, which can effectively analyze entity text descriptions. As a typical method, the learning knowledge graph embedding with entity descriptions (KGDL) utilizes long short-term memory networks to encode related text descriptions, and then uses triplets to encode entity descriptions, realizing the prediction of missing knowledge [36]. An RNN is designed to repeatedly

interact with an attention module to derive logical inferences from the representations of multiple paths to capture the semantic correlations [37].

GNN has been widely explored for knowledge reasoning since it enlarges the learning scope from a single triplet in CNN or knowledge path in RNN to knowledge subgraphs in GNN [38]. For example, Wang et al. introduced attention to utilize the context information by using GNN (Graph Neural Networks) [39]. HRRL [40] performs path-based inference using GNN to encode neighborhood information. Some scholars have tried to apply graph auto-encoders (GAE) and spatiotemporal graph neural networks (STGNN) to knowledge reasoning tasks. The GAE uses a multi-layer perceptron as an encoder to obtain a low-dimensional representation of a node for knowledge reasoning [41]. It is an unsupervised learning framework that is widely applicable, but its overall performance needs to be improved. The STGNN can consider both spatial and temporal dependence and can handle highly nonlinear and complex problems [42]. Meanwhile, the complexity of the model is high, and its application in time series needs improvements.

The recent graph convolutional network (GCN) provides another way of learning graph node embedding by successfully utilizing the graph connectivity structure. The basic idea of GCN is to reduce the higher-dimensional information of the nodes in the graph data to a lower-dimensional vector representation. It is a form of "transductive learning", which means that all information of the graph data is required for training, and the model cannot quickly obtain the representations of new nodes. To overcome this limitation, researchers have made modifications. For example, a novel end-to-end Structure Aware Convolutional Network (SACN) [43] combines the benefit of GCN and ConvE. It has learnable weights that adapt the amount of information from neighbors used in local aggregation, leading to more accurate graph nodes embeddings.

D. Reinforcement-learning-based reasoning

Reinforcement learning methods mainly involve reasoning based on relation path exploration and utilize agents' traversal for state transitions based on reward policy [4]. Xiong et al. first introduced reinforcement learning methods [44] and proposed the DeepPath model. Subsequently, Xian et al. [45] presented a policy-guided path reasoning (PGPR) method, which provides real paths in the knowledge graph. Based on this, Yang et al. [5] proposed a knowledge-aware path reasoning (KAPR) method, establishing a multi-feature inference component (MFI) as the reward function for reinforcement learning in product relationship inference. Generally, reinforcement learning-based knowledge reasoning methods offer interpretability but entail higher training costs when the action space is large.

2.3. E-Commerce Recommendation

The last stream of related literature is the recommendation methods for e-commerce. we will present an overview of the most relevant method to this research, namely recommendation based on deep learning. Leveraging deep learning techniques within the context of rating matrices has the potential to significantly enhance recommendation performance. Deep learning models can extract intricate features from data automatically. When applied to personalized product bundle recommender systems, they can utilize the abundance of features within the system and uncover complex relationships among these features.

Some researchers have attempted to enhance the recommendation effectiveness from the perspective of encoders. The AutoRec model [46] relies on autoencoders to reconstruct co-occurrence matrices, thereby obtaining customized encodings of users and items for rating prediction. Bai et al. have [47] introduced a novel bundle generation network (BGN) that alleviates the limitations of traditional softmax representations.

Additionally, some scholars have incorporated Convolutional Neural Networks (CNNs) into recommender systems, leveraging features such as product images and descriptive text. For instance, Kim et al. [48] proposed a context-aware CNN recommendation model, enhancing the accuracy of rating predictions. In addressing the issue of missing item data in the rating matrix, Defferard et al. [49], Bronstein et al. [50], and Hamilton et al. [51] have made notable progress in the application of Graph Convolutional Networks (GCNs).

Building upon the BPR model introduced by Steffen Rendle and colleagues [52], Apurva Pathak and others [53] have ventured into creating personalized new bundles for users through a Product Bundle Personalization Rank (BPR) model.

Attention-based recommendation models have witnessed significant advancements recently. Han et al. [54] introduced an Adaptive Deep Latent Factor Model (ADLFM) that takes into account individual diversity. Chen et al. [55] proposed an attention-driven factor model that tailors recommendations based on a user's varying levels of attention towards different aspects of products. This approach can generate attention distributions specific to individual users, resulting in notably high prediction accuracy. Furthermore, Hada et al. [56] suggested the use of the Plug-and-Play Language Model to generate individual recommendations through a jointly trained cross-attention network.

2.4. Summary

In summary, existing studies on how to derive relationships in an interpretable way have become increasingly significant. Some researchers have explored the application of knowledge graph techniques in relationship inference problems to enhance the interpretability of models. To enhance the efficiency of knowledge-graph-based model, we conduct in-depth research on the following issues.

(1) Data heterogeneity. There are great differences in data structures, including unstructured textual data such as product description, and structured data such as product attributes and behavioral data generated by users. The question concerning how to integrate these data and extract adequate evidence is the main focus of this research.

(2) Effectiveness of reasoning. The question concerning how to modify the reasoning process based on valuable knowledge and integrate domain knowledge for more effective relationship inference will be further investigated in this paper.

3. Methodology

In this section, we first formalize our relationship inference problem. Then we present our approach based on reinforcement learning over knowledge graphs to solve the problem.

3.1. Problem Formulations and Principles of the Methods

The knowledge graph is defined as $G = \{e,\ r,\ e' | e, e' \in E,\ r \in R\}$, where E is the entity set and R is the relationship set. The tuple $(e,\ r,\ e')$ reflects the fact that head entity e and tail entity e' have the relationship r.

According to [45], we give a relaxed definition of k-hop paths over graph G and the formalized problem as follows.

Definition 1 (k-hop path). *A k-hop path from entity* $e_0$to entitye_k *is defined as a sequence of k+1 entities connected by k relations, denoted by* $p_k(e_0, e_k) = \left(e_0 \xrightarrow{r_1} e_1 \cdots e_{k-1} \xrightarrow{r_k} e_k\right)$, *where* $e_{k-1} \xrightarrow{r_k} e_k$ *means* $(e_{i-1},\ r_i, e_i) \in G$, $i \in [k]$.

The problem of inferring complementary and substitutable products based on a knowledge graph (IPKG) can be formalized as below.

Definition 2 (IPKG Problem). *Given a knowledge graph G item,* $v_0 \in V$ *and integers K and N, the goal is to find a set of products* $\{v_n\}\ n \in [N] \subseteq V$ *such that each pair* (v_0, v_n) *is associated with one explanatory path* $p_k\ (v_0, v_n)$ $(2 \leq k \leq K)$, *and N is the number of complementary and substitute products.*

As an example illustrated in Figure 1, given a knowledge graph with entities and relationships, the algorithm is expected to find potential complementary and substitutable products, e.g., (Earphone, Laptop) and (Keyboard A, Keyboard B), along with their reasoning paths in the graph, e.g., {Keyboard A → Category A → Keyboard B} and {Laptop → Mouse → Brand A → Earphone}.

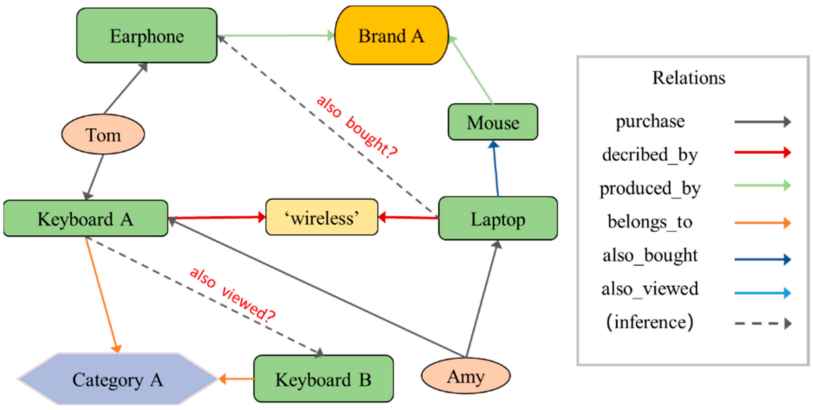


Figure 1. Illustration of the relationship inferring based on knowledge graph.

Generally, the proposed knowledge-graph-based relationship reference method (KGRR) can be divided into three stages as illustrated in Figure 2. Firstly, we construct a knowledge graph based on a triplet structure. Next, a two-stage knowledge representation learning model is introduced that combines TransE and SBERT to learn knowledge representation from the structured and unstructured data. Finally, a Markov decision process-based approach is used to convert the task of relationship inferring into a path inference problem on the knowledge graph.

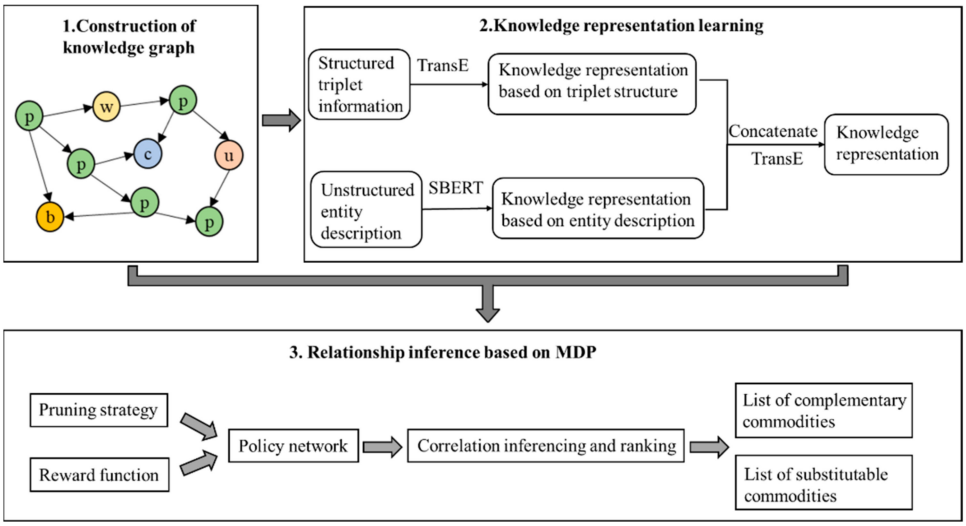


Figure 2. Principles of product relationship inferring methods based on the knowledge graph.

