Multi-modal Knowledge Graphs for Recommender Systems

Rui Sun[1†], Xuezhi Cao[2], Yan Zhao[3], Junchen Wan[2], Kun Zhou[4], Fuzheng Zhang[2]
Zhongyuan Wang[2] and Kai Zheng[1*]
[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China
[2]Meituan-Dianping Group
[3]Aalborg University, Danmark
[4]School of Information, Renmin University of China
sunrui@std.uestc.edu.cn, {caoxuezhi, zhangfuzheng, wangzhongyuan02}@meituan.com, yanz@cs.aau.dk, {wan_junchen, francis_kun_zhou}@163.com, zhengkai@uestc.edu.cn

ABSTRACT

Recommender systems have shown great potential to solve the information explosion problem and enhance user experience in various online applications. To tackle data sparsity and cold start problems in recommender systems, researchers propose knowledge graphs (KGs) based recommendations by leveraging valuable external knowledge as auxiliary information. However, most of these works ignore the variety of data types (e.g., texts and images) in multi-modal knowledge graphs (MMKGs). In this paper, we propose Multi-modal Knowledge Graph Attention Network (MKGAT) to better enhance recommender systems by leveraging multi-modal knowledge. Specifically, we propose a multi-modal graph attention technique to conduct information propagation over MMKGs, and then use the resulting aggregated embedding representation for recommendation. To the best of our knowledge, this is the first work that incorporates multi-modal knowledge graph into recommender systems. We conduct extensive experiments on two real datasets from different domains, results of which demonstrate that our model MKGAT can successfully employ MMKGs to improve the quality of recommendation system.

CCS CONCEPTS

• **Information systems** → **Recommender systems**;

KEYWORDS

Recommender systems, Graph Convolutional Networks, Multi-modal Knowledge Graph

ACM Reference Format:
Rui Sun[1†], Xuezhi Cao[2], Yan Zhao[3], Junchen Wan[2], Kun Zhou[4], Fuzheng Zhang[2] and Zhongyuan Wang[2] and Kai Zheng[1*]. 2020. Multi-modal Knowledge Graphs for Recommender Systems. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3411947

*Corresponding author.
†This paper was done during the internship in Meituan-Dianping Group.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM '20, October 19–23, 2020, Virtual Event, Ireland
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3411947

1 INTRODUCTION

Recently, knowledge graphs (KGs) are widely used in recommender systems (i.e., KG-based recommendation) due to their comprehensive auxiliary data for effective recommendation [24, 28]. Specifically, the KG-based recommendation alleviates the sparsity problem of user-item interactions and the cold start problem by introducing high quality side information (KGs). These problems often arise in Collaborative Filtering (CF) [11] based methods.

However, existing KG-based recommendation methods largely ignore the multi-modal information, such as images and text descriptions of items. Those visual or textual features may play a significant role in recommendation systems. For instance, before watching a movie, users tend to watch the trailer or read some related film reviews. When going to a restaurant for dinner, users normally browse the pictures of dishes or the reviews of the restaurant on some online platforms, such as Yelp [1] or Dianping [2] at first. So it is necessary to introduce these multi-modal information into knowledge graph. The benefit is that multi-modal knowledge graphs (MKGs) introduce visual or textual information into the knowledge graph, regarding image or text as an entity or as an attribute of the entity. It is a more general way of acquiring external multi-modal knowledge, without giving the expert definitions of visual or textual information. A simple example of MKGs is shown in the figure 1.

The knowledge graph representation learning plays a key role for the KG-based recommendation. The KG-based recommendation models usually use the knowledge graph representation model to learn the embedding of the KGs entities, which are then fed into the downstream recommendation task. There are two types of multi-modal knowledge graph representation learning: the feature-based methods and the entity-based methods.

The feature-based methods [17, 30] treat the modal information as an auxiliary feature of the entity. It extends the translational models (TransE) [2] by considering visual representations, which are extracted from images corresponding to the knowledge graph entities. The energy of a triple (e.g., the scoring function for triples in TransE) is defined in terms of the structure of the KGs as well

[1]https://www.yelp.com/
[2]https://www.dianping.com/

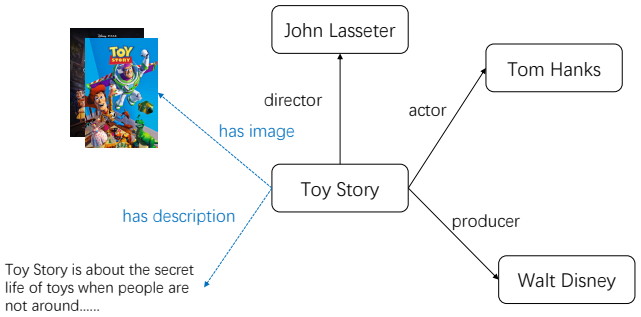

Figure 1: Example of a multi-modal knowledge graph.

as the visual representation of the entities. However, the feature-based methods pose relatively requirements on the data source of the knowledge graph since it requires that every entity in the knowledge graph has multi-modal information.

In order to address the strict requirement on KGs data source, the entity-based methods [19] is proposed. The entity-based methods treat different types of information (e.g., texts and images) as relational triples of the structured knowledge instead of auxiliary features, i.e., first-class citizens of the knowledge graph. It introduces visual and textual information by considering new relation, such as *hasImage* (denoting if an entity has image information) and *hasDescription* (denoting if an entity has text information to describe it). Then, the entity-based method processes each triple, (h, r, t), by independently applying translational models [2] or Convolutional Neural Network (CNN) based models [18] to learn the knowledge graph embedding, where h and t denote a head and tail entity respectively, r is the relationship (e.g., *hasImage* and *hasDescription*) between h and t.

Although the entity-based methods solve the problem of high demand for data sources of MKGs in the feature-based methods, it only focuses on the reasoning relation between entities and ignores the multi-modal information fusion. In fact, multi-modal information is usually used as an auxiliary information to enrich the information of other entities. Therefore, we need a direct interactive way to explicitly fuses the multi-modal information into its corresponding entity before modeling the reasoning relation between the entities.

