A SURVEY OF DEEP LEARNING BASED RECOMMENDER SYSTEM

Wencao Yang
Strike Exchange LLC
yangwencao@gmail.com

ABSTRACT

Recommender systems are widely utilized in domains such as content consumption and product recommendations. In recent years, deep learning has achieved remarkable success in fields like natural language processing and computer vision, leading to its application in recommender systems. This survey reviews recent advancements in deep learning-based recommender systems, highlighting their development and the common deep learning techniques employed in the field. Additionally, we present and compare the architectures of various models to provide a comprehensive understanding of their capabilities and differences.

***K*eywords** Recommender System · Neural Network · Collaborative Filtering · Embedding · Knowledge Graph

1 Introduction

The rapid growth of information on the Internet and the abundance of choices have made recommender systems indispensable for retrieving and ranking items tailored to individual users. These systems are widely applied in domains such as content consumption (e.g., movies, music, news) and e-commerce. By leveraging user-item interactions and auxiliary information about users or items, they effectively capture short- and long-term user interests to generate predictions and recommendations.

Recent advancements in machine learning, particularly deep learning, have significantly improved recommender systems, addressing challenges such as accuracy, sparsity, cold-start issues, and scalability. Platforms like YouTube [1], Netflix [2], eBay [3], and TikTok [4] have successfully implemented advanced algorithms to enhance their recommendation quality.

Despite these advancements, comprehensive reviews of deep learning-based recommender systems were scarce until the in-depth works of S. Zhang et al. [5] and Z. Batmaz et al. [6]. While brief surveys, such as [7, 8], exist to aid beginners, they often overlook specialized topics like knowledge graphs, sequence embeddings, and attention-based models. This survey seeks to address these gaps by focusing on significant progress and recent developments, particularly in the area of knowledge graphs, with the aim of offering valuable insights into modern recommender systems.

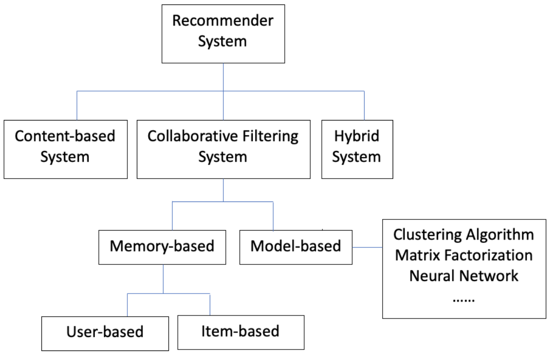
This survey is organized as follows: Section 1 provides a high-level introduction to recommender systems. Section 2 reviews their recent advancements. Section 3 outlines key deep learning techniques and terminologies relevant to recommender systems. Section 4 discusses the integration and application of knowledge graphs. Section 5 compares various model architectures. Finally, Section 6 concludes the survey with insights and future research directions.

As illustrated in Figure 1(a), recommender systems are generally categorized into three types [9]: collaborative filtering (CF) systems, content-based systems, and hybrid systems.

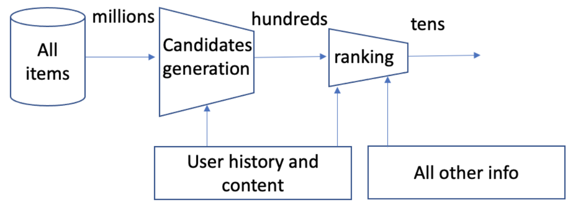
Collaborative filtering systems rely on user-item interactions to identify related or similar users and items. These systems are further divided into:

Memory-based methods, which determine similarities directly from the user-item matrix to find similar users (user-based) or items (item-based). However, these methods often struggle with sparse data. Model-based methods, which use machine learning techniques such as matrix factorization or neural networks to predict ratings for new items, offering improved scalability and handling of sparsity. Content-based systems focus on the intrinsic features of items, leveraging side information such as textual descriptions, images, or audio to make recommendations based on item similarity.

Hybrid systems combine the strengths of collaborative filtering and content-based approaches to overcome their individual limitations, often achieving superior performance.



(a) Classification of recommender systems



(b) Structure of a recommender system

Figure 1: Summary of recommender systems

Figure 1(b) depicts the general structure of a recommender system, which is typically divided into two stages:

Retrieval stage, where a large pool of items is filtered down to a manageable set of relevant candidates. Ranking stage, where these candidates are further ordered based on their predicted relevance to the user. This two-stage design ensures scalability and efficiency, enabling recommender systems to handle the vast number of available items effectively.

2 Development of Recommender System

Content-based filtering recommends items by analyzing their features to find those similar to the ones a user has previously interacted with, whether explicitly or implicitly. In contrast, collaborative filtering simultaneously leverages similarities between users and items. For example, to predict a user's rating for an item, collaborative filtering identifies other users with similar preferences and uses their ratings to estimate the target user's preference.

