Expert Systems With Applications 186 (2021) 115708

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Is the suggested food your desired?: Multi-modal recipe recommendation with demand-based knowledge graph

Zhenfeng Lei, Anwar Ul Haq, Adnan Zeb, Md Suzauddola, Defu Zhang *

School of Informatics, Xiamen University, Xiamen 361005, China

ARTICLE INFO

Keywords:
Recipe recommender
Interpretation generation
Multi-modality
Knowledge graph
BERT

ABSTRACT

Personalized recipe recommender systems help users mine certain dishes they want to find and even really desire, which play a significant role in matching dishes, balancing nutrients, and preventing non-communicable diseases. Generally, customer's preferences or needs vary from person to person, and people are often reluctant to accept recommended food without reasonable explanation, especially when their demands are not explicitly addressed. In this paper, we are devoted to providing recipe suggestions accompanied by rational interpretations generated from images or videos. First, we construct a recipe knowledge graph (RcpKG) through the use of multi-modality and hierarchical thought, which focuses on the underlying demands of users and the consideration of multiple fine-grained factors. On this basis, a novel multi-modal recipe recommendation method via the knowledge graph (RcpMKR) is proposed, which represents nodes in multiple aspects and performs multi-relational graph structure extraction of the RcpKG. It not only takes into account local associations within the graph but also global information, and incorporates user concerns at different levels. Then, we adopt BERT-based multi-modal models and generative adversarial networks to generate interpretations. Additionally, dynamic convolution and random synthetic attention are utilized in our work to discriminate among features. Experimental results show that the proposed method and BERT-based fusion models improve recipe recommendation performance and explanation generation. Specifically, the precision of the RcpMKR method through RcpKG, user concerns and graph convolutional network improves by 7.82%, and the viExpCBTBERT method via 2D&3D convolutional neural networks for developing text interpretations enhances the F1-score by 10% compared with the baseline.

1. Introduction

With rapid development of the internet of things and digital communication technologies, more and more indispensable intelligent products and services are flooding people's daily lives. The exponential growth of data generated by these products and services has inevitably brought about information overload and asymmetry problems (Subramaniyaswamy et al., 2019). The rapid and continuous advancement in technology has changed our lives in many ways, and people pay more attention to their health. Personalized diet recommender systems (RS) were introduced to enable health-conscious people find their desired recipes quickly (Adaji et al., 2018; Chen et al., 2019). At present, personalized RS are commonly used and recognized by enterprises to quickly and conveniently meet various user needs and reduce the information load. A number of recommender systems have been developed for different scenarios, such as music and movie recommendation (Gong et al., 2019), news and advertisement recommendation (Zhou et al., 2020b), book recommendation (Wang et al., 2019b), on-line shopping recommendation (Feng et al., 2020), etc. Although, these systems are developed for different businesses according to their specific requirements, their functions are similar. Based on these successful recommendation applications, a personalized recipe recommender system (RcpRS) can be derived in terms of diet and health. Fig. 1 illustrates flowchart of a typical recipe recommender system.

The RcpRS is an intelligent mining platform based on massive dietary health data. A successful RcpRS should be able to suggest potential recipes that may be of interest to users to achieve personalized information services and decision support by analyzing user's historical behavior, mining user's preferences and health-related information, and predicting user's substantive actions on items such as collection, sharing, purchase, etc (Guo et al., 2020). Note that the recipes recommended here are considered whether they are beneficial to customer's health, rather than a simple third-party suggestion. For example, if

* Corresponding author.
E-mail addresses: zflei621@foxmail.com (Z. Lei), anwar@uom.edu.pk (A. Ul Haq), adnanzeb@stu.xmu.edu.cn (A. Zeb), suzau@stu.xmu.edu.cn (M. Suzauddola), dfzhang@xmu.edu.cn (D. Zhang).

https://doi.org/10.1016/j.eswa.2021.115708
Received 8 April 2021; Received in revised form 10 July 2021; Accepted 31 July 2021
Available online 14 August 2021