Considering the limitations of the existing solutions, we believe it is essential to develop a MKGs representation model that can exploit MKGs in an efficient manner. Specifically, the model should satisfy two conditions: 1) low requirements for MKGs data sources, 2) multi-modal information fusion is considered while preserving the reasoning relation between entities. Towards this end, we follow the entity-based methods to construct the multi-modal knowledge graph. And then, we propose Multi-modal Knowledge Graph Attention Network (MKGAT), which models the multi-modal knowledge graph from two aspects: 1) entity information aggregation, which aggregates the entity's neighbor node information to enrich the entity itself, 2) entity relation reasoning, which constructs reasoning relations by the scoring function of the triple (e.g., TransE). We first propose a new method to improve the graph attention neural network (GATs), which aggregates neighbor entities while taking into account the relation in the knowledge graph to complete entity information aggregation. And then we use a translational model to model the reasoning relation between entities. A visible advantage of our MKGAT model lies in that it does not require each entity in the knowledge graph to have multi-modal information, which means it has no particularly high requirements for knowledge graph data. Besides, the MKGAT model does not process each knowledge graph triple independently but aggregates the neighbor information of the entity. As a result, it can learn the entity embedding that fuses other modal information better. The primary contributions of this work can be summarized as follows:

- To the best of our knowledge, this is the first work to introduce a multi-modal knowledge graph into a recommendation system.
- We develop a new MKGAT model, which employs information propagation on the multi-modal knowledge graph, to obtain better entity embedding for recommendation.
- Extensive experiments conducted on two large-scale real-world datasets demonstrate the rationality and effectiveness of our model.

The remainder of this paper is organized as follows. Section 2 surveys the related work. The preliminary concepts are introduced in Section 3. We then present the MKGAT model in Section 4, followed by the experimental results in Section 5. Section 6 concludes this paper.

2 RELATED WORK

In this section, we introduce existing works that are related to our research, including multi-modal knowledge graph and KG-based recommendation.

2.1 Multi-modal Knowledge Graphs

Multi-modal Knowledge Graphs (MKGs) enriches the types of knowledge by introducing information of other modals into the traditional KGs. Entity images or entity description could provide significant visual or textual information for knowledge representation learning. Most conventional methods learn knowledge representations merely from structured triples, ignoring the variety of data types (such as texts and images) that are often used in knowledge base. Recently, several efforts have been made to explore the multi-modal knowledge graph representation learning. These works have proved that the multi-modal knowledge graph plays an important role in knowledge graph completion and triple classification [5, 17, 30]. From the perspective of knowledge graphs construction, multi-modal knowledge graph representation learning works can be categorized into two types: features-based methods and entity-based methods.

Features-based methods. [17, 30] treat multi-modal information as auxiliary features of the entity. These methods extend TransE [2] by taking visual representations into account. The visual representations can be extracted from images associated with the knowledge graph entities. In these methods, the energy of a triple (e.g., the scoring function for triple in TransE) is defined in terms of the structure of knowledge graph as well as visual representations of entities, which means each entity must contain the image attribute. However, in real scenes, some entities do not contain multi-modal information. So this method cannot be widely used.

Entity-based methods. [19] treats different modal information (e.g., texts and images) as relational triples of the structured knowledge instead of predetermined features. In these works, multi-modal information is considered as first-class citizens of the knowledge graphs. And then entity-based methods use CNN-based KGE method to train the knowledge graph embedding. Nevertheless, existing entity-based methods process each triple independently ignoring multi-modal information fusion, which is not friendly to multi-modal tuples.

As multi-modal knowledge graph has only been introduced in recent years, there are only limited research works in this direction.

2.2 KG-based Recommendation

Recently, some researches have attempted to leverage KGs structure for recommendation, which can be categorized into three types, embedding-based methods, path-based methods and unified methods.

Embedding-based methods. Embedding-based methods [23, 25, 27] first use Knowledge Graph Embedding (KGE) [27] algorithms to preprocess knowledge graph and then use the learned entity embeddings in a recommendation framework, which unifies various types of side information in the CF framework. Collaborative Knowledge base Embedding (CKE) [35] combines a Collaborative Filtering (CF) module with knowledge embedding, text embedding, and image embedding of items in a unified Bayesian framework. Deep Knowledge-based Network (DKN) [25] treats entity embeddings and word embeddings as different channels, and then uses a Convolutional Neural Networks (CNN) framework to combine them together for news recommendation.

Embedding-based methods show high flexibility in utilizing knowledge graph to assist recommender systems, but the KGE algorithms (translational models or CNN-based models) adopted in these methods is not suitable for multi-modal tuples (The reason is the same as the entity-based method in MKGs). In other words, these methods are not friendly to multi-modal knowledge graphs.

Path-based methods. Path-based methods [33, 36] explore various patterns of connections among items in knowledge graph to provide additional guidance for recommendations. For example, regarding knowledge graph as a Heterogeneous Information Network (HIN), Personalized Entity Recommendation (PER) [33] and meta-graph based recommendation [36] extract the meta-path/meta-graph based latent features to represent the connectivity between users and items along different types of relation paths/graphs.

Path-based methods make use of knowledge graph in a more natural and intuitive way, but they rely heavily on manually designed meta-paths, which is hard to be optimized in practice. Another concern is that it is impossible to design hand-crafted meta-paths in certain scenarios where entities and relations do not belong to one domain.

Unified methods. Embedding-based methods leverage the semantic representations of entities in the KGs for recommendation, while path-based methods use the patterns of connections among entities in KGs. Both of them utilize only one aspect of information in the KGs. To fully exploit the information in the KGs for better recommendations, unified methods have been proposed, which integrate the semantic representations of entities and relations, as well as the patterns of connectivity information. However, unified methods also depend on knowledge graph embedding technology. Translational models are widely used for training knowledge graph embeddings. Representative works includes attention-enhanced knowledge-aware user preference model (AKUPM) [12] and knowledge graph attention network (KGAT) [28]. They process each triple independently without considering multi-modal information fusion. Similar to embedding-based methods, Unified methods are not friendly to multi-modal knowledge graphs.

3 PROBLEM FORMULATION

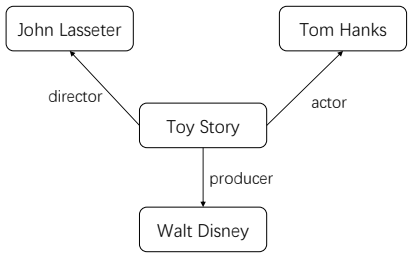

Figure 2: Example of a knowledge graph.

In this section, we introduce a set of preliminary concepts, and then formulate the task of multi-modal knowledge graph based recommendation.