The Netflix Prize competition [2], launched in October 2006, significantly boosted interest in matrix factorization (MF) [10] due to its superior performance compared to traditional nearest-neighbor techniques. MF is a foundational embedding-based approach that decomposes the user-item interaction matrix into two low-dimensional matrices: a user embedding matrix and an item embedding matrix. Figure 2 illustrates an example where the user-item interaction matrix of size $m \times n$ is decomposed into:

$$U \in \mathbb{R}^{m\times d} \quad (user embedding matrix),$$

$$V \in \mathbb{R}^{n\times d} \quad (item embedding matrix),$$

where d is the embedding dimension ($d = 2$ in this example).

Factorization Machines (FM) [11], a generalization of linear regression and MF, extend these models by capturing both linear interactions and pairwise feature interactions. FM models the prediction $\hat{y}$ for a feature vector $\mathbf{x}$ as:

$$\hat{y} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j,$$

where w_0 is the global bias, w_i is the weight for feature i, and $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \mathbf{v}_i^\top \mathbf{v}_j$ represents the dot product of feature embeddings $\mathbf{v}_i$ and $\mathbf{v}_j$.

Additionally, neural collaborative filtering extends the capacity of traditional models by utilizing neural networks to learn complex, non-linear interactions between user and item embeddings, significantly enhancing predictive accuracy and flexibility.

Compared to traditional matrix factorization, which decomposes the user-item interaction matrix into latent vectors for users and items, neural collaborative filtering models leverage neural networks to construct these latent representations. The commonly used neural networks include:

- Multiple Layer Perceptron (MLP),

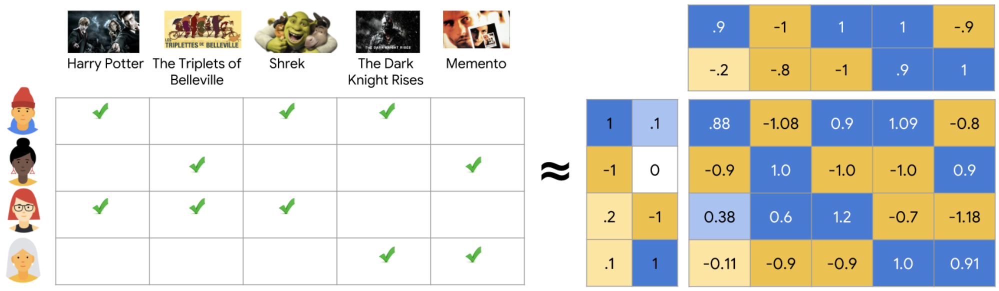


Figure 2: An example of MF. Image taken from [12]

- Convolutional Neural Network (CNN),
- Recurrent Neural Network (RNN),
- Long Short-Term Memory (LSTM),
- Gated Recurrent Unit (GRU).

As illustrated in Figure 3, numerous models have been proposed to improve recommendation performance, including:

- **Deep Crossing** [13],
- **Factorization-Machine Supported Neural Networks (FNN)** [14],
- **Product-based Neural Network (PNN)** [15],
- **Wide & Deep Model** [16],
- **DeepFM** [17],
- **Deep & Cross Network (DCN)** [18],
- **Attentional Factorization Machines (AFM)** [19],
- **Deep Interest Network (DIN)** [20],
- **Deep Interest Evolution Network (DIEN)** [21].

These models have introduced significant advancements in recommendation tasks by capturing complex interactions and enhancing predictive accuracy. Further details and comparisons of these architectures can be found in Section 5.

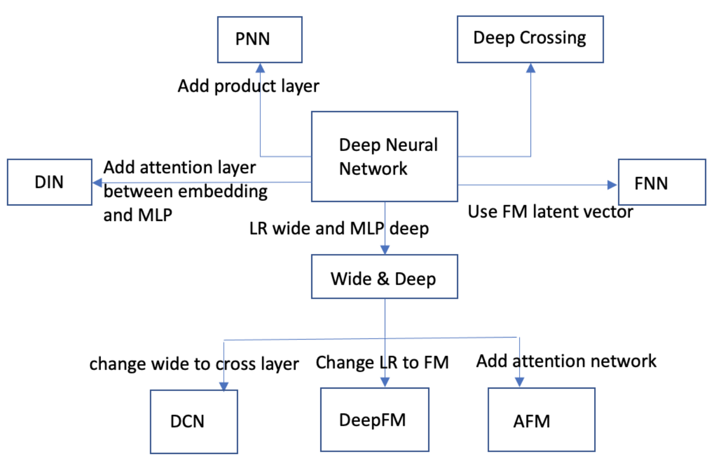


Figure 3: An overview of deep learning models.

deep neural networks for youtube recommendations [1] is a seminal paper that demonstrates the use of neural networks in video retrieval and ranking in real-world applications. In this work, video retrieval is modeled as an extreme multiclass classification problem.

As illustrated in Figure 4, embeddings for video watches and search tokens are averaged and concatenated with other demographic embeddings, which are then passed through a fully connected neural network. A similar neural network architecture is employed for the ranking stage.