3.2. Construction of the Knowledge Graph

The primary purpose of constructing a knowledge graph is to organize the information of products, users as well as relationships among them. Hence, the basic elements of the knowledge graph are entities and relationships. The entity is defined as the independent objectives such as products, brands, product categories, users, and the meta-word for product descriptions and reviews as shown in Table 1. The relationship represents the interaction or connection between entities as shown in Table 2. For example, "*also_viewed*" and "*also_bought*" keep information about products that have been viewed or purchased together. The relationship of "*described_by*" indicates which words are included in the product's description.

Table 1. List of entities.

Entity	Defined Variables	Descriptions
Product	v	Product in the datasets
User	u	Users in the datasets
Word	w	Words in comments
Brand	b	Brand of products
Category	c	Category of products

Table 2. List of relationships.

Relationship	Description	Head Entity	Tail Entity	Defined Triplet
Also_viewed	Products that also be viewed together	Product	Product	(*v1*, *also_viewed*, *v2*)
Also_bought	Products that also be bought together	Product	Product	(*v1*, *also_bought*, *v2*)
Purchase	Products purchased by users	User	Product	(*v*, *purchase*, *u*)
Described_by	words extracted from the reviews that describe the product	Product	Word	(*v*, *described_by*, *w*)
Produced_by	Brand that the product is made of	Product	Brand	(*v*, *produced_by*, *b*)
Belong_to	Category that the product belongs to	Product	Category	(*v*, *belong*, *c*)

In this research, the entity set E includes sets of items V, users U, words W, brands B, and categories C. The relationship set R consists of six types of relationships and is formulated as six types of triplets as shown in Table 2. Following the definitions from previous relevant research [5,28], we categorize two products with an "*also_bought*" relationship as complementary products and products with an "*also_viewed*" relationship as substitute products.

Figure 3 provides a specific example of the knowledge graph. There are three types of entities, namely word, user and product, which are marked as dots with different colors. The arrows represent the relationship between entities exacted from triplet information. For example, the triplet information, "Keyboard A, also_bought, mouse B", is represented by an arrow from Keyboard A to mouse B as shown in the knowledge graph. After extracting the entities and relationships from the dataset, the knowledge graph will be constructed.

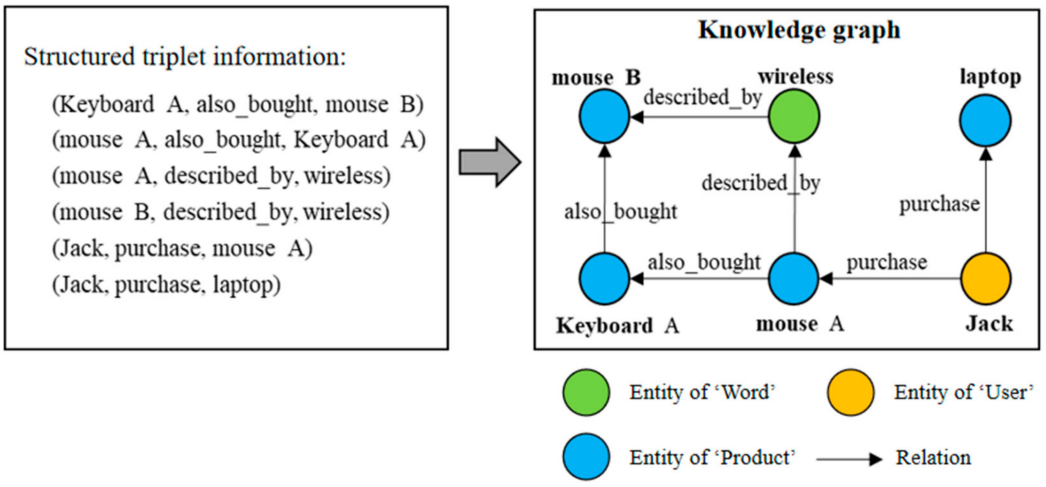


Figure 3. An example of knowledge graph construction.

3.3. Knowledge Representation Learning

3.3.1. The Framework

Knowledge representation learning on the knowledge graph is to represent entities and relations in a knowledge base as low-dimensional real-valued vectors. Most knowledge representation learning methods have focused on triplet structured information [29]. However, the corresponding unstructured entity descriptions can also provide precise and reliable supplementary information for reasoning. For example, when two products have similar descriptions, they probably have complementary relationships. Consequently, we integrate triplet structure information and entity description information for knowledge representation.

Meanwhile, the two types of knowledge sources are heterogeneous. The factual triplets are structured while the entity descriptions are unstructured textual features. In order to handle the heterogeneous knowledge representation problem, we proposed a two-stage learning method based on TransE and SBERT. As shown in Figure 4, the first stage, the learning model based on TransE for triplets and that based on SBERT for entity descriptions are conducted in parallel. In the second stage, the vectors learned by the two parts are concatenated together and trained by TransE again to generate the final knowledge representations.

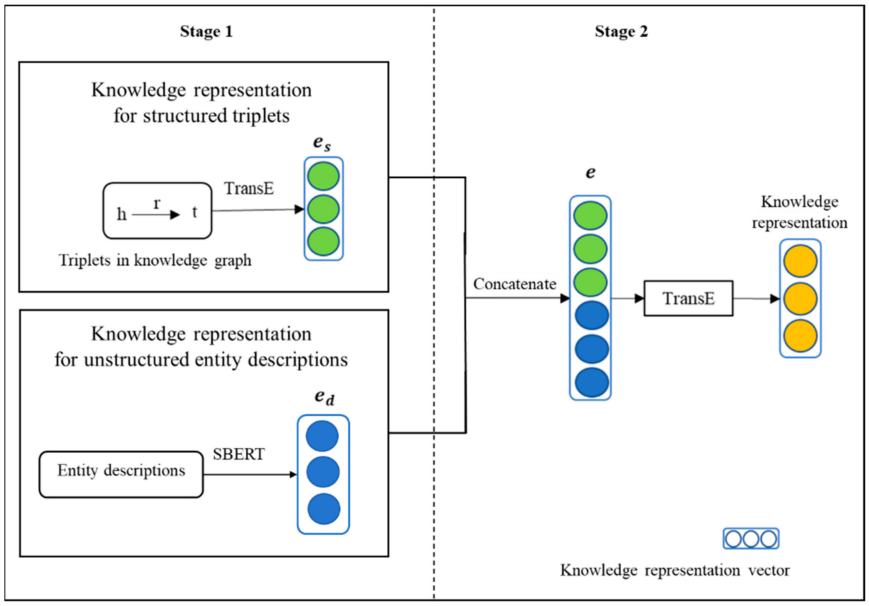


Figure 4. Overall structure of the two-stage knowledge representation learning.

TransE [30] is a way to model relationships on the knowledge graph which shows advantages for the problem with multiple relational data. Moreover, experiments show that TransE is significantly superior to most advanced methods in linking prediction on two knowledge bases. SBERT [57] uses independent submodules to extract high-dimensional information from the text feature vectors. SBERT processes large amounts of similar text very quickly. Consequently, we choose to merge the two models for the heterogeneous knowledge representation problem.

3.3.2. Definitions and Formulations

The TransE models relationships as "translations". Its principle is illustrated in Figure 5, where the relationship r is treated as a translation from the head entity h to the tail entity t, which means $h + r = t$. The word embeddings trained by the TransE model

can generalize some semantic information contained in the knowledge graph and have spatial translatability. The energy function of TransE is defined as:

$$f(h,r,t) = \| \boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t} \|_2 \tag{1}$$

where $\|\cdot\|$ represents the norm of the difference vector.

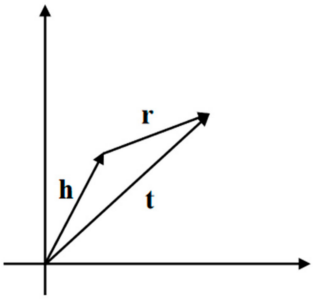


Figure 5. Schematic diagram of the TransE principle.

The knowledge representation based on entity descriptions is learned using the SBERT model. The structure of the SBERT model is illustrated in Figure 6. Initially, sentences of entity descriptions are input into a pre-trained BERT network and pooled. Siamese networks and the triplet structure are employed to obtain the representation vectors for each sentence. Subsequently, the sentence vectors s_A and s_B, along with their difference vectors, are concatenated to form a new vector. This concatenated vector is then multiplied by the weight parameter $W_t \in R^{3n*k}$, where k represents the number of classification labels and n denotes the vector dimension. Finally, the optimization is performed using the cross-entropy loss function as shown in Equation (2).

$$o = softmax(W_t(s_A, s_B, |s_A - s_B|)) \tag{2}$$

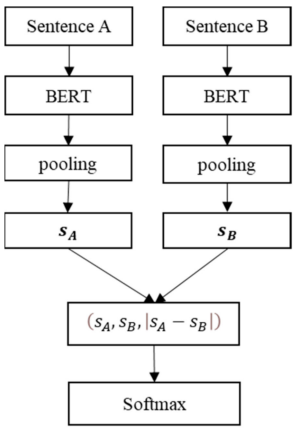


Figure 6. SBERT model structure diagram.

In the two-stage model, the proposed knowledge representation model projects two types of knowledge representations into a unified low-dimensional vector space. In this framework, the overall energy function [58] is defined as follows:

$$\boldsymbol{e} = \boldsymbol{e_s} \oplus \boldsymbol{e_d} \tag{3}$$

where, e_s and e_d represent knowledge representation based on triplet structure and entity description learned by TransE and BERT respectively, and $\oplus$ denotes the concatenation operator.

The set of existing factual triplets is represented as $S = \{(h, r, t) | h \in E, r \in R\}$. The parameters of the knowledge representation learning model are denoted as $\theta = \{E, R, X\}$, where E and R are the knowledge representation sets of entities and relationships based on triplet structure, and X represents the knowledge representation set based on entity descriptions.

The distance hypothesis proposed in TransE is extended to combinate with SBERT. The training of the knowledge representation model, which integrates entity descriptions and triplet structure information, is achieved by minimizing the interval-based loss function. The formula is shown in Equation (4).

$$L = \sum_{(h,r,t) \in S} \sum_{(h',r',t') \in S'} max\left(\gamma + d(h + r, t) - d\left(h' + r', t'\right), 0\right) \tag{4}$$

where γ is a hyperparameter used to measure the boundary between correct and incorrect triplets. S represents the set of positive example triplets composed of factual triplets $(h,\ r,\ t)$, and S' represents the set of negative example triplets formed by erroneous triplets $(h',\ r\prime,\ t\prime)$. Erroneous triplets are generated by randomly replacing the head entity, tail entity, or relationship in the set of positive example triplets, as shown in Equation (5).

$$\begin{aligned} S' = & \{(h', r,\ t) | h' \in E, (h', r,\ t) notin S\} \cup \\ & \{(h,\ r, t') | t' \in E, (h,\ r, t') notin S\} \cup \\ & \{(h,\ r',\ t) | r' \in R, (h,\ r',\ t) notin S\} \end{aligned} \tag{5}$$

3.3.3. The Algorithm

The improved knowledge representation learning model consists of four steps as follows.