Definition 1 (Knowledge Graph). *In order to improve the recommend performance, we consider side information of items in knowledge graphs. Typically, such auxiliary data consists of real-world entities and relationships among them to profile an item.*

A Knowledge Graph (KG), $G = (V, E)$, is a direct graph, where V denotes the node set and E denotes the edge set. The nodes are entities and edges are subject-property-object triple facts. Each edge belongs to a relation type $r \in \mathcal{R}$, where $\mathcal{R}$ is a set of relation types. Each edge in the form of (head entity, relation, tail entity) (denoted as (h, r, t), where $h, t \in V$, $r \in \mathcal{R}$) indicates a relationship of r from head entity h to tail entity t.

Figure 2 illustrates an example of a knowledge graph, in which a movie (called Toy Story) described by its director, actor, and producer. We can use *(Toy Story, DirectorOf, John Lasseter)* to state the fact that Toy Story is directed by John Lasseter.

Definition 2 (Multi-modal Knowledge Graph). *A Multi-modal Knowledge Graphs (MKGs) are certain type of knowledge graphs, which additionally introduce multi-modal entities (e.g., texts and images) as first-class citizens of the knowledge graph.*

Taking Figure 1 as an example that shows a multi-modal knowledge graph, we use *(Toy Story, hasImage, a picture of film promotion)* to represent the fact that the movie entity (i.e., Toy Story) has an image entity, which describes some visual information of this movie entity.

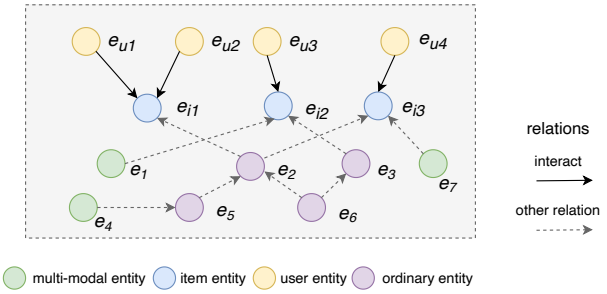

Figure 3: Example of a collaborative knowledge graph.

Definition 3 (Collaborative Knowledge Graph). *A Collaborative Knowledge Graph (CKG) encodes user behaviors and item knowledge as a unified relational graph. A CKG first define a user-item bipartite graph, which is formulated as* $\{(e_u, y_{ui}, e_i) \,|e_u \in \mathcal{U}, e_i \in \mathcal{I})\}$ *where* e_u *is a user entity,* y_{ui} *denotes the link between user* u *and item* i*,* e_i *denotes an item entity,* $\mathcal{U}$ *and* $\mathcal{I}$ *separately denote the user and item sets. When there is an interaction between* e_u *and* e_i*,* $y_{ui} = 1$*; otherwise,* $y_{ui} = 0$*. Then, CKG incorporates the user-item bipartite graph into the knowledge graph, in which each user's behavior is represented as a triplet,* $(e_u, Interact, e_i)$*.* $Interact = 1$ *means there exists an additional interact relation between* e_u *and* e_i*. Based on the item-entity alignment set, the user-item graph can be seamlessly integrated with knowledge graph as a unified graph. As illustrated in Figure 3,* e_{i1}*,* e_{i2} *and* e_{i3} *appear in both knowledge graph and user-item bipartite graph, and the alignment of CKG depends on them.*

Task description. We now formulate the multi-modal KGs based recommendation task to be addressed in this paper:

- **Input** Collaborative knowledge graph that includes the user-item bipartite graph and multi-modal knowledge graph.
- **Output** A prediction function that predicts the probability of a user adopting an item.

4 METHOD

In this section, we present the MKGAT model proposed in this paper. The framework overview of MKGAT model is illustrated in Figure 5, which consists of two main sub-modules: **multi-modal knowledge graph embedding** module and **recommendation** module.

Before discussing the sub-modules, we first introduce two key components, **multi-modal knowledge graph entity encoder** and **multi-modal knowledge graph attention layer**, which serve as the basic building blocks for both KG embedding module and recommendation module.

- **Multi-modal knowledge graph entity encoder**, which use different encoder to embed each specific data type.
- **Multi-modal knowledge graph attention layer**, which aggregates the neighbor entity information of each entity to each entity itself to learn a new entity embedding.

Now we present the two sub-modules in MKGAT.

- **Multi-modal Knowledge Graph Embedding Module**

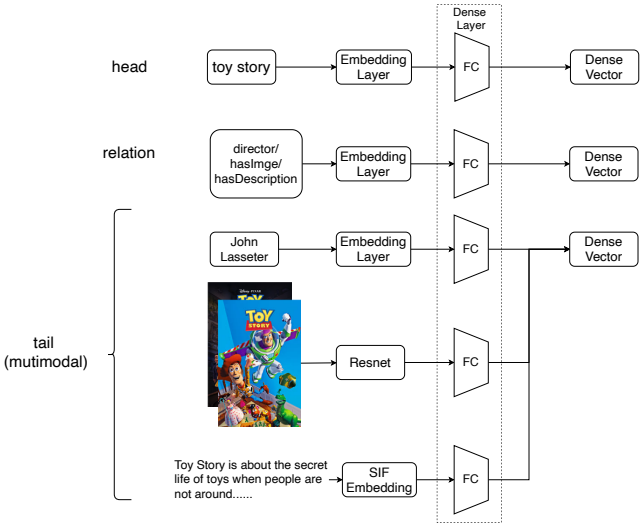

Figure 4: Multi-modal knowledge graph encoder.

Taking a collaborative knowledge graph as input, the knowledge graph embedding module utilizes the Multi-modal Knowledge Graph (MKGs) entity encoder and MKGs attention layer to learn a new entity representation for each entity. The new entity representation aggregates information of their neighbors while retaining information about itself. Then the new entity representations can be used to learn the knowledge graph embedding in order to represent the knowledge reasoning relation.

- **Recommendation Module**
  Taking knowledge graph embedding of entities (obtained by the knowledge graph embedding module) and a collaborative knowledge graph as input, the recommendation module also employ the MKGs entity encoder and MKGs attention layer to leverage corresponding neighbors to enrich the representation of users and items. Finally, the matching scores between users and items can be generated following traditional recommendation models.

In the following, we will elaborate the knowledge graph embedding module and recommendation module.