The primary distinction between the retrieval and ranking networks lies in the input features. The ranking model incorporates significantly more features, such as:

- *Time since last watch*,
- *Number of previous impressions*.

These additional features enable the ranking network to provide more precise and context-aware recommendations.

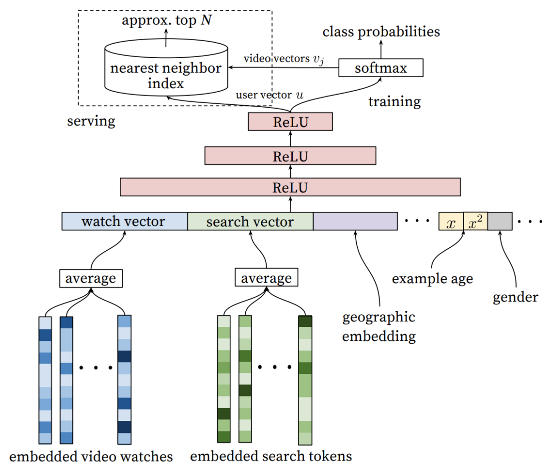


Figure 4: An overview of DNN for youtube. Image taken from [1]

The transformer has emerged as the state-of-the-art feature extractor since its introduction in the field of natural language processing (NLP). It has demonstrated exceptional performance across various tasks, making it a natural choice for leveraging attention networks in recommender systems. Numerous recommender system models incorporating attention mechanisms have been proposed, including:

- **DeepCoNN** [22],
- **TransNet** [23],
- **D-ATT** [24].

Although a comprehensive discussion of these models is beyond the scope of this survey, their development highlights the growing importance of attention mechanisms in recommendation tasks.

Recently, significant efforts have been directed towards integrating knowledge graphs into recommender systems, as they enhance recommendations by providing rich, contextual side information. Two prominent approaches have been developed to utilize knowledge graphs in recommendation tasks:

- **Embedding-based methods**,
- **Path-based methods**.

A detailed discussion of these approaches can be found in Section 4.

3 Techniques of Deep Learning

This section outlines fundamental deep learning techniques and their applications in recommender systems.

3.1 Autoencoder

An autoencoder is a type of neural network designed to reconstruct its input data at the output layer. In recommendation systems, autoencoders are often used to learn low-dimensional representations (embeddings) of features in the hidden layer or to predict missing values in the reconstruction layer. Examples include **AutoRec** [25] and **CDAE** [26].

3.2 Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) [27, 28] are a class of neural networks extensively used in computer vision (CV). As shown in Figure 5, a typical CNN consists of convolutional layers, pooling (subsampling) layers, and fully connected layers. CNNs are highly effective at capturing global and local information, making them ideal for extracting features from images and text data.

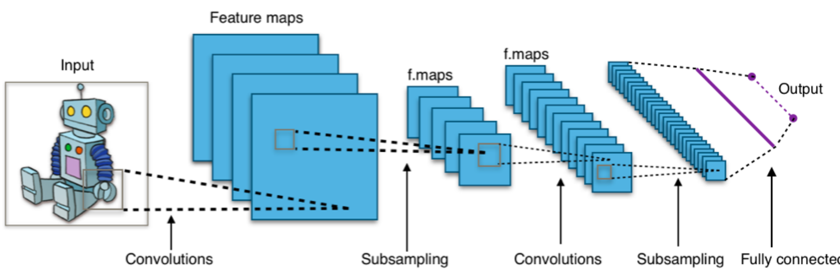


Figure 5: A typical CNN architecture. Image taken from [29]

CNNs have been applied in:

- Fashion trend analysis [30, 31],
- Style preference modeling [32],
- Point-of-interest (POI) recommendations using images [33],
- Document encoding for news recommendations [34].

ConvMF [35] integrates CNNs into probabilistic matrix factorization (PMF), generating latent vectors for documents. J. McAuley et al. [36] utilized CNNs to uncover relationships between the appearances of object pairs. Hybrid models, such as attention-based CNNs [37], have also been proposed for tasks like review rating prediction.

3.3 Recurrent Neural Network (RNN)

Recurrent neural networks (RNNs) [38] process sequential data by maintaining hidden states over time. As shown in Figure 6, the hidden layer h_t at time t depends on both the input vector x_t and the hidden state h_{t-1} from the previous time step. The computation is expressed as:

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h), \tag{1}$$

where σ_h is the activation function, and W_h, U_h, and b_h are parameter matrices and biases.

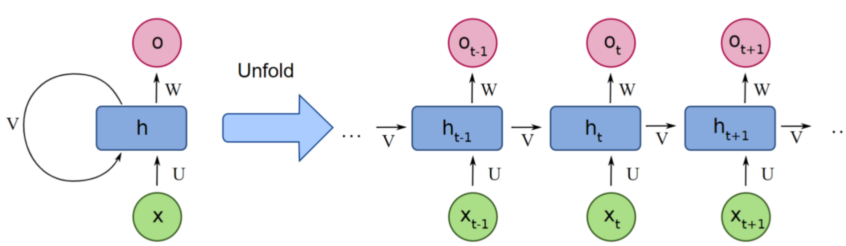


Figure 6: An RNN structure unfolded over time. Image taken from [39]

Variants like **Long Short-Term Memory (LSTM)** [40] and **Gated Recurrent Unit (GRU)** [41] address the gradient vanishing problem in RNNs.