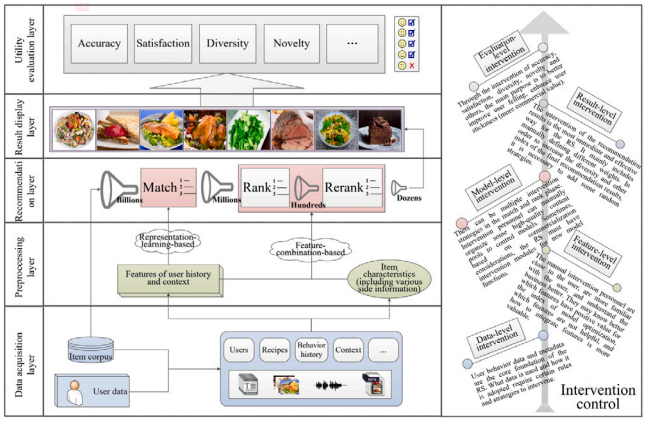

Fig. 1. Flowchart of a typical recipe recommender system.

a user likes seafood but has cold, the recommendation should be a seafood recipe with vegetables, because vegetables can relieve cold. A personalized recipe recommender system includes two steps :matching and ranking (Parul, 2019; Paul et al., 2016). The matching (also called candidate generation) is based on user's past activities such as browsing, search history and contexts to quickly retrieve a small number of potentially interesting recipes from the massive inventory. The ranking step uses more refined features to calculate the predicted scores of user-recipe pairs generated during the matching step. The ranking step can be further divided into rough-rank and precise-rank (marked as Rank and Rerank respectively in Fig. 1). The rough-rank usually adopts a simple ranking model that uses few features to initially rank the matched items, and to further reduce the number of these items. The re-rank utilizes a complex model that employs more features to accurately rank a small number of items to achieve higher accuracy, satisfaction, diversity, novelty, and so on. The rough-rank is optional according to the specific application (Zhang, 2020). It is worth mentioning that in matching step the emphasis is on speed, which refers to efficient feedback results, and the ranking step focuses on accuracy, which means high precision of recommended results.

A recommender system automatically provides personalized content for users with the help of machine learning (ML). Although recommender systems can recommend better than individuals in many cases, they cannot be compared with humans in many aspects, especially in scenarios involving human health. Intervention control is embedded to achieve high-quality user experience and to better achieve business goals (Liu, 2020). Each stage in the RcpRS needs intervention control, such as what kind of data to crawl, what features to choose, how to define the model, and how to display the suggested results, etc., as shown in the right segment of Fig. 1. Intervention control is an important part of recipe recommender systems as the RcpRS is not only required to accurately discover a customer's potential choices but also meet the user's abstract requirements, such as explanation, surprise, and others. Moreover, user intervention enables us to manually scrutinize suggested items to avoid allergic reactions or worsen any underlying health issues. For example, it is not allowed when spicy food is recommended for a person who has gastric ulcer, even if that is the favorite dish. In this paper, we integrate intervention control in the feature extraction and generating the final list of recommended recipes.

Different applications have their special services or functions, which leads to multiple heterogeneous data sources for items people need, i.e., multimedia. A user may comment about a certain recipe in the form of text, and may express his views in the form of image, audio or video. Although different ways or mediums express opinion about the same recipe, they may deliver different intentions. Through representation learning (RL) based on multi-modality, powerful multi-level abstract representation capabilities can be obtained. In recent years, it has received extensive attention (Xu et al., 2018). In this paper, we propose an extensible multi-modal recipe fusion framework, which combines knowledge graph (KG) (Guo et al., 2020) to point out a direction for improving predictive performance and generating interpretations in the recipe recommendation. Among the current recommender systems, there are common problems such as repeated recommendations and lack of novelty. This is because they match items only from single-modal user historical behaviors and do not consider modeling the specific needs of users (Luo et al., 2020). Moreover, user demands can be divided into scenarios, multiple behaviors, numerous intentions, etc., which make it challenging to optimize the recommendation model from multiple perspectives (Xu et al., 2020). Nowadays, exploring and using the homogeneous or heterogeneous relationship in the graph is a very promising direction for developing personalized RS. This paper first constructs a recipe KG with hierarchical and diverse requirements according to the specific needs of users, and then proposes a novel multi-modal recommendation method based on this to improve recommender performance.