Step 1: The knowledge representation E and R are derived from the triplet structure information of entities and relations.

Step 2: Entity descriptions are used as input to the SBERT model to obtain the knowledge representation X for entity descriptions.

Step 3: E, R and X are combined using Equation (3) to form the initial knowledge representations in the retraining model.

Step 4: By using stochastic gradient descent to update the model parameters and minimize the value of the loss function, the model is optimized.

The training process of the knowledge representation learning model that incorporates entity descriptions is outlined by the pseudocode as shown in Algorithm 1.

(1) The embeddings of entities and relationships are initialized.

(2) The triplet-based entity and relational embeddings, h_s, r, t_s, are then generated through the TransE model.

(3) Entity embeddings based on entity descriptions are generated by the SBERT model.

(4) After both embeddings are combined and put into the TransE model for training, the final entity and relationship, h, r, t, are output, indicating the ultimate knowledge representation of entities and relationships.

Algorithm 1 Training process of the knowledge representation learning model with entity description fusion

Input: triple fact set, entity description set. Entity set E, relation set R, and learning rate lr.
initialize
for $e \in E$ do $e \leftarrow$uniform$(-\frac{6.0}{\sqrt{k}}, \frac{6.0}{\sqrt{k}})$ .k is a random number
  $r \in R$ do $r \leftarrow$uniform$(-\frac{6.0}{\sqrt{k}}, \frac{6.0}{\sqrt{k}})$
end for
loop
$S_{\text{batch}} \leftarrow$sample (S, b) .sample a minibatch of size b
$T_{\text{batch}} \leftarrow \emptyset$ .initialize the set of pairs of triple facts
  for $(h, r, t) \in S_{\text{batch}}$ **do**
    $(h', r, t') \leftarrow$sample$(S'_{(h,r,t)}$ .sample a corrupted triplet
    $T_{\text{batch}} \leftarrow T_{\text{batch}} \cup \left\{ \left((h, r, t), (h, r, t)' \right) \right\}$
  end for
  .train the structure-based representations $\boldsymbol{h_s}$, $\boldsymbol{r}$, $\boldsymbol{t_s}$
  Update representations w.r.t. $L = \sum_{(h,r,t) \in S} \sum_{(h',r',t') \in S'} max(\gamma + \boldsymbol{e_s}(h + r, t) - e_s(h' + r', t'), 0)$
  .use $\boldsymbol{e_s} = \|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|_2$
  .train the entity description-based representations $\boldsymbol{h_d}$, $\boldsymbol{t_d}$
  Update representations w.r.t. $o = softmax(W_t(s_A, s_B, |s_A - s_B|))$
end loop
output: All of the representations of $\boldsymbol{h}$,$\boldsymbol{r}$,$\boldsymbol{t}$.

3.4. A Markov Decision Process-Based Reasoning Model

3.4.1. Formulation of the Markov Decision Process

In this paper, the knowledge reasoning method based on reinforcement learning (RL) is adopted for product relationship reasoning. It is mainly composed of four components, namely the agent, the environment, the state, and the action and reward. Here, the Markov Decision Process (MDP) is used to model the environment. Formally, a MDP can be represented using a tuple (S, A, p, R), where S represents states, A represents actions, p denotes the probability of state transitions, and R represents rewards. The specific definitions of each component are as follows:

(1) State. The state, $s_t \in S$, is the agent's observation of its current environment, which is the sequence of entities and relationships traversed by the agent after t steps. It can be represented as $s_t = (v_0, r_1, e_1, \cdots, r_{t-1}, e_{t-1}, r_t, e_t)$, where e_t is the entity reached at step t, and r_t is the relationship connecting e_{t-1} and e_t. When t = 0, the product v_0 is the given initial entity.

(2) Action. An action $a_t \in A_t$ is a binary tuple, (R, E), containing a relationship and its corresponding entity. For a given state s_t, the policy network returns the next action $a_t = (r_{t+1}, e_{t+1})$. For a given path $\{e_0, \cdots, e_t\}$, the action space A_t for the state s_t at time t is defined as all edges connecting to the entity e_t, excluding historical entities and relationships, i.e., $A_t = \{(r, e) | (e_t, r, e) \in G, enotin\{e_0, \cdots, e_t\}\}$. Since some nodes on the knowledge graph have much larger out-degrees, it is fairly inefficient to maintain such a big size of the action space. Thus, a pruning strategy that effectively keeps the promising edges based on a scoring function is needed to keep a proper size of the action space. The details of the pruning strategy as well as the scoring function will be introduced in Section 3.4.2.

(3) Transition. The probability of state transition, $P(s_t, a_t)$, is the probability that the environment enters the next state s_{t+1} after the agent selects an action a_t based on the current state s_t. It is represented by a mapping function, $S \times A \rightarrow S$, which can be described using a policy function $\pi_\theta(a|s)$. Given the state $s_t = (v_0, r_1, e_1, \cdots, r_t, e_t)$, action $a_t = (r_{t+1}, e_{t+1})$, and fact $(e_t, r_{t+1}, e_{t+1}) \in G$, the transition equation for the state at t + 1 is:

$$s_{t+1} = (v_0, r_1, e_1, \cdots, e_t, r_{t+1}, e_{t+1}) \tag{6}$$

(4) Reward. In the training process of the agent, the target entity is unknown. Therefore, a reward must be defined, which is only computed when the path searched by the agent ends with a product, otherwise, it is 0. The reward function will be described in detail in the next section.

An MDP refers to a stochastic control process where agents continuously observe discretely structured systems with Markov properties, resulting in a sequence of decisions [59]. The state transition process is illustrated in Figure 7. Here, agents observe the current state, select an action from the action set based on a policy, undergo a state transfer upon action selection, and subsequently make a new decision based on the new state observed. Through this iterative interaction with the environment, a continuous decision-making process is formed.

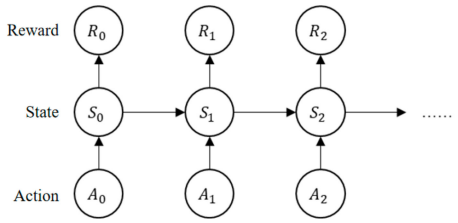


Figure 7. State transition of Markov Decision Process.

3.4.2. The Path Reasoning Based on Knowledge Graph

The problem of product relationship inference has been transformed into a path reasoning problem on the knowledge graph based on MDP. The pipeline of path reasoning method based on the knowledge graph is shown in Figure 8. This is achieved by having the agent traverse the knowledge graph to find products that potentially have complementary or substitutable relationships with the given item. The agent learns a policy network through the reward function in its interaction with the knowledge graph, utilizing the acquired policy network for relationship inference tasks. In the inference stage, the trained agent samples a series of explicit paths for each product, and all of the commodities on the path constitute the candidate commodity set. Following ranking based on their respective scores, the final rankings for complementary and substitutable products are obtained. The top n-ranked goods for each relationship type are selected as the ultimate inference result product set.

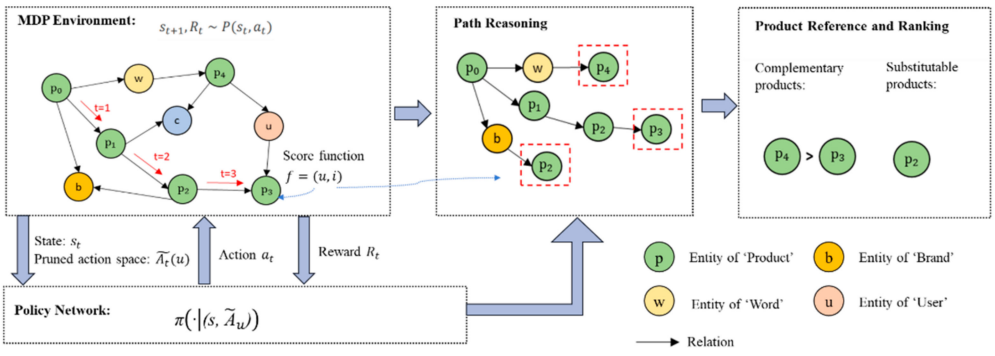


Figure 8. The pipeline of path reasoning based on the knowledge graph.

(1) The Policy Network

The policy of Markov decision process is the decision basis for the agent to choose the corresponding action in the current state. The goal of path inference is to learn a policy network π that maximizes the expected cumulative reward for path reasoning. A policy network is a neural network trained to approximate a policy function. We employ a stochastic policy, which introduces some randomness for the agent to explore more potential paths. Then, the policy function $\pi(a|s)$ is a probability density function,

used to control the agent's movement. The $\pi(a|s)$ takes a state s as input and outputs a probability distribution over each action a. When modeling a reinforcement learning environment based on Markov decision processes, the agent's goal is to learn a policy network $\pi(\cdot|(s, \tilde{A}_u))$ to approximate the policy function that maximizes the expected cumulative discounted reward of the obtained inference paths. The training process of the policy network and its structure are illustrated in Figure 9.

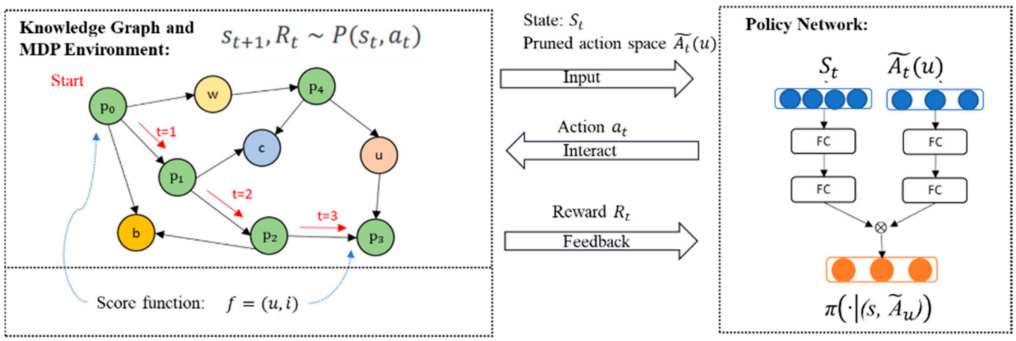


Figure 9. Illustration of the policy network structure and its training process.