Applications of RNNs in recommender systems include:

- Deep coevolutionary networks (**DeepCoevolve**) for time-dependent user-item interactions [42],
- Bi-directional RNN and GRU for text recommendations [43],
- Sequential recommendations using GRU [44],
- Session-based recommendations [45].

3.4 Sequence Embedding

Sequence embedding refers to generating dense representations for entities in sequential data, inspired by techniques like **word2vec** [46]. Word2vec uses a neural network to capture word associations and employs techniques such as continuous bag-of-words (CBOW) and skip-gram for training.

For recommendation systems, similar approaches, such as **item2vec** [47], are used to generate embeddings from user behavior sequences. The objective function for skip-gram in item2vec is:

$$\frac{1}{K}\sum_{i=1}^{K}\sum_{j\neq i}^{K}\log p(w_j|w_i), \tag{2}$$

where w_i and w_j represent items in the same session.

For example, Airbnb applied skip-gram models to learn embeddings for listing recommendations by analyzing uninterrupted click sessions [48].

3.5 Attention-Based Methods

The transformer architecture [49] is a powerful feature extractor that forms the backbone of NLP models such as **BERT** [50] and **GPT-3** [51]. The core of the transformer is the attention mechanism, computed as:

$$Attention(Q,K,V) = softmax\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{3}$$

where Q, K, and V are the query, key, and value matrices, respectively.

Attention-based models like **SASRec** [52] use self-attention mechanisms for sequential recommendations. As shown in Figure 7, the input consists of previous items, and the output predicts the next item.

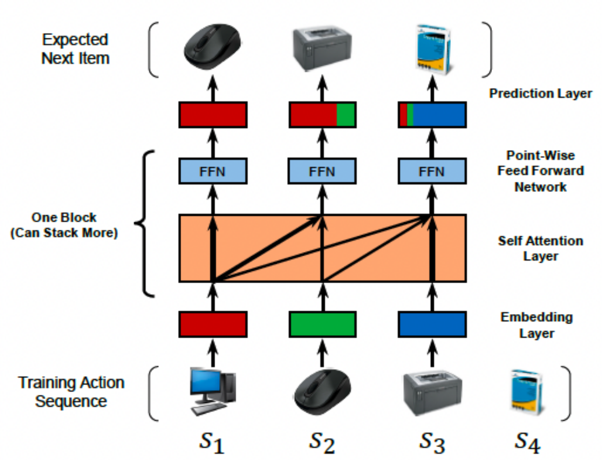


Figure 7: SASRec. Image taken from [52]

Other attention-based applications include:

- Hashtag recommendations for multimodal microblogs [53],

- Multi-Pointer Co-Attention Networks for review-based recommendations [54],
- Enhanced models such as DeepCoNN [22], TransNet [23], and D-ATT [24].

4 Knowledge Graph

A knowledge graph, sometimes referred to as a knowledge base, is a graph-structured representation of knowledge comprising a collection of interlinked entities, such as events, concepts, or products. Knowledge graph embedding methods have been developed to generate meaningful vector representations of these entities, often leveraging sequence embedding techniques.

Both **DeepWalk** [55] and **node2vec** [56] use random walks to create sequences from the knowledge graph. These sequences are then processed using sequence embedding methods to generate embeddings for the entities.

Another approach represents relationships between entities as translations in the embedding space. Relationships in knowledge graphs are typically expressed as triplets (h, r, t), where:

- h: head entity,
- t: tail entity,
- r: relationship between h and t.

Several models based on this translation principle have been proposed to embed knowledge graphs, including:

- **TransE** [57],
- **TransH** [58],
- **TransR** [59],
- **TransD** [60].

As illustrated in Figure 8, **TransR** maps entity embeddings from the entity space to the relation space. The relationship r between two entities is modeled as the difference between their projections, h_r and t_r, in the relation space.

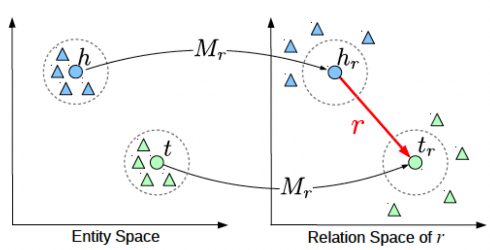


Figure 8: Simple illustration of TransR. Image taken from [59]

Knowledge graphs can enhance recommender systems in several ways:

1. **Semantic relatedness:** Knowledge graphs introduce semantic relationships among items.
2. **Relationship diversity:** Knowledge graphs encode different types of relationships, extending users' interests and increasing the diversity of recommended items.
3. **Explainability:** Knowledge graphs connect users' historical interactions to recommended items, improving the explainability of recommendations.