Another common problem with existing recommender systems is that their recommended results have no or limited interpretation. The lack of a reasonable explanation is very likely to cause users to distrust the RS (Chari et al., 2020). People like things that change dynamically, and their needs vary at any time. Users are putting forward new requirements according to different scenarios all the time. If users can get an experience beyond expectations, that is, the suggested items are not only accurate but also surprising, then they will definitely trust the recommender system more. Research showed that without adequate explanation, users may not take any action based solely on the results recommended by the RS (Dong et al., 2020). From a user perspective, the recommender system not only needs to show the results but also explain the reason and meaning of the suggested results to the user. Because if users think that the recommended content is not accurate enough, they will consider the RS to be biased in some way. Especially, when it comes to highly subjective recommendations, the interpretation for the recommendation is as equally important as the results. For example, when recommending a dish with extremely high nutrients to users, it is difficult to recommend it successfully without a good explanation. As users cannot verify which nutrients the recipe contains in a short time (the subjectivity is relatively larger). Therefore, how to generate better explanations will not only affect user experience, but will also greatly promote the development of the interpretable recommender system and even the explainable artificial intelligence (AI) system. Our work generates more reasonable interpretations from the perspective of multi-modality model construction.

In this paper, the main contributions are:

- An extensible multi-modal recipe fusion framework with a knowledge graph is proposed, which lays the foundation for using multi-modality to improve recommendation performance and generate explanations for the recommendations.
- Constructing a recipe knowledge graph centered on the fine-grained needs of users with multi-modality and hierarchical idea. On this basis, a novel recipe recommendation method that can express multiple aspects of graph nodes and extract multi-relational graph structures is proposed, which not only reflects the global information and the local associations within the graph, but also hierarchically deals with different concerns of users.
- Applying the BERT-based multi-modal model and generative adversarial networks to generate reasonable explanations for suggested results in two ways, from image to text and from video to text.

The rest of the paper is organized as follows: Section 2 discusses the latest research work and technologies of the recommender system. Section 3 is the methodology, which involves the construction of the recipe knowledge graph based on user demands, the proposed framework and methods, and specific models. Section 4 shows the experimental details,

including data set acquisition, evaluation metrics, experiment settings, and the corresponding results and discussions. Finally, conclusions and directions for future work are given in Section 5.

2. Related work

Information overload and asymmetric information have made it difficult for users to accurately identify relevant information items (Wang et al., 2018a). Personalized recommender systems tend to enable users to lock the content they are interested in, so as to provide convenience for users' information acquisition and online experience. As an effective tool to meet the unclear demands of human beings, RS are of great value, especially in health dietary. This value is not only reflected in improving the user experience, but also in obtaining commercial benefits. Therefore, personalized recommender systems have received extensive attention from academia and industry (Fadhil, 2018; Gao et al., 2019; Wang et al., 2020e; Zhang et al., 2016).

However, in view of the huge commercial benefits of recommender systems, most companies define the primary objective of RS as achieving higher monetary benefits, which has led to excessive commercialization. For example, there are some short videos and images that are suitable for mature audiences only, and their comment areas are even more unsightly in the information flow recommender of some APPs. Some bad recommendation algorithms give too much "praise" to items they provide, maliciously increasing the value of those items to get more traffic. Moreover, some recommender systems may keep pushing items based on customer's search history or others, while ignoring high-quality new products. This approach gradually marginalizes new products, leading to repeatedly recommending similar items. The problem can also be avoided with intervention control (Parul, 2019). In China, some restaurants have good faith requests for customers to limit the number of dishes based on the number of people at the table. In other words, restaurants do not allow customers to order more dishes to avoid food wastage. In some universities, such as Xiamen University, canteens are advocating and implementing the "Scientific Ordering" action. Another good example is embodied in the login of the mobile game, such as specially set up youth model (Zhang et al., 2020a).