4.1 Multi-modal Knowledge Graph Embedding

In this section, we first introduce MKGs entity encoder and MKGs attention layer, and then introduce the training process of knowledge graph embedding.

4.1.1 Multi-modal Knowledge Graph Entity Encoder. To incorporate multi-modal entities into the models, we propose to learn embeddings for different modal data as well. We utilize recent advances in deep learning to construct encoders for these entities to represent them, essentially providing an embedding for all entities. Here we describe the encoders we use for multi-modal data. As Figure 4 shows, we use different encoders to embed each specific data type.

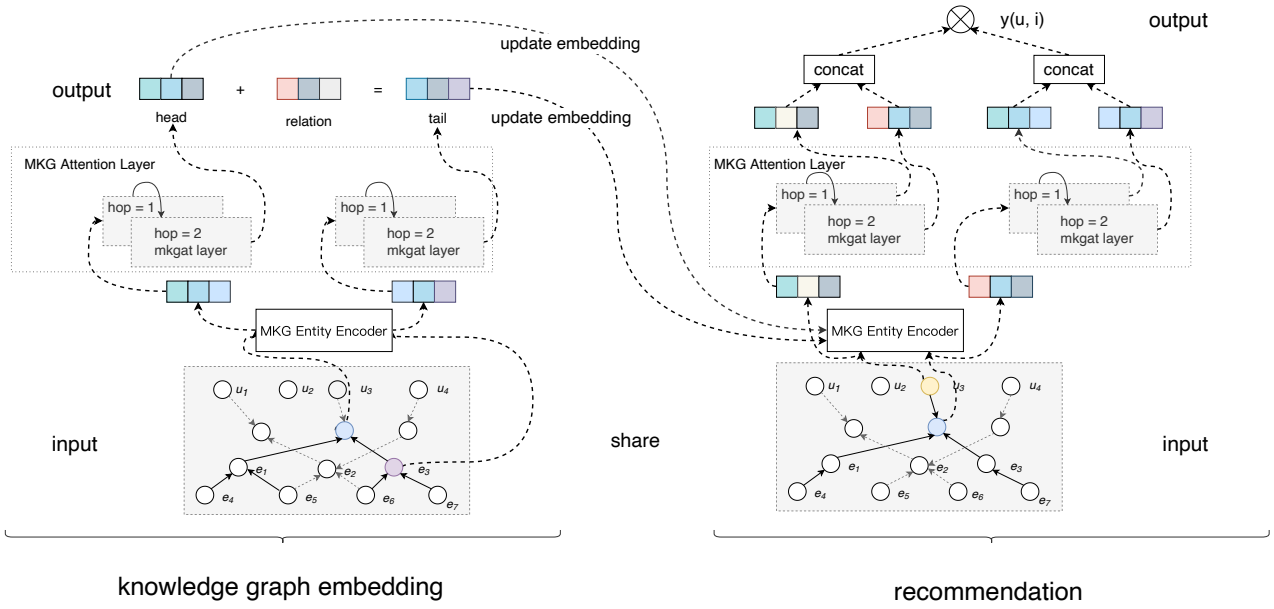

Figure 5: Framework overview of the MKGAT model.

Structured knowledge. Consider a triplet of information in the form of (h, r, t). To represent head entity h, tail entity t and relation r as independent embedding vectors, we pass their entity *id* or relation *id* through an embedding layer to generate dense vectors.
Images. A variety of models have been developed to compactly represent the semantic information in the images, and have been successfully applied to tasks such as image classification [8], and question-answering [32]. In order to embed images to make the encoding represent such semantic information, we use the last hidden layer of ResNet50 [6], which is pretrained by Imagenet [3].
Texts. These textual information is highly related to the content, and can capture users' preferences. For text entities, we use Word2Vec [16] to train word vectors, and then apply Smooth Inverse Frequency (SIF) model [1] to obtain the weighted average of the word vectors of a sentence, which is used as the sentence vector to represent the textual features. For the efficiency of the model, we use the sentence vector technique instead of using the LSTM to encode sentences. And SIF will have better performance than simply using the average of word vectors.

Finally, as illustrated in Figure 4, we use dense layers to unify all modal of entities into the same dimension, so that we can train it on our model.

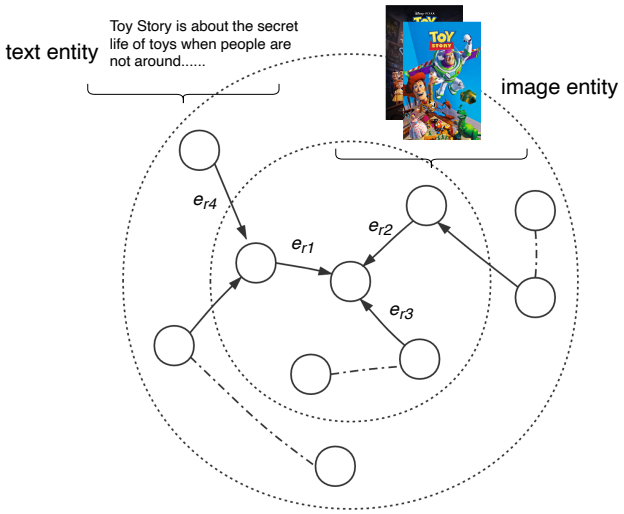

Figure 6: A 2-hop propagation in MKGs attention layer, the calculation of propagation probability takes into account the relation of knowledge graph.

4.1.2 Multi-modal Knowledge Graph Attention Layer. The MKGs attention layer illustrated in Figure 6, which recursively propagate embeddings along high order connectivity [10]. Moreover, by exploiting the idea of graph attention network (GATs) [22], we generate attentive weights of cascaded propagations to reveal the importance of such connectivity. Despite the success of GATs, they are unsuitable for KGs as they ignore relation of KGs. So we modify the GATs to take into account the embedding of the KGs relation. And the introduction of attention mechanism [21] can reduce the influence of noise and make the model focus on useful information.

Here we start by describing a single layer, which consists of two components: propagation layer and aggregation layer, and then discuss how to generalize it to multiple layers. The multi-modal knowledge graph attention layer is not only used in knowledge graph embedding but also used in recommendation.