Various models have been proposed to integrate knowledge graphs into recommender systems. These models can be classified into three categories [61]:

1. Embedding-based methods,
2. Path-based methods,
3. Unified methods.

4.1 Embedding-based Methods

Embedding-based methods directly use embeddings derived from knowledge graphs to enrich the representations of users or items. For instance:

- The **Collaborative Knowledge Base (CKE)** model [62] employs Bayesian TransR, Bayesian stacked denoising autoencoders (SDAE), and Bayesian stacked convolutional autoencoders (SCAE) to generate structural, textual, and visual embeddings. These embeddings are combined with latent item offset vectors.
- The **Deep Knowledge-Aware Network (DKN)** [34] is used for news recommendation. It learns news title embeddings using Kim CNN [63]. Words in news titles are concatenated with corresponding entity embeddings generated by TransD and fed into the CNN. The model also incorporates an attention mechanism.

Other notable works include **entity2rec** [64], **SHINE** [65], and **RCF** [66]. These models rely on raw latent vectors learned from knowledge graph embeddings. Some recent approaches aim to refine these embeddings to enhance recommendation performance.

4.2 Path-based Methods

Path-based methods, often referred to as heterogeneous information networks (HIN), utilize the connectivity patterns within a knowledge graph. These methods learn connectivity similarities to improve recommendation quality.

For example, **PathSim** [67] is commonly used to measure similarity along predefined meta-paths. The **Personalized Entity Recommendation (PER)** model [68] defines a meta-path $\mathcal{P}$ and evaluates the score between user i and item j along $\mathcal{P}$ using PathSim.

However, PER relies on predefined meta-paths, which limits flexibility. Models like the **Knowledge-aware Path Recurrent Network (KPRN)** [69] address this limitation by automatically learning paths. KPRN processes sets of paths between user-item pairs as input and outputs scores indicating the likelihood of user interaction with the target item. As shown in Figure 9, the input embedding layer in KPRN consists of three components: entity, entity type, and relation type. These embeddings are processed by LSTM layers to predict interactions between entities (e.g., Alice, e_1, and "I See Fire," e_5).

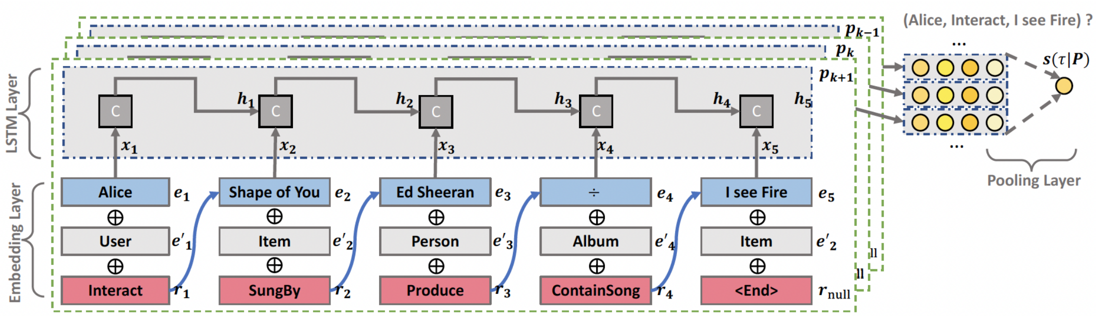


Figure 9: Schematic overview of KPRN. Image taken from [69]

Other notable models in this category include:

- **Recurrent Knowledge Graph Embedding (RKGE)** [70],
- **Meta-Graph based Recommendation Fusion over HIN** [71],
- **Meta-path-based Recommendation with a Neural Co-Attention Model** [72].

Path-based methods generally focus on extracting or learning meta-paths to enhance recommendation tasks.

4.3 Unified Methods

Unified methods combine the strengths of embedding-based and path-based approaches to leverage both direct embeddings and graph connectivity patterns.

For instance, the **RippleNet** [73] model integrates both techniques. As shown in Figure 10, RippleNet uses a preference propagation mechanism where user history serves as the seed entity set. The model iteratively expands these seed entities to explore multi-hop connections within the graph, enriching user embeddings with contextually relevant information.