The policy network $\pi(\cdot|(s, \tilde{A}_u))$ takes state s and pruned action space embedding $\tilde{A}_u$ from knowledge graph environment as inputs and outputs the probability of each action. Following the model of [5], a neural network with two hidden layers is applied in the policy network. It maps s and $\tilde{A}_u$ to a learnable feature space, calculates the correlation between the state s and each action, and then applies the SoftMax function to normalize the correlations into a probability distribution. The structures of the dynamic policy network are defined as Equation (7):

$$\begin{aligned} \boldsymbol{s}' &= \sigma(\sigma(\boldsymbol{s}W_s)\ W_1) \\ \boldsymbol{a}_s &= \tilde{A}_{v_0} W_A \\ \boldsymbol{a}'_s &= \sigma(\sigma(\boldsymbol{a}_s W_a)\ W_2) \\ \pi(\cdot|(s, \tilde{A}_u)) &= softmax(\boldsymbol{s}' \otimes \boldsymbol{a}'_s) \end{aligned} \tag{7}$$

Here, $\otimes$ is the Hadamard product, which is used to mask invalid actions. The parameters of both network models are denoted as $\theta = \{W_1, W_2, W_s, W_a\}$. Let M be the size of the space action and D be the maximum size of the space action.

The vector $s \in R^{d_s}$ represents the embedding of the state, which is obtained by concatenating of embeddings along the path, $(v_0, r_1, e_1, \cdots, r_t, e_t)$, from entity, v_0 to entity e_t with relations r. Then $s' \in R^{d_{s'}}$ represent the learned hidden features of the state. σ is a non-linear activation function, and in this work, we use ReLU (rectified linear unit) which introduce property of nonlinearity to a deep learning model and is widely used to solve the vanishing gradients issue.

$a_s \in R^{d_a \times M}$ is the representation of pruned action spaces $\tilde{A}_u$ with a size of D. W_A is an action-to-vector lookup table. Then a_s is formed by concatenating a set of pruned actions' embeddings $(a_0, \ldots, a_i, \ldots, a_M,)$. Each action a_i is represented as the embedding concatenation of (r_{t+1}, e_{t+1}) for a given stage t. Here, v_t, e_t r_t are representations of product, entity and relationships learned in Section 3.3. $\boldsymbol{a}'_s \in R^{d_{a'} \times M}$ are the embedding and hidden features of all pruned actions space of $\tilde{A}_u$.

The training process of the policy network adopted the classic policy gradient algorithm, namely the reinforcement algorithm [60]. The gradient definition of the policy network is shown in Equation (8):

$$f\, \nabla_\theta J(\theta) = E_\pi \left[\nabla_\theta log \pi_\theta (\cdot|s, \tilde{A}_u)\, G \right] \tag{8}$$

where, G represents the cumulative discounted reward from the initial state to the final state.

(2) The Pruning Strategy

In the knowledge graph, the outdegree of nodes follows a long-tail distribution. Therefore, a pruning strategy is needed to preliminarily filter out low-value actions in order to reduce the model's action space and enhance learning efficiency. The pruning strategy can prune actions in the action space A_t, and the action space after pruning is represented as $\tilde{A}_t$. The main purpose is to select and retain nodes that contribute to inferring node relationships and are closely related to v_0.

According to the KAPR [5], the pruning process includes two steps. Firstly, a series of improbable actions has been eliminated based on meta-path patterns, which specifically describe potential path patterns between products with complementary and substitutable relationships. For example, two products with the binary-hop pattern, "$product \xrightarrow{also\_viewed} product \xrightarrow{also\_viewed} product$", may have complementary or substitutable relationships. Then, based on the scoring function $f((r,e)|v_0)$, the real-valued scores of all actions $(r,\ e)$ are computed with a given initial product v_0. The top D actions are selected to form the pruned action space, i.e.,:

$$\tilde{A}_t = \{(r,e)|rank(f((r,e)|v_0)) \leq \alpha, (r,e) \in A_t\} \tag{9}$$

where α is a predefined integer, with its upper limit being the size of the action space. For a given initial product v_0, its scoring function, which can measure the correlation between products, is defined in Equation (10):

$$f((r,e)|v_0) = \begin{cases} \langle \boldsymbol{v}_0 + \boldsymbol{r}, \boldsymbol{e} \rangle + b_e, enotinV \\ max(\langle \boldsymbol{v}_0 + \boldsymbol{r}_{\boldsymbol{also\_viewed}}, \boldsymbol{e} \rangle + b_e, \\ \langle \boldsymbol{v}_0 + \boldsymbol{r}_{\boldsymbol{also\_bought}}, \boldsymbol{e} \rangle + b_e), e \in V \end{cases} \tag{10}$$

where $\langle \cdot, \cdot \rangle$ represents the dot product operation, e and r are vectors representing entities and relationships, b_e is the bias for entity e [45].

(3) The Reward Function

In the policy network training, the reward function is a key factor in guiding the agent to make effective decisions. Some research provides corresponding rewards by setting a maximum path length threshold [45,61,62]. However, it usually results in sparse feedback rewards for the agent in the interaction with the environment, which may cut potentially relevant paths for relationship learning. Therefore, we introduce a soft reward strategy providing rewards for each path, which alleviates the sparsity of rewards.

Zhao et al. [61] pointed out that in the process of knowledge inference, meta-paths correspond to meta-level interpretive strategies that can guide policy generation. Thus, a series of meta-paths are designed as rule constraints to ensure that the interpretable paths selected by the agent possess strong persuasiveness. From the semantic perspective, for each query triplet (v_0, r_q, e_t), the inference path should consist of a set of relationships from the source entity v_0 to the target entity e_t. The query relationships r_q, namely complementary and substitutable relationships, theoretically represent the direct connections between the source and target entities. Therefore, reliable inference paths and the semantics of query relationships should be similar. To increase the reliability of the inference path, the similarity between the inference path and the query relationship is proposed as reward feedback to the agent. The higher the similarity, the more reliable the path. The specific reward function is defined as follows:

$$R_{global}(t) = R_t + cos\langle \sum_{s=1}^{t} \boldsymbol{r_s}, \boldsymbol{r_q} \rangle \tag{11}$$

$$R_t = \begin{cases} max\left(0, \frac{f(v,e_t)}{max_{e \in V} f(v,e)}\right), e_t \in V \\ 0, \qquad\qquad\qquad\qquad else \end{cases} \tag{12}$$

$$f(v,e) = f(v_0,e_t) = \langle \boldsymbol{v}_0 + \sum_{s=1}^{t} \boldsymbol{r_s}, \boldsymbol{e_t} \rangle + b_{e_t} \tag{13}$$

where $cos\langle\cdot,\cdot\rangle$ is cosine similarity, $\sum_{s=1}^{t} r_s$ represents the sum of all relationships along the inference path, r_q represents query relationships, namely complementary and substitutable relationships, and the value of R_t is normalized to the range of [0,1].

Specifically, if entity R_t and entity R_t are connected by multi-hop paths, such as v_0 and e_2 on the path $\{v_0, r_1, e_1, r_2, e_2\}$, the calculation formula is:

$$f(v_0,e_2) = \langle \boldsymbol{v_0} + \boldsymbol{r_1} + \boldsymbol{r_2}, \boldsymbol{e_2} \rangle + b_{e_2} \tag{14}$$

The fundamental principle of knowledge graph representation learning assumes that $h + r \approx t$, thus the knowledge representation of relationships serves as a bridge between entities. Under the soft reward strategy, the essence of rewards lies in evaluating the relevance between entity v_0 and entity e_t. Therefore, $f(v,e)$ not only considers entity v_0 and entity e_t, but also takes into account the connecting relationship r between them.

3.4.3. Relationship Reasoning

The final step is to solve our relationship inference over the knowledge graph guided by the trained policy network. For a given initial product v_0, the goal is to find a set of products $\{v_i\}$ that have complementary and substitutable relationships with v_0, and generate interpretable paths.

Since the policy network will guide the agent to search for the path with the highest rewards, it may lead to the same repeated path, which loses the diversity for relationship inference. We employ the Beam Search method [63], which is a heuristic graph search algorithm guided by the action probability and reward to explore the candidate paths. The process is described as Algorithm 2.

It takes as input the initial product v_0, the policy network $\pi(\cdot|(s, \tilde{A}(u)))$, horizon T, and the predefined sampling at step t can be represented as $K_1, \cdots, K_T$. The output is a candidate set of T-hop paths P_T for v_0 with corresponding path generative probabilities Q_T and path rewards R_T. Each path $p_T(v_0, v_n) \in P_T$ ends with a product v_n associated with v_0.

In Algorithm 2, for each step t, the agent initiates its search from the last node of P_t and obtains a pruned action space through the pruning strategy. Then the Beam search based on the action probabilities obtained by the policy network is explored. Here, the agent will select K_t actions with the highest probabilities. The discovered paths are saved in P_{t+1}.

After T horizon, a network composed of T-hop paths P_T can be obtained. All nodes in this network are closely related to the product v_0, and the connecting paths between them can serve as interpretable evidence for the relationships between products. Paths with a length greater than 1 and ending with a product are retained. Finally, we rank and select the top n interpretable paths from P_T according to the path reward in R_T. Products are selected as the final relationship inference results. If there are multiple paths between v_0 and v_n, the path with the highest probability will be selected to interpret the relationship between v_0 and v_n.

Algorithm 2 Policy network guided path reasoning

Input : $u, \pi\left(\cdot \middle| \boldsymbol{s}, \widetilde{A}(u)\right), T, \{K_1, \ldots, K_T\}$

Output : path set P_{T}, probability setQ_{T}, reward setR_{T}

Initialize $P_0 \leftarrow \{\{v_0\}\}, Q_0 \leftarrow \{1\}, R_0 \leftarrow \{0\}$;
for $t \leftarrow 1$ **to** T **do**
Initialize $P_t \leftarrow \varnothing$, $Q_t \leftarrow \varnothing$, $R_t \leftarrow \varnothing$;
forall $\hat{p} \in P_{t-1}$, $\hat{q} \in Q_{t-1}$, $\hat{r} \in R_{t-1}$ **do**
▷ path $\hat{p} \doteq \{v_0, r_1, \ldots, r_{t-1}, e_{t-1}\}$;
Set $s_{t-1} \leftarrow (v_0, e_{t-1}, h_{t-1})$
Get pruned action space $\widetilde{A}_{t-1}(u)$
from environment given state s_{t-1};
▷ $p(a) \doteq \pi\left(a \middle| \boldsymbol{s}_{t-1}, \widetilde{A}_u, t-1\right)$ and $a \doteq (r_t, e_t)$;
Action $A_t \leftarrow \left\{a \middle| \text{rank}\,(p(a)) \leq K_t, \forall a \in \widetilde{A}_{t-1}(u)\right\}$;
forall $a \in A_t$ ***do***
Get s_t, R_{t+1} from environement given action a;
Save new path $\hat{p} \cup \{r_t, e_t\}$ *to* P_t;
Save new probability $p(a)\hat{q}$ *to* Q_t;
Save new reward $R_{t+1} + \hat{r}$ *to* R_t;
end
end
end
Save $\forall\, \hat{p} \in P_T$if the $\hat{p}$ ends with an product;
return filtered P_T, Q_T, R_T;

4. Experiments

To prove the improvements of our model, numerical experiments are conducted on four datasets. We introduce the dataset statistics, evaluation indicators, and experimental settings first. Results of the model's performance, ablation experiments and sensitivity analysis are demonstrated in the following sections.