Propagation layer. Given a candidate entity h, we must consider two aspects when learning its knowledge graph embedding. Firstly, we learn the structured representation of knowledge graph through the transE model, i.e., $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$. Secondly, for entity h's multi-modal neighbor entities, we want to aggregate this information to entity h to enrich the representation of entity h. Following the way in [28], we use $\mathcal{N}_h$ to denote the set of triplets directly connected to h. $\mathbf{e}_{agg}$ is a representation vector that aggregates neighbor entities information, which is the linear combination of each triple representation and can be calculated in Equation 1.

$$\mathbf{e}_{agg} = \sum_{(h,r,t)\in\mathcal{N}_h} \pi(h,r,t)\mathbf{e}(h,r,t), \tag{1}$$

where $\mathbf{e}(h,r,t)$ is the embedding of each triplet (h,r,t) and $\pi(h,r,t)$ is the attention score on each triplet $\mathbf{e}(h,r,t)$. $\pi(h,r,t)$ controls how much information being propagated from triplets $\mathbf{e}(h,r,t)$.

Since relation is important in the knowledge graph, we keep the embedding of relation in $\mathbf{e}(h,r,t)$ and $\pi(h,r,t)$, and the parameters in them are learnable. For triplet $\mathbf{e}(h,r,t)$, we learn this embedding by performing a linear transformation over the concatenation of the embedding of head entity, tail entity and relation, which is formulated as follows:

$$\mathbf{e}(h,r,t) = \mathbf{W}_1(\mathbf{e}_h\|\mathbf{e}_r\|\mathbf{e}_t). \tag{2}$$

where e_h and e_t are embedding of entities, and e_r is embedding of relation. We implement $\pi(h,r,t)$ via relational attention mechanism, which can be computed in the following:

$$\pi(h,r,t) = \text{LeakyReLU}\left(\mathbf{W}_2\mathbf{e}(h,r,t)\right), \tag{3}$$

where we follow the way in [22] to select LeakyReLU [15] as the nonlinear activation function. Hereafter, we normalize the coefficients across all the triplets connected with h by adopting the softmax function:

$$\pi(h,r,t) = \frac{\exp(\pi(h,r,t))}{\sum_{(h,r',t')\in\mathcal{N}_h}\exp\left(\pi\left(h,r',t'\right)\right)}. \tag{4}$$

Aggregation layer. This phase is to aggregate the entity representation $\mathbf{e}_h$ and the corresponding $\mathbf{e}_{agg}$ as the new representation of entity h in order not to lose the initial $\mathbf{e}_h$ information. In this work, we implement the aggregation function $f\left(\mathbf{e}_h, \mathbf{e}_{agg}\right)$ via the following two methods.

1) *Add aggregation method* considers the element-wise add feature interaction between $\mathbf{e}_h$ and $\mathbf{e}_{agg}$, which can be obtained by Equation 5.

$$f_{\text{add}} = \mathbf{W}_3\mathbf{e}_h + \mathbf{e}_{agg}, \tag{5}$$

where we perform a linear transformation on the initial $\mathbf{e}_h$ and add it to the $\mathbf{e}_{agg}$. $\mathbf{W}_3$ is a weight matrix that transfers the current representations into the common space, which denotes the trainable model parameters. This operation is similar to that in the residual network [6].

2) *Concatenation aggregation method* concatenates the $\mathbf{e}_h$ and $\mathbf{e}_{agg}$, using a linear transformation:

$$f_{\text{concat}} = \mathbf{W}_4\left(\mathbf{e}_h\|\mathbf{e}_{agg}\right), \tag{6}$$

where $\|$ is the concatenation operation, and $\mathbf{W}_4$ is the trainable model parameters.

High-order propagation. By stacking more propagation and aggregation layers, we explore the higher-order connectivity inherent in the collaborative knowledge graphs. In general, for a n-layer model, the incoming information is accumulated over a n-hop neighborhood.

4.1.3 Knowledge Graph Embedding. We learn a new entity representation for each entity after passing it through the MKGs entity encoder and MKGs attention layer. And then, we input these new entity representations to knowledge graph embedding, which is an effective way to parameterize entities and relations as vector representations, while preserving the reasoning of relation in knowledge graph structure.

More specifically, we employ the translational scoring function [2], a widely used method in knowledge graph embedding, to train knowledge graph embedding. It learns to embed each entity and relation by optimizing the translation principle $\mathbf{e}_h + \mathbf{e}_r \approx \mathbf{e}_t$ when a triplet (h,r,t) is valid, in which $\mathbf{e}_h$ and $\mathbf{e}_t$ are the new entity embeddings from MKGs attention layer, $\mathbf{e}_r$ is the embeddings of relation. Equation 7 depicts the score of triplet (h,r,t).

$$score(h,r,t) = \|\mathbf{e}_h + \mathbf{e}_r - \mathbf{e}_t\|_2^2. \tag{7}$$

The training of knowledge graph embedding considers the relative order between valid triplets and broken ones, and encourages their discrimination through a pairwise ranking loss:

$$\mathcal{L}_{\text{KG}} = \sum_{(h,r,t,t')\in\mathcal{T}} -\ln\sigma\left(score\left(h,r,t'\right) - score(h,r,t)\right), \tag{8}$$

where $\mathcal{T} = \{(h,r,t,t')\,|(h,r,t)\in\mathcal{G}, (h,r,t')\notin\mathcal{G}\}$, and $(h,r,t'$ is a broken triplet constructed by replacing one entity in a valid triplet randomly. $\sigma(\cdot)$ is the sigmoid function. This layer models the entities and relations on the granularity of triples, working as a regularizer and injecting the direct connections into representations, which can increase the model representation ability.

4.2 Recommendation

After each entity gets its corresponding embedding by the knowledge graph embedding module, it will be input to the recommendation module. Similar to the knowledge graph embedding module, the recommendation module also uses MKGs attention layer to aggregate neighbor entity information.

In order to retain the 1-n hop information, we follow the setup from [28] that retains the output of the candidate user and item from the l-th layer. The output of different layers represents the information of different hops. We hence adopt the layer-aggregation mechanism[31] to concatenate the representations at each step into a single vector, which can be found as follows:

$$\mathbf{e}_u^* = \mathbf{e}_u^{(0)}\|\cdots\|\mathbf{e}_u^{(L)}, \quad \mathbf{e}_i^* = \mathbf{e}_i^{(0)}\|\cdots\|\mathbf{e}_i^{(L)}, \tag{9}$$

where $\|$ is the concatenation operation and L is the number of the MKGs attention layers. By doing so, we can not only enrich the initial embeddings by performing the embedding propagation operations, but also allow controlling the strength of propagation by adjusting L.