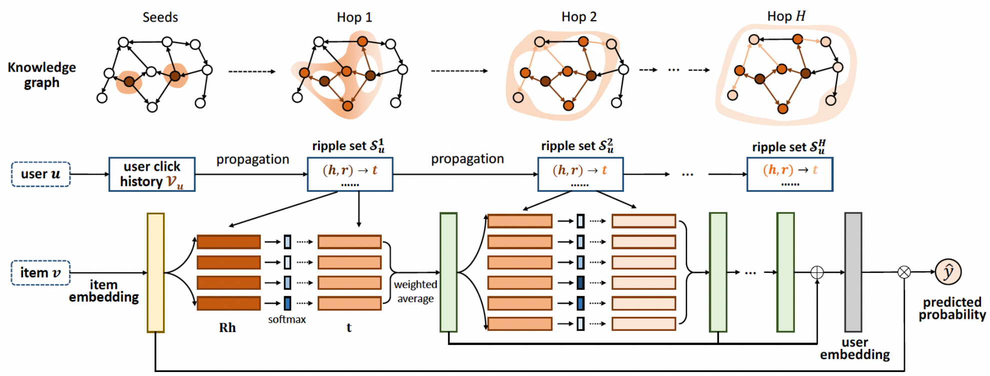


Figure 10: The overall framework of RippleNet. Image taken from [73]

RippleNet [73] is the first work to integrate both embedding-based and path-based methods for knowledge graph-aware recommendation. RippleNet derives user embeddings based on a user's interaction history, leveraging the concept of *preference propagation*.

As illustrated in Figure 10, the process begins with the user's clicked items, which serve as seed entities. Each triplet (h, r, t) represents a relationship in the knowledge graph:

- h: head entity (from seed entities),
- r: relation,
- t: tail entity (reachable from h).

The tail entities t from the first set of triplets form the 1-hop ripple set, which becomes the head entities for generating the 2-hop ripple set. This process is repeated recursively to form the k-hop ripple set from the $(k-1)$-hop ripple set.

The relevance probability P_i between an item v and a head entity h_i in the k-hop ripple set is calculated. Using these probabilities, the vector o^k is computed as a weighted sum of the tail entities t_i:

$$o^k = \sum_i P_i \cdot t_i.$$

Finally, the user embedding is constructed as the sum of all order vectors o^k across the hops:

$$UserEmbedding = \sum_k o^k.$$

This hierarchical propagation mechanism allows RippleNet to aggregate multi-hop contextual information, enhancing recommendation performance by capturing deeper relationships in the knowledge graph.

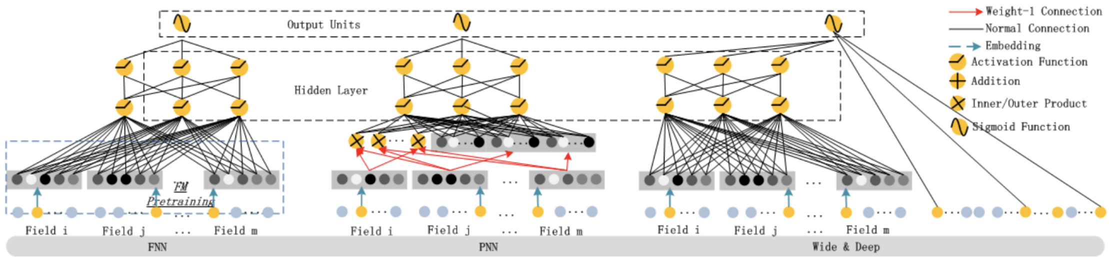
Y. Cao et al. [74] proposed a model unifying knowledge graph learning and recommendation.

5 Architectures of Deep Learning based Recommender Systems

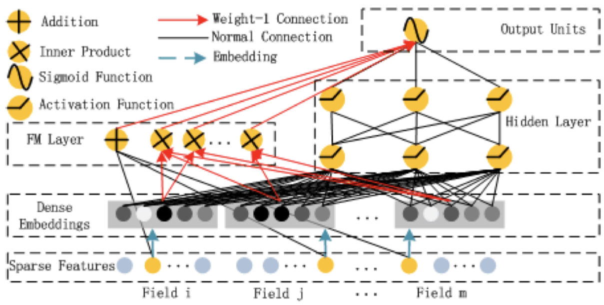
In this section, we compare some typical architectures in deep learning-based recommendation systems.

The **Deep Crossing** model, proposed by Microsoft in 2016, serves as a foundational model for deep learning recommendation systems. It converts features into dense embeddings, concatenates them into a stacking layer, and processes them through multiple residual units.

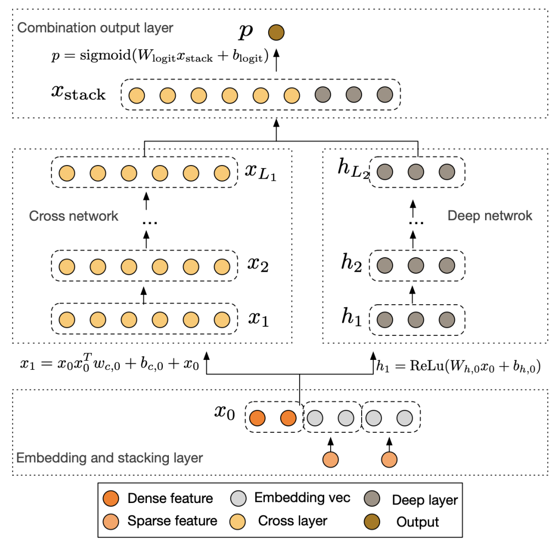
Factorization-Machine supported Neural Networks (FNN) improve upon Deep Crossing by using the latent vectors learned from Factorization Machines (FM) as initial embeddings instead of random initialization. In contrast, the **Product-based Neural Network (PNN)** introduces a product layer between the embeddings and hidden layers, where the product layer can compute either an inner product or an outer product.