4.1. Description and Preprocessing of the Dataset

The electronics data set from the open data set of Amazon is used for experimental verification [64,65]. They provide both rich meta-information of products and diverse user review records and have been widely used as domain-specific e-commerce datasets for research [5,40,45]. It contains data related to products and reviews. The product-related data consists of product ID, product description, brand name, product category, related product information, and the size is 1.1 GB. Review data includes user ID, product ID, and review text, and its size is 3.1 GB.

Specifically, the original data set contains some terms that are not helpful or redundant to the relational inference task in this paper, such as "overall rating", and we have deleted these fields. Then, to ensure the connectivity of the knowledge graph and improve the computational efficiency, we filtered out uncommon entities with low correlations. When processing review information, we employ the TF-IDF method to retain key words with a frequency of occurrence less than 5000 and set the minimum value of the TF-IDF score to 0.1 [45]. In extracting entity description information, the maximum length is set to 128 words, which corresponds to the maximum text length supported by the SBERT model.

After cleaning, filtering and word exaction, a knowledge grape is formed which contains five types of entities with a total number of 189,023, and 6 types of 750,100 relationships. Statistical information regarding the entities and relations are presented in Tables 3 and 4. For "*also_viewed*" and "*also_bought*" relationships, 80% of the data are randomly selected as the training set and the remaining 20% is used as the test set. Furthermore, all other relation data is employed as the training set. When training the agent to learn policy network, all of the products in the training set are taken as candidate commodities. However, when testing

the performance of the model, it is necessary to remove the products that have appeared in the training set to get the final relation reasoning list.

Table 3. Statistics for the entities.

Entity name	Description	Number of Entities
Product	Product	10,023
User	User	157,283
Word	Words in review	19,808
Brand	Brand	1302
Category	Category	607

Table 4. Statistics for the relationship data.

Relation Name	Description	Average Number of Relationships per Head Entity
also_viewed	Product $\xrightarrow{\text{also\_viewed}}$ Product	14.474
also_bought	Product $\xrightarrow{\text{also\_bought}}$ Product	29.682
described_by	Product $\xrightarrow{\text{described\_by}}$ Word	7.165
produced_by	Product $\xrightarrow{\text{produced\_by}}$ Brand	0.686
belong_to	Product $\xrightarrow{\text{belong\_to}}$ Category	4.390
purchase	User $\xrightarrow{\text{purchase}}$ Product	44.441

4.2. Design of the Experiments

This study aims to address the following four questions through numerical experiments:

(1) Does the model proposed in this paper outperform similar algorithms?

(2) Does the improved knowledge representation learning model incorporating entity descriptions improve relationship inference performance?

(3) Do parameters such as pruning action space size, path sampling size, multi-hop score patterns, and relationship attributes affect the model's performance, and if so, how do they influence it?

(4) Is the model interpretable, and if so, how does it interpret its inference results?

4.2.1. Evaluation Indicators

To answer these questions, we conduct comparative experiments using four widely used TOP-*k* metrics, and the value of *k* is set to 10. The product list is defined as an orderly list of *k* products related to a specific product. Most of the research in the area has used classical metrics such as normalized discounted cumulative gain (NDCG@10), Recall@10, Precision@10, and Hit-Ratio (HR@10) [5,28,40,45].

NDCG (normalized discounted cumulative gain) is a measure of the effectiveness of a ranking system, taking into account the position of relevant items in the ranked list. After the relationship inference, a list of products related to a given product is given, and let k be the length of the list, NDCG@k can measure the differences between the generated list and the original list, which measure the effectiveness of the model. Note that the higher the NDCG is, the better the performance is.

Recall and Precision are metrics that indicate the probability a real complementary or substitute product has been predicted and the accuracy of the prediction in the relevant product list. Obviously, a higher value is better.

Hit-Ration (HR) is simply the fraction of products which is included in the test dataset. It is another widely used matrix to measure the accuracy of the reasoning method.

4.2.2. Benchmark Algorithms and Experimental Settings

The following algorithms are selected as the benchmark algorithms for effectiveness experiments. They are typical and latest research of knowledge graph relationship inference based on neural network (NN), logic-rule (LR), reinforcement learning (RL), or Hybrid. Detailed characters of the benchmarks are listed in Table 5.

Table 5. Comparisions between the benchmarks.

Models	Research Problem	Reasoning Method	Explainable
DecGCN [6]	Relationship inference	NN-GCN	NO
Cleora [4]	Relationship inference	NN-GNN	NO
PGPR [7]	Recommendation	RL-Policy Network	YES
CAFE [28]	Recommendation	LR-Neural Symbolic	YES
KAPR [5]	Relationship inference	RL-Policy Network	YES
HRRL [40]	Relationship inference	Hybrid-Policy Network & GNN	YES
Our	Relationship inference	RL-Policy Network	YES

The most similar method is KAPR, which is an explainable knowledge-aware path reasoning model based on dynamic policy networks. Due to the different emphasis of the aforementioned algorithms, their dataset requirements vary. Therefore, according to the requirements of each algorithm, the fine-grained attributes of the dataset are adjusted respectively so that the models can run normally.

4.2.3. Parameter Settings

We classify the parameters into three types related to representation learning (RL), policy network (PN) and path reasoning (PR). The settings are illustrated in Table 6. Note that some parameters are used for two or three progress. For example, the embedding size of entities are the same in RL and PN.

Table 6. Parameter Settings.

Category	Parameters	Value
RL	Embedding size of word	100
	Embedding size of entities	100
	Embedding size of relations	100
	Learning rate	0.001
PN	W_1	$W_1 \in R^{400\times512}$
	W_2	$W_2 \in R^{512\times256}$
	W_s	$W_s \in R^{256\times250}$
	W_a	$W_a \in R^{250\times1}$
	Learning rate	0.001
	Batch size	16
PR	Maximum path length T	3
	History length of states K	1
	Embedding size of the state	400
	The maximum action space after pruning D	250
	The path sampling size of Beam Search	[25,5,1]

4.3. Results of Effectiveness Experiments

Comparisons between our model and benchmarks are presented in Table 7. Overall, our model achieves the best performance in each index, validating the effectiveness of this model.

Table 7. Evaluation indicators of each model on the Electronics dataset (@10).

Indicators / Relations / Models	Substitutable Relationship				Complementary Relationship			
	NDCG	**Recall**	**Precision**	**HR**	**NDCG**	**Recall**	**Precision**	**HR**
DecGCN [6]	0.312	0.523	0.125	0.726	0.178	0.237	0.102	0.578
Cleora [4]	0.310	0.498	0.135	0.765	0.173	0.223	0.096	0.533
PGPR [7]	0.314	0.527	0.129	0.730	0.181	0.248	0.108	0.596
CAFE [28]	0.322	0.531	0.137	0.753	0.189	0.257	0.111	0.632
KAPR [5]	0.457	0.585	0.162	0.828	0.222	0.265	0.124	0.687
HRRL [40]	**0.459**	**0.592**	**0.171**	**0.829**	**0.229**	**0.271**	**0.125**	**0.690**
Ours	**0.484**	**0.690**	**0.195**	**0.855**	**0.247**	**0.290**	**0.131**	**0.703**
Improvement (%)	**5.45**	**16.55**	**14.04**	**3.14**	**7.86**	**7.01**	**4.80**	**1.88**

With regards to the neural network-based method DecGCN [6] and the graph embedding-based method Cleora [4], the knowledge-graph-based model has achieved better results. The reason is that the relationships between entities in the knowledge graph ensure the common attribute features among entities used for better relationship inference. As for the deep learning methods, PGPR, CAFE, and HRRL, our model always outperformed in each index, especially for KAPR, which is based on the knowledge graph as well. The main reason is that the entity description information is introduced in the knowledge representation learning, and can provide more comprehensive and accurate evidence for the relationship inference.

4.4. Results of Ablation Experiments

The ablation experiments are conducted to test the effect of two improvements. The first one is the knowledge representation learning model that incorporates entity description information using TransE and SBERT. The other is a modified reward function based on cosine similarity. Specific experimental designs are as follows:

Method variant A: For the first improvement, only the TransE model is used as a replacement for knowledge representation learning.

Method variant B: For the second modification, the reward function used in KAPR [5] is used in the experiment to replace the improved reward function.

The results shown in Table 8 indicate that our model outperforms the variant models, which verifies the effectiveness of two modifications. Particularly, for variant model A, its output is slightly worse compared to KAPR. It demonstrates the importance of using entity descriptions in relationship reasoning.

Table 8. Ablation Experiment Results (@10).