Finally, we conduct inner product of user and item representations by Equation 10, so as to predict their matching score:

$$\hat{y}(u,i) = \mathbf{e}_u^{*\top} \mathbf{e}_i^* \tag{10}$$

Then, we optimize our recommendation prediction loss by using the Bayesian Personalized Ranking (BPR) loss [20]. Specifically, we assume that the observed records, which indicate more user preferences, should be assigned higher prediction scores than unobserved ones. The BPR loss can be constructed in Equation 11.

$$\mathcal{L}_{\text{recsys}} = \sum_{(u,i,j)\in O} -\ln \sigma(\hat{y}(u,i) - \hat{y}(u,j)) + \lambda \|\Theta\|_2^2, \tag{11}$$

where $O = \{(u,i,j)|(u,i) \in \mathcal{R}^+, (u,j) \in \mathcal{R}^-\}$ denotes the training set, $\mathcal{R}^+$indicates the observed interactions between user u and item j, $\mathcal{R}^-$ is the sampled unobserved interaction set, and $\sigma(\cdot)$ is the sigmoid function. And Θ is the parameters set, λ is the parameter of the L2 regularization.

We update the parameters in MKGs embedding module and recommendation module alternately. In particular, for a batch of randomly sampled (h, r, t, t'), we update the knowledge graph embeddings for all entities. Then we sample a batch of (u, i, j) randomly, retrieve their representations from knowledge graph embedding. The loss functions of the two modules are optimized alternately.

5 EXPERIMENTS

In this section, we evaluate MKGAT model using two real-world datasets from different domains. We first present our experimental settings in Section 5.1 and then discuss the major experimental results in Section 5.2. Furthermore, we also conduct detailed case study in Section 5.3.

5.1 Experimental Setup

5.1.1 Datasets. We conduct our experiments using two recommendation datasets from movie and restaurant domains. The details are as follows.

- **MovieLens** [3]**.** This dataset has been widely used for evaluating recommender systems. It consists of explicit ratings (ranging from 1 to 5) on the MovieLens website. In our experiment, we use the MovieLens-10M dataset. We transform the ratings into implicit feedback data, in which each entry is marked as 1 indicating that a user has rated the item, and 0 if unrated. The knowledge graph for MovieLens datasets comes from [26], which uses Microsoft Satori to construct the knowledge graph for this dataset. In particular, [26] first selects a subset of triples from the whole knowledge graph with a confidence level greater than 0.9. Given the sub-KG, [26] collect Satori IDs of all valid movies by matching their names with tail of triples. After having the set of item IDs, [26] match these item IDs with the head of all triples in Satori sub-KG, and select all well-matched triples as the final KG for each dataset. In order to construct the image entity of the MovieLens knowledge graph, we crawl the corresponding trailers instead of the full-length videos from Youtube [4]. We use FFmpeg [5] to extract the key frames of each trailer, and use the pre-trained ResNet50 [6] models to extract the visual features from key frames. In order to construct the text entity of the MovieLens knowledge graph, we crawl the corresponding movie descriptions from TMDB [6].
- **Dianping** [7], a Chinese life information service website, where users can search and get information of restaurants. Dianping is provided by Meituan-Dianping Group, wherein the types of positive interactions include buying, and adding to favorites. We sample negative interactions for each user. The knowledge graph for Dianping-Food is collected from Meituan Brain, an internal knowledge graph built for dining and entertainment by Meituan-Dianping Group. The types of entities include POIs (i.e., restaurants), first-level and second-level categories, business areas, and tags. In order to construct the image entities of the knowledge graph for Dianping dataset, we chose the images of the top recommended dishes of POIs. Similar to MovieLens datasets, we use the pre-trained ResNet50 [6] models to extract the visual features from the images of recommended dishes. In order to construct the text entity of the Dianping knowledge graph, we use user reviews for every POIs.

Table 1: Statistics of datasets

dataset	MovieLens	Dianping
# of users	41849	40388
# of items	4828	29969
# of interactions	1813381	624499
# of entities	65801	93798
# of relations	19	6
# of triplets	145406	635656

The statistics of the two datasets are shown in Table 1.

5.1.2 Evaluation Metrics. For each user in test set, we treat the items that user has not interacted with as negative items. Then each method outputs the user's preference scores over all items, except positive ones in training set. We randomly select 20% of the interactions as ground truth for testing and the remaining interactions for training. To evaluate the effectiveness of top-k recommendation and preference ranking, we adopt two widely-used evaluation metrics: $recall@k$ and $ndcg@k$. The default value of k is 20. We report the average results for all users in the test set.

5.1.3 Baselines. We compare our proposed MKGAT model with some state-of-the-art baselines, which include FM-based Method (NFM), KG-based method (CKE, KGAT), Multi-modal method (MMGCN).

- **NFM:** Neural Factorization Machines (NFM) [7]is a state-of-the-art Factorization Machines (FM), which subsumes FM under neural network. Specially, we employ one hidden layer on input features as suggested in [7].

[3] https://grouplens.org/datasets/movielens/
[4] https://www.youtube.com/
[5] http://ffmpeg.org/
[6] https://www.themoviedb.org/
[7] https://www.dianping.com/

Table 2: Overall performance of recommendation

Models	MovieLens		Dianping	
	recall	ndcg	recall	ndcg
NFM	0.3591	0.4698	0.1163	0.0724
CKE	0.3600	0.4723	0.1321	0.0895
KGAT	0.3778	0.4827	0.1522	0.1301
MMGCN	0.3966	0.5023	0.1424	0.1255
MKGAT	**0.4134**	**0.5181**	**0.1646**	**0.1433**
%Improv.	4.2%	3.1%	8.1%	10.1%

- **CKE:** Collaborative Knowledge Base Embedding (CKE) [35] combines Collaborative Filtering (CF) with structural knowledge, textual knowledge, and visual knowledge in a unified framework for recommendation. We implement CKE as CF plus structural knowledge module in this paper.
- **KGAT:** Knowledge Graph Attention Network (KGAT) [28] first applies the TransR model [13] to obtain the initial representations for entities. Then, it runs the entity propagation from the entity itself outwardly. In this way, both the user representation and the item representation can be enriched with the corresponding neighbor information.
- **MMGCN:** Multi-modal Graph Convolution Network (MMGCN) [29] is a state-of-the-art multi-modal model, which considers individual user-item interactions for each modal. specifically, MMGCN builds user-item bipartite graph for each modal, then uses GCN to train each bipartite graph, and finally merges the node information of different modals.