(a) FNN, PNN, Deep & Wide, Image taken from [17]



(b) DeepFM. Image taken from [17]



(c) DCN, Image taken from [18]

Figure 11: Different Architectures

The **Wide & Deep model** combines a wide component for memorization and a deep component for generalization, which are fed into a final sigmoid function for prediction. Figure 11(a) illustrates the differences among FNN, PNN, and Wide & Deep models.

As shown in Figure 11(b), **DeepFM** replaces the wide component of the Wide & Deep model with an FM component. This change allows DeepFM to capture order-2 feature interactions more effectively due to the inner product units in the FM component.

Figure 11(c) demonstrates that the **Deep & Cross Network (DCN)** is similar to DeepFM, but it replaces the wide component with a cross network to enhance feature interactions. The cross layer in DCN computes feature interactions explicitly and is further detailed in [18].

Additional architectures include:

- **Attentional Factorization Machines (AFM)** [19], an FM model augmented with an attention mechanism,
- **Deep Interest Network (DIN)** [20], which applies attention mechanisms within the neural network to focus on user-specific features,
- **Deep Interest Evolution Network (DIEN)** [21], designed to capture the evolution of user interests over time.

6 Conclusion and Future Work

S. Zhang et al. [5] have identified several promising future research directions and unresolved challenges in the field of recommender systems. These include:

- *Explainable Recommendation with Deep Learning*, to enhance the transparency and interpretability of recommendations,
- *Cross-Domain Recommendation with Deep Neural Networks*, to enable effective knowledge transfer across different domains,
- *Deep Multi-Task Learning for Recommendation*, to jointly optimize related tasks for improved performance,
- *Development of Standardized Evaluation Datasets*, to ensure fair and consistent benchmarking of recommender system models.

We encourage readers to explore these areas further to address open issues and advance the field of recommender systems.

The vast and ever-growing amount of data generated on the Internet poses significant challenges for information retrieval and decision-making. Deep learning-based recommender systems (DLRS) are emerging as powerful tools to address these challenges, offering innovative solutions to improve recommendation quality and user satisfaction.

While this survey is not exhaustive, we hope it provides readers with a foundational understanding of recent advancements in deep learning-based recommender systems and inspires further exploration in this dynamic and impactful domain.

References

[1] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 191–198, New York, NY, USA, 2016. Association for Computing Machinery.

[2] James Bennett, Stan Lanning, and Netflix Netflix. The netflix prize. In *In KDD Cup and Workshop in conjunction with KDD*, 2007.

[3] Tian Wang, Yuri M. Brovman, and Sriganesh Madhvanath. Personalized embedding-based e-commerce recommendations at ebay. *CoRR*, abs/2102.06156, 2021.

[4] Xiangyu Zhao, Xudong Zheng, Xiwang Yang, Xiaobing Liu, and Jiliang Tang. Jointly learning to recommend and advertise. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3319–3327, New York, NY, USA, 2020. Association for Computing Machinery.

[5] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.*, 52(1), feb 2019.

[6] Zeynep Batmaz, Ali Yurekli, Alper Bilge, and Cihan Kaleli. A review on deep learning for recommender systems: Challenges and remedies. *Artif. Intell. Rev.*, 52(1):1–37, jun 2019.

[7] Ruihui Mu. A survey of recommender systems based on deep learning. *IEEE Access*, 6:69009–69022, 2018.

[8] Basiliyos Tilahun Betru, Charles Awono Onana, and Bernabe Batchakui. Deep learning methods on recommender system: A survey of state-of-the-art. *International Journal of Computer Applications*, 162(10):17–22, Mar 2017.

[9] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[10] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[11] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000, 2010.

[12] Matrix factorization. `https://developers.google.com/machine-learning/recommendation/collaborative/matrix`.

[13] Ying Shan, T. Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 255–262, New York, NY, USA, 2016. Association for Computing Machinery.

[14] Weinan Zhang, Tianming Du, and Jun Wang. Deep learning over multi-field categorical data: A case study on user response prediction, 2016.

[15] Yanru Qu, Bohui Fang, Weinan Zhang, Ruiming Tang, Minzhe Niu, Huifeng Guo, Yong Yu, and Xiuqiang He. Product-based neural networks for user response prediction over multi-field categorical data. *ACM Trans. Inf. Syst.*, 37(1), oct 2018.

[16] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, DLRS 2016, page 7–10, New York, NY, USA, 2016. Association for Computing Machinery.

[17] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: A factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 1725–1731. AAAI Press, 2017.

[18] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions, 2017.

[19] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. Attentional factorization machines: Learning the weight of feature interactions via attention networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 3119–3125. AAAI Press, 2017.