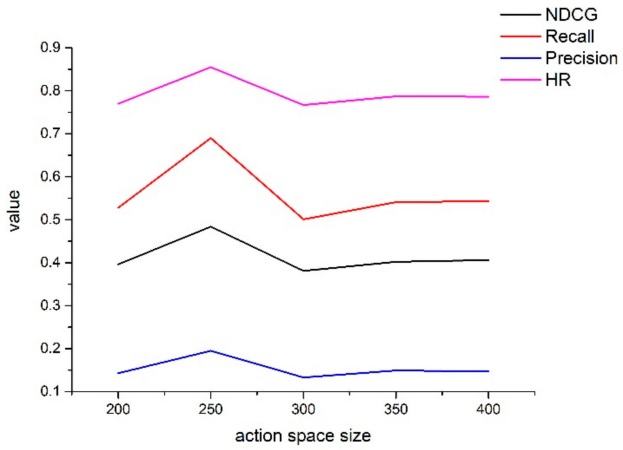
Indicators / Relations / Models	Substitutable Relationship				Complementary Relationship			
	NDCG	**Recall**	**Precision**	**HR**	**NDCG**	**Recall**	**Precision**	**HR**
KAPR [5]	0.457	0.585	0.162	0.828	0.222	0.265	0.124	0.687
variant A	0.376	0.463	0.123	0.743	0.207	0.248	0.111	0.651
variant B	0.469	0.660	0.190	0.833	0.245	0.283	0.128	0.694
Ours	**0.484**	**0.690**	**0.195**	**0.855**	**0.247**	**0.290**	**0.131**	**0.703**

4.5. Sensitivity Analysis

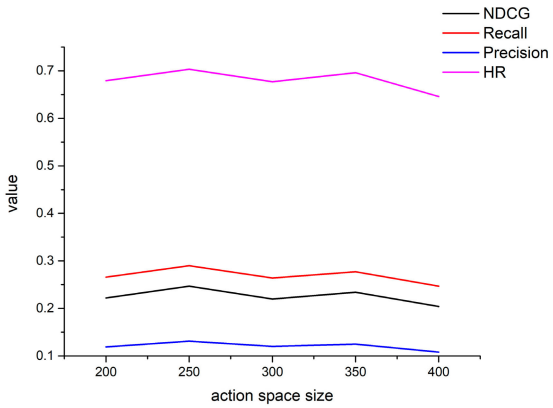
4.5.1. Impact of the Action Space Size

This experiment aims to investigate whether a larger action space helps explore more reasoning paths. Here, the size of the pruned action space size is set from 200 to 400 with an increment of 50. The results presented in Figure 10 indicate that as the action space increases, the output of the model increases at first, reaches the optimum at 250, and then decreases. It implies that the model's performance is influenced by the size of the action space and there exists a proper size. Neither a small size nor a large size will decrease the results. When the action space is small, the model cannot adequately explore the

surrounding action nodes. Meanwhile, under the larger size, the model will explore too many irrelevant action nodes, which also affects the learning outcomes.



(**a**) On substitutable relationship



(**b**) On complementary relationship

Figure 10. Impact of the action space size on the model's performance.

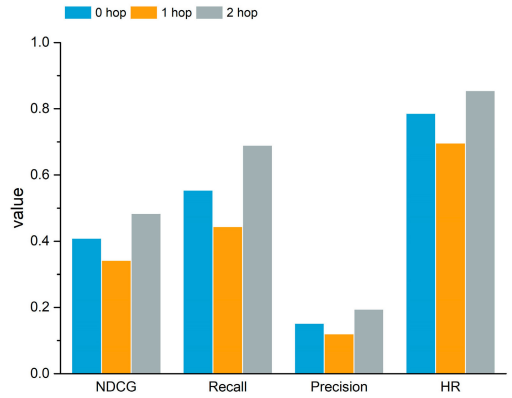
4.5.2. Impact of the Multi-hop Score Modes

When training the policy network, the score function based on 2-hop modes is included as part of the objective function. Here, to test the effect of multi-hop score modes, we first define the 0-hop and 1-hop score functions in the following formulas.

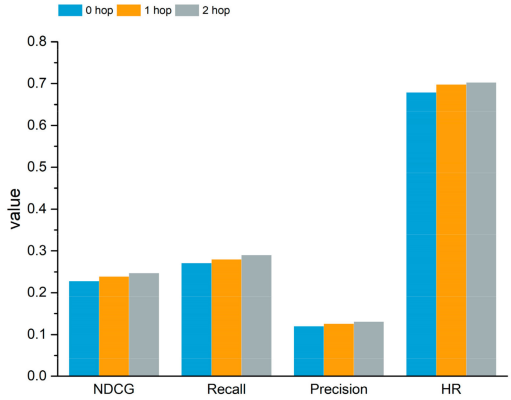
$$f((r,e)|v_0) = \langle \boldsymbol{v}_0, \boldsymbol{e} \rangle + b_e \tag{15}$$

$$f((r,e)|v_0) = \langle \boldsymbol{v}_0 + \boldsymbol{r}, \boldsymbol{e} \rangle + b_e \tag{16}$$

Then, these functions are brought into the objective function Equation (8) for the training policy network instead of 2-hop nodes. The comparison results shown in Figure 11 indicate a significant performance improvement when using the 2-hop score function. This is due to the fact that the multi-hop score functions can capture the interactions between entities that are farther apart. For instance, if two products are purchased by the same user, the 2-hop score function can enhance the relevance between the two products through the 2-hop pattern as "$product \xrightarrow{purchase} user \xrightarrow{purchase} product$", and can increase the likelihood of inferring a certain relationship between the two products.



(**a**) On substitutable relationship inference



(**b**) On complementary relationship inference

Figure 11. Impact of multi-hop modes on relationship inference performance.

4.5.3. Impact Relationship Attributes

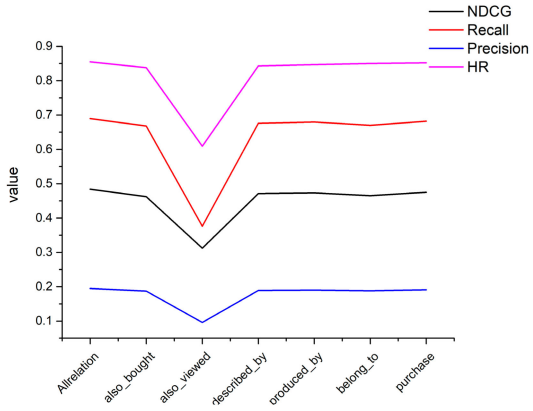
To further explore the impact of relationship attributes on model performance, an experiment was designed by removing the data of different relationship attributes from the data set separately. As can be observed in Figure 12, the absence of "*also_viewed*" data and "*also_bought*" data significantly affects the accuracy of results compared with other types of relationships. The main reason is that the information lies in the "*also_viewed*" data provides most of the knowledge in substitutable relationships. Similarly, the absence of "*also_bought*" data reduces the accuracy of complementary relationships. The absence of other data does not significantly affect their meta-path patterns, and in some cases, the removal of some comparative edge relationships can even improve the performance of the model.

4.6. Analysis on the Interpretability

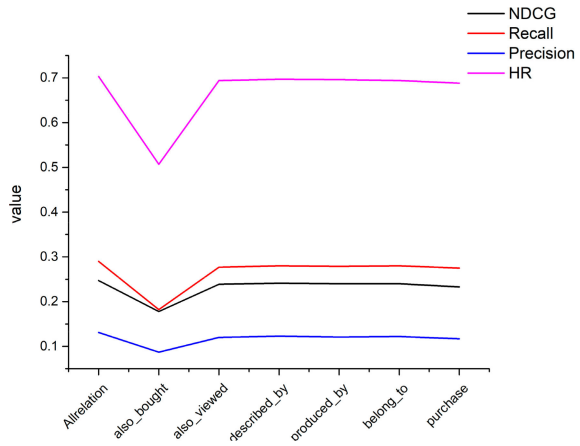
An important feature of the relationship inference model proposed in this paper is its interpretability. After training, the model can not only provide lists of complementary and substitutable products, but also the interpretable paths along with the products. Here, the total sampled paths for the model are 250, and effective paths are defined as paths with a length greater than 1. The proportion of effective paths of complementary and substitutable relationships is 0.78 and 0.53, respectively.

The specific path statistics for the electronics product dataset as well as the results of KAPR are shown in Table 9. The metrics of 'Path/Product' and 'Products' indicate the diversity of interpretable paths, and a higher value means a higher diversity. 'Path/Pair' can represent for the correlation degree of the product pairs, and a smaller number indicates a higher degree. We can find that 'Path/Product' and 'Products' have been increased for

complementary and substitutable relations compared with KAPR, While 'Path/Pair' has reduced a little. The comparisons demonstrated that the model has a strong relationship inference capability. Also, a certain diversity of interpretable paths as well as the correlation degree has been improved compared with the baseline method KAPR.



(**a**) On substitutable relationship inference



(**b**) On complementary relationship inference

Figure 12. Impact of relationship attributes on relationship inference performances.

Table 9. Statistics comparisons of search paths. 'Path/Product' = the average number of effective paths find in the dataset. 'Products' = the average number of products with a specific relationship find for each product. 'Path/Pair' = the average number of paths between a pair of products.

Method	Metrics	Complementary	Substitutable
KAPR	Path/Product	193.89	131.54
	Products	69.01	51.87
	Path/Pair	2.81	2.53
Our	Path/Product	203.89	137.54
	Products	72.54	56.87
	Path/Pair	2.77	2.48

Figure 13 displays examples of interpretable paths. The product in the green box represents complementary products related to the initial product, while that in the red box represents substitutable products. Here, three typical paths are provided. In Case 1, the inference path is "*Wireless earphones* $\xrightarrow{described\_by}$ *Wireless* $\xrightarrow{described\_by}$ *Wireless earphones* $\xrightarrow{described\_by}$ *Phone*". Since different types of wireless earphones are described by the same word "*wireless*", it can be inferred that they are in a substitutable relationship. Additionally, through a common purchase relationship link, it can be inferred that wireless earphones

and phones have a complementary relationship. For the inference path in Case 2, the game controller and rechargeable battery are purchased by the same person Jack. So, it can be inferred that they have a complementary relationship. Through the common browsing relationship link between the rechargeable battery and wireless receiver, it is inferred that the game controller and wireless receiver are complementary. In Case 3, there is a common purchase relationship link between two laptops, the USB disk, and the mouse. It is demonstrated that laptops and the mouse are complementary, and the two laptops are in a substitutable relationship.

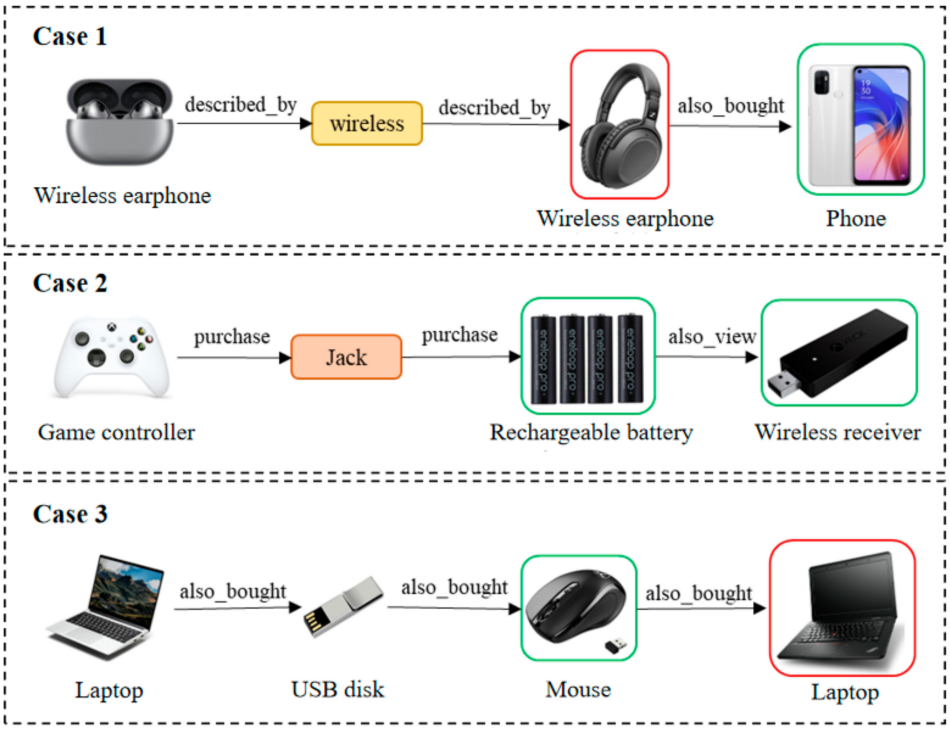


Figure 13. Examples of the interpretable paths.