5.1.4 Parameter Settings. We use Xavier initializer [4] to initialize the model parameters, and optimize the model with Adam optimizer [9]. The mini-batch size and learning rate are searched in [1024; 5120; 10240] and [0:0001; 0:0005; 0:001] respectively. The number of MKGAT layers in the recommendation component and knowledge graph embedding component is searched in [1; 2; 3]. For visual entity, we use the 2048-dimensional features of the last hidden layer of Resnet. For text entity, we train 300-dimensional word embeddings by word2vec [16] and generate the corresponding sentence vectors using the SIF [1] algorithm. Finally, we set the dimension of the all entity as 64.

5.2 Experimental Results

We first report the performance of all the methods, and then study the effect of different factors (i.e., modalities, model depth and combination layers) on the models.

5.2.1 Performance of All the Methods. The results of all models are shown in Table 2. We can see that our proposed MKGAT model(where the modalities include structured knowledge, text and vision; the model depth of the knowledge graph part and the recommendation part is set as 2; and combination layers is set as add aggregation layer) outperforms all the baselines on both two dataset in terms of *recall* and *ndcg*. Besides, we have the following observations.

- MKGAT consistently yields the best performance on both datasets. Specifically, In particular, MKGAT improves over the strongest KG-based baseline KGAT w.r.t. ndcg@20 by **7.33%**, and **10.14%** and w.r.t. recall@20 by **9.42%**, and **8.15%** in MovieLens, Dianping, respectively. This verifies the validity of the multi-mode knowledge graph. And jointly analyzing Tables 2 and 3, in the situation of introducing multi-modal entities, our method can achieve a greater improvement compared to other KG-based methods. This verifies that our method is more friendly to multi-modal information than other methods.
- Among all the comparison methods, KG-based methods (i.e., CKE and KGAT) outperform the plain CF-based method (i.e., NFM) on the two datasets across two evaluation metrics, which demonstrates that the usage of KGs indeed greatly improves the recommendation performance.
- Comparing the performance of the two KG-based methods, CKE and KGAT, we find that KGAT has a better performance than CKE in both metrics, which demonstrates the power of graph convolutional networks in recommender systems.
- It's worth mentioning that MKGAT can beat MMGCN model on MovieLens data, which is a state-of-the-art multi-modal recommendation method. This shows that our method can make rational use of multi-modal information.

Table 3: Performance of recommendation: effect of modalities on Dianping dataset

Models	KGAT		MKGAT	
	recall	ndcg	recall	ndcg
base	0.1522	0.1301	0.1542	0.1341
base + text	0.1544	0.1343	0.1589	0.1389
%Improv.	1.5%	3.2%	**3.1%**	**3.5%**
base + image	0.1572	0.1352	0.1612	0.1396
%Improv.	3.3%	3.9%	**4.5%**	**4.1%**
base + text + image	0.1598	0.1361	0.1646	0.1433
%Improv.	4.9%	4.6%	**6.7%**	**6.8%**

5.2.2 Effects of Modalities. To explore the effects of different modalities, we compare results of the KGAT and our MKGAT model on different modalities over the Dianping dataset. The performance comparison results are presented in Table 3. We have the following observations:

- As expected, the methods with multi-modal features outperforms those with single-modal features in both KGAT and MKGAT, as shown in Table 3.
- The visual modality plays a more important part than the text modality in recommendation effectiveness due to the fact that the visual information (such as pictures) tends to attract a user's attention when he/she views the restaurant information on an online platform.
- Our MKGAT model, also as a KG-based method, can take advantage of image information to improve the recommendation performance better compared to KGAT. In other words, compared with other KG-based methods, our method will have a greater improvement when multi-modal information is introduced. The reason behind it is that when we train the knowledge graph embeddings, MKGAT can aggregate the information of image entities into item entities better, as shown in Table 3.

Table 4: Performance of recommendation: effect of model depth

Models		MovieLens		Dianping	
		recall	ndcg	recall	ndcg
KGE	one layer	0.4113	0.5169	0.1632	0.1424
	two layer	0.4134	0.5181	**0.1646**	**0.1433**
	three layer	**0.4149**	**0.5192**	0.1604	0.1413
REC	one layer	0.4082	0.5152	0.1546	0.1365
	two layer	**0.4134**	**0.5181**	**0.1646**	**0.1433**
	three layer	0.4104	0.5147	0.1628	0.1420

5.2.3 Effect of Model Depth. To evaluate the effectiveness of layers stack, we conduct experiments on different numbers of layers. The number of layers is regarded as the model depth. In our model, both knowledge graph embedding and recommendation components use the MKGAT layers, so we discuss knowledge graph embedding component and recommendation component separately. When it comes to the knowledge graph embedding part, we fix the number of the MKGAT layers of recommendation to 2. And when discussing the recommendation part, we fix the number of the MKGAT layers of knowledge graph embedding to 2. The experimental results are shown in Table 4.

The effect of different model depth (i.e., different numbers of MKGAT layers) in the knowledge graph embedding (marked as KGE) and recommendation (marked as REC) can be summarized in the following.

- For knowledge graph embedding, in the MovieLens dataset, as the number of MKGAT layers increases, the evaluation metrics (i.e., recall and ndcg) also increases. It demonstrates that the effectiveness of neighborhood information fusion in knowledge graph embedding. In the Dianping dataset, as the number of MKGAT layers increasing, the evaluation metrics grow first and then decreases. This may be caused by the multi-hop information of Dianping data is relatively sparse. Combined with the results in Table 3, we can see that our method (that takes into account the information of neighbor entities when doing knowledge graph embedding) can provide higher quality entities embeddings for recommendation, compared with those methods (e.g., KGAT) that consider knowledge graph entity triplets independently.
- When referring to the recommendation part, with the number of MKGAT layers increasing, the evaluation metrics grow first in two datasets, which verifies that knowledge graph embeddings of different hops are helpful for the recommendation system. However, when the number of layers in the two datasets is greater than 2, the evaluation metrics will decline. In other words, when the number of layers increases to a certain level, the evaluation metrics decline. This may be caused by overfitting due to the sparsity of data.