[20] Guorui Zhou, Chengru Song, Xiaoqiang Zhu, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction, 2017.

[21] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. Deep interest evolution network for click-through rate prediction. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019.

[22] Lei Zheng, Vahid Noroozi, and Philip S. Yu. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, page 425–434, New York, NY, USA, 2017. Association for Computing Machinery.

[23] Rose Catherine and William Cohen. Transnets: Learning to transform for recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 288–296, New York, NY, USA, 2017. Association for Computing Machinery.

[24] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 297–305, New York, NY, USA, 2017. Association for Computing Machinery.

[25] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, page 111–112, New York, NY, USA, 2015. Association for Computing Machinery.

[26] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, page 153–162, New York, NY, USA, 2016. Association for Computing Machinery.

[27] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[29] Convolutional neural network. `https://en.wikipedia.org/wiki/Convolutional_neural_network`.

[30] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 507–517, Republic and Canton of Geneva, CHE, 2016. International World Wide Web Conferences Steering Committee.

[31] Ruining He and Julian McAuley. Vbpr: Visual bayesian personalized ranking from implicit feedback. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 144–150. AAAI Press, 2016.

[32] Qiang Liu, Shu Wu, and Liang Wang. Deepstyle: Learning user preferences for visual recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 841–844, New York, NY, USA, 2017. Association for Computing Machinery.

[33] Suhang Wang, Yilin Wang, Jiliang Tang, Kai Shu, Suhas Ranganath, and Huan Liu. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 391–400, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[34] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1835–1844, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.

[35] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 233–240, New York, NY, USA, 2016. Association for Computing Machinery.

[36] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 43–52, New York, NY, USA, 2015. Association for Computing Machinery.

[37] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. Representation learning of users and items for review rating prediction using attention-based convolutional neural network. 2017.

[38] David E. Rumelhart and James L. McClelland. *Learning Internal Representations by Error Propagation*, pages 318–362. 1987.

[39] File:recurrent neural network unfold.svg. `https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg`.

[40] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[41] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014. Association for Computational Linguistics.

[42] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. Deep coevolutionary network: Embedding user and item features for recommendation, 2016.

[43] Trapit Bansal, David Belanger, and Andrew McCallum. Ask the gru: Multi-task learning for deep text recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 107–114, New York, NY, USA, 2016. Association for Computing Machinery.

[44] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Sequential user-based recurrent neural network recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 152–160, New York, NY, USA, 2017. Association for Computing Machinery.

[45] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks, 2015.

[46] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[47] Oren Barkan and Noam Koenigstein. Item2vec: Neural item embedding for collaborative filtering, 2016.

[48] Mihajlo Grbovic and Haibin Cheng. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 311–320, New York, NY, USA, 2018. Association for Computing Machinery.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[50] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[51] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[52] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation, 2018.

[53] Qi Zhang, Jiawen Wang, Haoran Huang, Xuanjing Huang, and Yeyun Gong. Hashtag recommendation for multimodal microblog using co-attention network. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3420–3426, 2017.

[54] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. Multi-pointer co-attention networks for recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, jul 2018.

[55] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 701–710, New York, NY, USA, 2014. Association for Computing Machinery.

[56] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks, 2016.

[57] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[58] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1), Jun. 2014.

[59] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2181–2187. AAAI Press, 2015.

[60] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, Beijing, China, July 2015. Association for Computational Linguistics.

[61] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. A survey on knowledge graph-based recommender systems, 2020.

[62] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 353–362, New York, NY, USA, 2016. Association for Computing Machinery.

[63] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[64] Enrico Palumbo, Giuseppe Rizzo, and Raphaël Troncy. Entity2rec: Learning user-item relatedness from knowledge graphs for top-n item recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 32–36, New York, NY, USA, 2017. Association for Computing Machinery.

[65] Hongwei Wang, Fuzheng Zhang, Min Hou, Xing Xie, Minyi Guo, and Qi Liu. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 592–600, New York, NY, USA, 2018. Association for Computing Machinery.

[66] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon Jose. Relational collaborative filtering: Modeling multiple item relations for recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 125–134, New York, NY, USA, 2019. Association for Computing Machinery.

[67] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proc. VLDB Endow.*, 4(11):992–1003, aug 2011.

[68] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, page 283–292, New York, NY, USA, 2014. Association for Computing Machinery.

[69] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019.

[70] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. Recurrent knowledge graph embedding for effective recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 297–305, New York, NY, USA, 2018. Association for Computing Machinery.

[71] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. Meta-graph based recommendation fusion over heterogeneous information networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 635–644, New York, NY, USA, 2017. Association for Computing Machinery.

[72] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. Leveraging meta-path based context for top- n recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 1531–1540, New York, NY, USA, 2018. Association for Computing Machinery.

[73] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 417–426, New York, NY, USA, 2018. Association for Computing Machinery.

[74] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The World Wide Web Conference*. ACM, may 2019.