Numerical experiments indicate that the model can not only infer complementary and substitutable relationships between products with high qualities but also can find different inference paths for the inference results, making the model interpretable.

4.7. Discussions

The results of the numerical experiments will be discussed in this subsection. Through the effectiveness experiments, ablation experiments, sensitivity analysis as well as the analysis of the interpretability, we can find the following results.

(1) Compared with the knowledge graph reasoning models based on neural network, the path reasoning method can explain the relationships, which increases the reliability of the results and can provide more insights for retailers' decision makings.

(2) Compared with the explainable models, the proposed method can retain a better precision and recall rate. It is mainly due to the fact that the unstructured description data has been used for better knowledge exaction.

(3) Through modification of the reward function and pruning strategy, the diversity of paths and relationships between product pairs have been increased.

Meanwhile, nothing is perfect. The limitation of the model lies in the following aspects.

(1) The model is dependent on the meta-path patterns. To ensure the interpretability of the inference path, the meta-path is adopted in this paper. However, meta-path patterns vary in different data sets, limiting the ability to explore knowledge in a new knowledge graph and also affect the quality of the results when the quality of the meta-path is different.

(2) The model is not good for adaptability. When more different structured data are provided, the representation learning has to be redesigned. Moreover, the method has its limitations for large-scale problems since the training is space and time-consuming.

5. Conclusions

In this paper, we have addressed the problem of inferring product relationships from plenty of heterogeneous data in e-commerce.

We presented an improved knowledge-graph based reasoning model, which can infer interpretable path for identifying substitutable and complementary products. Our methodology consists of the following steps. First, a knowledge graph was constructed from multi-source data. Then, we investigated a knowledge representation learning model which integrates both structured and unstructured knowledge. Finally, the task of inferring relationships was transformed into a Markov Decision Process (MDP), which helped to derive interpreted paths consisting of substitutable and complementary products.

Our research contributes in the following three aspects:

(1) A two-stage knowledge representation learning method based on TransE and SBERT has been proposed. It provides an efficient way to enhance the knowledge learning ability by leveraging both structured triplet knowledge and unstructured textual data.

(2) A proper pruning strategy and a modified reward function that incorporate cosine similarity along the inference path has been used for the dynamic policy network training. It enhances the precision of reasoning within the MDP.

(3) Extensive numerical experiments on real-world datasets demonstrates the effectiveness of our improvements.

The knowledge-graph-based relationship inferring model provides credible and reliable results, offering potential application value for online retailers in decision-making processes such as recommendations, bundling promotions and warehouse storage assignments. Meanwhile, the following questions are worth of future research.

(1) How to explore more flexible patterns or rule reorganization to reduce the dependence on meta-path patterns can be further investigated

(2) How to leverage knowledge from more data. There is more relevant data that contains relationship information, such as the prices, customer preferences and differences in purchasing patterns. How to use these data to improve the adaptability of the model is worthy of study in the future.

Due to the numerous abbreviations in this paper, for the sake of clarity, the full forms of the abbreviations are provided in Table 10 for reference.

Table 10. Full name of abbreviations.

Abbreviation	Full Name
TransE	Translating Embedding
SBERT	Sentence Bidirectional Encoder Representation from Transformers
NDCG	Normalized Discounted Cumulative Gain
HR	Hit-Ration
BB2Vec	Baskets and Browsing to Vector
RSC	Rating Estimation Framework
DecGCN	Decoupled Graph Convolutional Network
MDP	Markov Decision Process
Node2Vec	Node to Vector
Struct2Vec	Structure to Vector
GraphGAN	Graph Representation Learning with Generative Adversarial Nets
prod2vec	Product to Vector
SVM	Support Vector Machine
LDA	Linear Discriminant Analysis
PMSC	Path-constrained Framework

Table 10. *Cont.*

Abbreviation	Full Name
LVAE	Linked Variational Autoencoder
SPEM	Substitute Products Embedding Model
CAFE	Coarse-to-Fine Neural Symbolic Reasoning for Explainable Recommendation
TransAt	Translating Embeddings with Attributes and Types
TransA	Translating Embeddings with Attributes
CTransR	Complex Translational Relational Model
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
GNN	Graph Neural Networks
GCN	Graph Convolutional network
MCKRL	Multi-source Information Combined Knowledge Representation Learning
KGDL	Learning Knowledge Graph Embedding with Entity Descriptions based on LSTM Networks
LSTM	Long Short-Term Memory
GAE	Graph Auto-encoders
STGNN	Spatiotemporal Graph Neural Networks
SACN	Structure Aware Convolutional Network
ConvE	Convolutional Translating Embedding
KAPR	Knowledge-aware Path Reasoning
MFI	Multi-feature Inference
AutoRec	Autoencoder-based Recommendation
BGN	Bundle Generation Network
BPR	Bundle Personalization Rank
ADLFM	Adaptive Deep Latent Factor Model
IPKG	Inferring Complementary and Substitutable Products based on a Knowledge Graph
KGRR	Knowledge-graph-based Relationship Reference
BERT	Bidirectional Encoder Representation from Transformers
RL	Reinforcement Learning
ReLU	Rectified Linear Unit
TF-IDF	Term Frequency-Inverse Document Frequency
NN	Neural Network
LR	Logic-rule
PGPR	Policy-guided Path Reasoning
HRRL	Heterogeneous Relational Reasoning in Knowledge Graphs with Reinforcement Learning
PN	Policy Network
PR	Path Reasoning

The table is constructed based on the order of abbreviation appearances in this paper.

Author Contributions: Conceptualization, Y.F.; methodology, Y.F. and Y.D.; software, Y.F. and J.Y.; validation, J.Y. and Y.D.; investigation, J.Y., Y.D. and X.L.; data curation, Y.D. and X.L.; writing—original draft preparation, Y.D.; writing—review and editing, Y.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Natural Science Foundation of China (71801028), Social Science Planning Fund Program of Liaoning Province (L17BGL015), and the Fundamental Research Funds for the Central Universities (DMU: 3132023270).

Data Availability Statement: The data presented in this study are openly available in reference [64,65].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhao, T.; McAuley, J.; Li, M.Y.; King, I. Improving recommendation accuracy using networks of substitutable and complementary products. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 3649–3655.
2. Tian, Y.; Lautz, S.; Wallis, A.O.; Renaud, L. Extracting complements and substitutes from sales data: A network perspective. *Epj Data Sci.* **2021**, *10*, 45. [CrossRef]
3. Trofimov, I. Inferring complementary products from baskets and browsing sessions. *arXiv* **2018**, arXiv:1809.09621.
4. Tkachuk, S.; Wroblewska, A.; Dabrowski, J.; Lukasik, S. Identifying Substitute and Complementary Products for Assortment Optimization with Cleora Embeddings. In Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Padua, Italy, 18–23 July 2022; pp. 1–7.
5. Yang, Z.J.; Ye, J.B.; Wang, L.L.; Lin, X.; He, L. Inferring substitutable and complementary products with Knowledge-Aware Path Reasoning based on dynamic policy network. *Knowl.-Based Syst.* **2022**, *235*, 107579. [CrossRef]
6. Liu, Y.D.; Gu, Y.L.; Ding, Z.Y.; Gao, J.C.; Guo, Z.Y.; Bao, Y.J.; Yan, W.P. Decoupled Graph Convolution Network for Inferring Substitutable and Complementary Items. In Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM), Virtual Event, 19–23 October 2020; pp. 2621–2628.
7. Guan, S.P.; Jin, X.L.; Jia, Y.T.; Wang, Y.Z.; Cheng, X.Q. Knowledge Reasoning Over Knowledge Graph: A Survey. *J. Softw.* **2018**, *29*, 2966–2994.
8. Bliss, C.A.; Frank, M.R.; Danforth, C.M.; Dodds, P.S. An evolutionary algorithm approach to link prediction in dynamic social networks. *J. Comput. Sci.* **2014**, 750–764. [CrossRef]
9. Das, S.; Das, S.K. A probabilistic link prediction model in time-varying social networks. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017.
10. Balvir, S.U.; Raghuwanshi, M.M.; Singh, K.R. A Comprehensive Survey on Learning Based Methods for Link Prediction Problem. In Proceedings of the 2023 6th International Conference on Information Systems and Computer Networks (ISCON) GLA University, Mathura, India, 3–4 March 2023.
11. Kök, A.G.; Fisher, M.L.; Vaidyanathan, R. *Assortment Planning: Review of Literature and Industry Practice*; Springer: Boston, MA, USA, 2015; pp. 175–236.
12. Raeder, T.; Chawla, N.V. Market basket analysis with networks. *Soc. Netw. Anal. Min.* **2011**, *1*, 97–113. [CrossRef]
13. 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 (KDD '16), Association for Computing Machinery, New York, NY, USA, 3 July 2016; pp. 855–864.
14. Ribeiro, L.F.R.; Saverese, P.H.P.; Figueiredo, D.R. Struc2vec: Learning node representations from structural identity. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 385–394.
15. Wang, H.; Wang, J.; Wang, J. Learning Graph Representation with Generative Adversarial Nets. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 3090–3103. [CrossRef]
16. Mihajlo, G.; Vladan, R.; Nemanja, D.; Narayan, B.; Jaikit, S.; Varun, B.; Doug, S. E-commerce in Your Inbox: Product Recommendations at Scale Categories and Subject Descriptors. In Proceedings of the SIGKDD 2015, 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 1809–1818.
17. Co, J.M.; Fernandez, P. Link prediction in a weighted network using support vector machine. In Proceedings of the 6th International Workshop on Computer Science and Engineering (WCSE 2016), Tokyo, Japan, 17–19 June 2016.
18. McAuley, J.; Pandey, R.; Leskovec, J. Inferring networks of substitutable and complementary products. *arXiv* **2015**, arXiv:1506.08839.
19. Wang, Z.; Jiang, Z.; Ren, Z.; Tang, J.; Yin, D. A path-constrained framework for discriminating substitutable and complementary products in ecommerce. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Marina Del Rey, CA, USA, 5–9 February 2018; pp. 619–627.
20. Rakesh, V.; Wang, S.; Shu, K.; Liu, H. Linked variational autoencoders for inferring substitutable and supplementary items. In Proceedings of the 12th ACM International Conference on Web Search and Data Mining (WSDM), Melbourne, Australia, 11–15 February 2019; pp. 438–446.
21. Zhang, S.; Yin, H.; Wang, Q.; Chen, T.; Chen, H.; Nguyen, Q.V.H. Inferring substitutable products with deep network embedding. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 4306–4312.
22. Wang, H.; Zhang, F.; Zhang, M.; Leskovec, J.; Zhao, M.; Li, W.; Wang, Z. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), Anchorage, AK, USA, 4–8 August 2019.
23. Ren, H.; Dai, H.; Dai, B.; Chen, X.; Yasunaga, M.; Sun, H.; Schuurmans, D.; Leskovec, J.; Zhou, D. Lego: Latent execution-guided reasoning for multi-hop question answering on knowledge graphs. In Proceedings of the International Conference on Machine Learning (ICML), Vienna, Austria, 18–24 July 2021.
24. Ruiz, C.; Zitnik, M.; Leskovec, J. Identification of disease treatment mechanisms through the multiscale interactome. *Nat. Commun.* **2021**, *12*, 1796. [CrossRef]