5.2.4 Effect of Combination Layers. In this set of experiments, we study the effect of combination layers in our model. Specifically, we use two types of aggregation layers, i.e., add layers and concatenate layers, to learning the knowledge graph embeddings. The model depth is fixed as 2. Table 5 summarizes the experimental results,

Table 5: Performance of recommendation: effect of combination layers

combine method	MovieLens		Dianping	
	recall	ndcg	recall	ndcg
ADD	0.4134	0.5181	0.1646	0.1433
CONCAT	**0.4162**	**0.5209**	**0.1657**	**0.1452**

which depicts the method with concatenate layers (marked by CONCAT) is superior to that with add layers (marked by ADD). One possible reason is that the neighbor entities of each entity contain textual and visual information, which are heterogeneous to the general entities in the knowledge graph. They are not in the same semantic space. In fact, ADD is an element-by-element feature interaction method, which is suitable for features in the same semantic space. Because in the same semantic space, the meaning of each dimension of each feature is the same, it makes sense to add each dimension of each feature. However, CONCAT is an extension of the dimension between features, which is more suitable for the interaction of features in different semantic spaces.

5.3 Case study

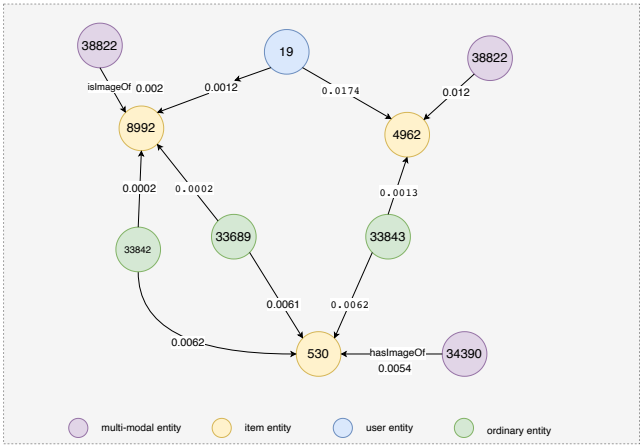

Figure 7: Real Example from dianping datasets, we use different colored dots to represent different types of entities.

To intuitively demonstrate the role of multi-modal entities in the MKGAT model, we give a case study by randomly selecting a user u from the Dianping dataset, and a relevant item. Benefiting from the attention mechanism, we can calculate the relevance score (unnormalized) between the candidate items and the entity (or items and users). We can also observe the relevance scores between each entity and other entities. The higher the relevance score is, the model believes that the current entity has a greater effect on the model. We visualize the relevance score in Figure 7.

In Figure 7, for item entities (i.e., item entity 8992, 4962 and 530), their neighboring entities include multi-modal entities and non-multimodal entities (interactions or ordinary KG entities). We visualize the edge weight of each item entity and its neighboring entities, as shown in Figure 7. The multi-modal relation usually has

a relatively high relevance score in collaborative knowledge graph, indicating the importance of multi-modal entities.

6 CONCLUSION AND FUTURE WORK

In this paper, we present a novel KG-based model for recommendation, called Multi-modal Knowledge Graph Attention Network (MKGAT), which introduces the multi-modal knowledge graph to the recommendation system innovatively. By learning the reasoning relationship among entities and aggregating the neighbor entity information of each entity to itself, the MKGAT model can make better use of the multi-modal entity information. Extensive experiments on two real-world datasets demonstrate the rationality and effectiveness of our proposed MKGAT model.

This work represents an initial attempt to explore the use of multi-modal knowledge graphs in recommendation systems, based on which further interesting research can be conducted. For example, it is natural to explore more ways of multi-modal fusion under the framework of multi-modal knowledge graph, such as Tensor Fusion Network (TFN) [34] or Low-rank Multimodal Fusion (LMF) [14].

7 ACKNOWLEDGMENTS

This work is partially supported by NSFC (No. 61972069, 61836007, 61832017, 61532018).

REFERENCES

[1] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016. A simple but tough-to-beat baseline for sentence embeddings. (2016).
[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*. 2787–2795.
[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.
[4] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 249–256.
[5] Maoxiang Hao, Zhixu Li, Yan Zhao, and Kai Zheng. 2018. Mining High-Quality Fine-Grained Type Information from Chinese Online Encyclopedias. In *International Conference on Web Information Systems Engineering*. 345–360.
[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[7] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 355–364.
[8] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3128–3137.
[9] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[10] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[11] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
[12] Qianyu Li, Xiaoli Tang, Tengyun Wang, Haizhi Yang, and Hengjie Song. 2019. Unifying task-oriented knowledge graph learning and recommendation. *IEEE Access* 7 (2019), 115816–115828.
[13] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.
[14] Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. 2018. Efficient low-rank multimodal fusion with modality-specific factors. *arXiv preprint arXiv:1806.00064* (2018).
[15] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. [n.d.]. Rectifier nonlinearities improve neural network acoustic models.
[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[17] Hatem Mousselly-Sergieh, Teresa Botschen, Iryna Gurevych, and Stefan Roth. 2018. A multimodal translation-based approach for knowledge graph representation learning. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*. 225–234.
[18] Tu Dinh Nguyen, Dat Quoc Nguyen, Dinh Phung, et al. 2018. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. 327–333.
[19] Pouya Pezeshkpour, Liyan Chen, and Sameer Singh. 2018. Embedding Multimodal Relational Data for Knowledge Base Completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 3208–3218.
[20] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
[22] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
[23] Hongwei Wang, Fuzheng Zhang, Min Hou, Xing Xie, Minyi Guo, and Qi Liu. 2018. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 592–600.
[24] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 417–426.
[25] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference*. 1835–1844.
[26] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 968–977.
[27] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.
[28] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 950–958.
[29] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal Graph Convolution Network for Personalized Recommendation of Micro-video. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1437–1445.
[30] Ruobing Xie, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Image-embodied knowledge representation learning. *arXiv preprint arXiv:1609.07028* (2016).
[31] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. *arXiv preprint arXiv:1806.03536* (2018).
[32] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 21–29.
[33] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*. 283–292.
[34] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2017. Tensor fusion network for multimodal sentiment analysis. *arXiv preprint arXiv:1707.07250* (2017).
[35] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 353–362.
[36] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. 2017. Meta-graph based recommendation fusion over heterogeneous information networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 635–644.