25. Muggleton, S.H. Inductive Logic Programming. *New Gener. Comput.* **1991**, *8*, 295–318. [CrossRef]
26. Wang, W.Y.; Mazaitis, K.; Lao, N.; Cohen, W.W. Efficient reasoning and learning in a large knowledge base—Reasoning with extracted information using a locally groundable first-order probabilistic logic. *Mach. Learn.* **2015**, *100*, 101–126. [CrossRef]
27. Jang, S.; Megawati; Choi, J.; Yi, M.Y. Semi-Automatic Quality Assessment of Linked Data without Requiring Ontology. In Proceedings of the 14th International Semantic Web Conference (ISWC), Bethlehem, PA, USA, 11 October 2015; pp. 45–55.
28. Xian, Y.K.; Fu, Z.H.; Zhao, H.D.; Ge, Y.Q.; Chen, X.; Huang, Q.Y.; Geng, S.J.; Qin, Z.; De Melo, G.; Muthukrishnan, S.; et al. CAFE: Coarse-to-Fine Neural Symbolic Reasoning for Explainable Recommendation. In Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM), Virtual Event, 19–23 October 2020; pp. 1645–1654.
29. Feng, H.J.; Duan, L.; Zhang, B.Y. Overview on Knowledge Reasoning for Knowledge Graph. *Comput. Syst. Appl.* **2021**, *30*, 21–30.
30. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 2787–2795.
31. Qian, W.; Fu, C.; Zhu, Y.; Cai, D.; He, X.F. Translating Embeddings for Knowledge Graph Completion with Relation Attention Mechanism. In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, 13–19 July 2018; pp. 4286–4292.
32. Xiao, H.; Huang, M.L.; Hao, Y.; Zhu, X.Y. TransA: An adaptive approach for knowledge graph embedding. *arXiv* **2015**, arXiv:1509.05490.
33. Lin, Y.K.; Liu, Z.Y.; Sun, M.S.; Liu, Y.; Zhu, X. Learning Entity and Relation Embeddin-gs for Knowledge Graph Completion. In Proceedings of the 29th Association-for-the-Advancement-of-Artificial-Intelligence (AAAI) Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2181–2187.
34. Zhao, F.; Xu, T.; Jin, L.J.Q.; Jin, H. Convolutional Network Embedding of Text-Enhanced Representation for Knowledge Graph Completion. *IEEE Internet Things J.* **2021**, *8*, 16758–16769. [CrossRef]
35. Xia, G.B.; Li, R.X.; Gu, X.W.; Liu, W. Knowledge Representation Learning Based on Multi-source Information Combination. *J. Front. Comput. Sci. Technol.* **2022**, *16*, 591–597.
36. Chen, W.R.; Hong, D.P.; Zheng, C. Learning knowledge graph embedding with entity descriptions based on LSTM networks. In Proceedings of the 2020 IEEE International Symposium on Product Compliance Engineering-Asia (ISPCE-CN), Chongqing, China, 6–8 November 2020; pp. 1–7.
37. Jagvaral, B.; Lee, W.K.; Roh, J.S.; Kim, M.S.; Park, Y.T. Path-based reasoning approach for knowledge graph completion using CNN-BiLSTM with attention mechanism. *Expert Syst. Appl.* **2020**, *142*, 112960. [CrossRef]
38. Nguyen, D.Q.; Tong, V.; Phung, D.; Nguyen, D.Q. Node co-occurrence based graph neural networks for knowledge graph link prediction. In Proceedings of the 15th ACM International Conference on Web Search and Data Mining (WSDM), Virtual Event, 21–25 February 2022; pp. 1589–1592.
39. Wang, Q.; Hao, Y.S.; Chen, F. Deepening the IDA* algorithm for knowledge graph reasoning through neural network architecture. *Neurocomputing* **2021**, *429*, 101–109. [CrossRef]
40. Saebi, M.; Kreig, S.; Zhang, C.X.; Jiang, M.; Kajdanowicz, T.; Chawla, N.V. Heterogeneous relational reasoning in knowledge graphs with reinforcement learning. *Inf. Fusion* **2022**, *88*, 12–21. [CrossRef]
41. Wang, S.; Wei, X.; Nogueira dos Santos, C.N.; Wang, Z.; Nallapati, R.; Arnold, A.; Xiang, B.; Yu, P.S.; Cruz, I.F. Mixed-curvature multi-relational graph neural network for knowledge graph completion. In Proceedings of the 30th Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; ACM: New York, NY, USA, 2021; pp. 1761–1771.
42. Khaled, A.; Elsir, A.M.T.; Shen, Y. TFGAN: Traffic forecasting using generative adversarial network with multigraph convolutional network. *Knowl.-Based Syst.* **2022**, *249*, 10899. [CrossRef]
43. Shang, C.; Tang, Y.; Huang, J.; Bi, J.B.; He, X.D.; Zhou, B.W. End-to-End Structure-Aware Convolutional Networks for Knowledge Base Completion. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 3060–3067.
44. Xiong, W.H.; Hoang, T.; Wang, W.Y. DeepPath: A reinforcement learning method for knowledge graph reasoning. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9–11 September 2017; pp. 564–573.
45. Xian, Y.K.; Fu, Z.H.; Muthukrishnan, S.; De Melo, G.; Zhang, Y.F. Reinforcement Knowledge Graph Reasoning for Explainable Recommendation. In Proceedings of the 42nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Paris, France, 21–25 July 2019; pp. 285–294.
46. Sedhain, S.; Menon, A.K.; Sanner, S.; Xie, L. Autorec: Autoencoders meet collaborative filtering. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 111–112.
47. Bai, J.; Zhou, C.; Song, J.; Qu, X.; An, W.; Li, Z.; Gao, J. Personalized Bundle List Recommendation. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 60–71.
48. Kim, D.; Park, C.; Oh, J.; Lee, S.; Yu, H. Convolutional matrix factorization for document context-aware recommendation. In Proceedings of the 10th ACM conference on recommender systems, Boston, MA, USA, 15–19 September 2016; pp. 233–240.
49. Michael, D.; Xavier, B.; Pierre, V. Convolutional neural networks on graphs with fast localized spectral filtering. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016.

50. Bronstein, M.M.; Bruna, J.; LeCun, Y.; Szlam, A.; Vandergheynst, P. Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Process. Mag.* **2017**, *34*, 18–42. [CrossRef]
51. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1025–1035.
52. Steffen, R.; Christoph, F.; Zeno, G. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* **2012**, arXiv:1205.2618.
53. Pathak, A.; Gupta, K.; McAuley, J. Generating and Personalizing Bundle Recommendations on "Steam". In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, 7–11 August 2017; pp. 1073–1076.
54. Han, J.Y.; Zheng, L.; Xu, Y.B. Adaptive Deep Modeling of Users and Items Using Side Information for Recommendation. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 737–748. [CrossRef] [PubMed]
55. Chen, J.W.; Zhuang, F.Z.; Hong, X. Attention-Driven Factor Model for Explainable Personalized Recommendation. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 909–912.
56. Hada, D.V.; Vijaikumar, M.; Shevade, S.K. ReXPlug: Explainable Recommendation Using Plug-and-Play Language Model. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 11–15 July 2021; pp. 81–91.
57. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Hong Kong, China, 3–7 November 2019; pp. 3982–3992.
58. Du, W.Q.; Li, B.C.; Wang, R. Representation Learning of Knowledge Graph Integrating Entity Description and Entity Type. *J. Chin. Inf. Process.* **2020**, *34*, 50–59.
59. Liu, Y.H.; Ott, M.; Goyal, N.; Du, J.F.; Joshi, M.; Chen, D.Q.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
60. Williams, R.J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **1992**, *8*, 229–256. [CrossRef]
61. Zhao, K.Z.; Wang, X.T.; Zhang, Y.R.; Zhao, L.; Liu, Z.; Xing, C.X.; Xie, X. Leveraging Demonstrations for Reinforcement Recommendation Reasoning over Knowledge Graphs. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Virtual Event, 25–30 July 2020; pp. 239–248.
62. Song, W.P.; Duan, Z.J.; Yang, Z.Q.; Zhu, H.; Zhang, M.; Tang, J. Explainable Knowledge Graph-based Recommendation via Deep Reinforcement Learning. *arXiv* **2019**, arXiv:1906.09506.
63. Kumar, A.; Vembu, S.; Menon, A.K.; Elkan, C. Beam search algorithms for multilabel learning. Machine learning. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD), Bristol, UK, 21–28 September 2012; pp. 65–89.
64. Amazon Review Data. Available online: https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/ (accessed on 5 May 2022).
65. Ni, J.; Li, J.; McAuley, J. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In Proceedings of the Conference on Empirical Methods in Natural Language Processing/9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 188–197.