Since most of the current data appear in multiple modal forms, multi-modal RL has received increased attention in recent years (Guo et al., 2019; Qiao et al., 2020). In Xu et al. (2018), the authors took into account information extracted from images or videos for improving the top-n recommendation performance. (Zhou et al., 2020b) used a transformer-based cross-modal encoder and other related technologies to obtain multi-modal information for automatically recommending advertisement themes. Suggested items must be considered in many aspects because user needs can be divided based on scenarios, behaviors, and points of interest. One of the effective ways to solve the multi-view problem is to use graphs (Hassani & Khasahmadi, 2020; Wang et al., 2020b, 2020c). The earliest use of graphs is how to introduce additional information such as item prices and user profiles into graph representation learning. Some researchers use supplementary information such as social networks or item attributes to improve recommendation performance, that is, naturally incorporating graphs into the recommender system (Berg et al., 2017; Saxena et al., 2020; Wang et al., 2020a, 2018a, 2018b). Feng et al. (2020) proposed an ATBRG framework to effectively capture the structural relationship of user–item pairs in the knowledge graph. In order to mine users' multiple intentions, Wang et al. (2020c) proposed a multi-component graph convolutional collaborative filtering method to distinguish potential multiple purchase motivations under the observed explicit user–item pairs. In order to accurately identify customer's real intentions from written or spoken language, Vedula et al. (2020) proposed a new domain-independent method, which expresses the problem as a sequence marking task in an open world environment. At present, much work has been carried out from hierarchical perspectives (Gao et al., 2019; Xu et al., 2020). Chen et al. (2020b) uniformly considered multi-level graph convolution on local network structure and hyper-graph structure to realize the aligned embedding space in profile-matching across different social networking platforms. Wang et al. (2019a) emphasized the importance of higher-order relationships in KG, and proposed a graph neural network framework to achieve higher-order relationship modeling.

In recent years, academia and industry have recognized the importance of interpretability, and interpretable recommender systems have received increasing attention (Dong et al., 2020). Explainability research is an interdisciplinary topic in many disciplines such as AI, ML, cognitive psychology, logic, etc. It has theoretical research significance and practical application value in the field of information push. Some researchers have been studying and dealing with interpretability centered on users, looking for reliable, understandable, clear, and context-aware explainability. This helps designers obtain reliable requirements and determine the priority of various demands, and even further help generate interpretations that better meet user needs in various scenarios. Some popular journals like *Nature*, *Science* and *MIT Technology Review* have special articles discussing this issue (Cheng et al., 2020; Gunning et al., 2019). Besides, some top conferences such as *AAAI2019* have also set up a discussion topic on explainable AI. In addition, the introduction of knowledge graphs as side information into the recommender system has stimulated great interest in recent years. Explainable AI not only alleviates the above issues and provides more accurate recommendations but also explains the recommended items (Bosselut et al., 2019; Luo et al., 2020). Zhao et al. (2020) designed a demonstration-based KG reasoning framework for interpretable recommendations, mainly to solve the problem of reasoning combination optimization by better supervising the pathfinding process. Another approach to improve the interpretability of the RS is to make full use of long-tail items[1] Kim et al. (2019) and Zhou et al. (2020a), to illustrate the rationality of suggested items. This paper tends to generate reasonable explanations by considering multimodal ideas.

3. Methodology

3.1. Construction of demand-based recipe KG

Generally, any data can be topologically associated in the normed space, i.e., it can form a graph. Many forms of data in the recipe recommender system can also be represented as graphs. Similar to social networks found among users, a recipe graph can be built among dishes. The two graphs can then be used to construct a bipartite graph. A heterogeneous information network can be generated by taking the above graphs into consideration. Furthermore, the dynamic evolution generated by taking the time factor into consideration can form a dynamic graph. Their structural inclusion relationships are shown in Fig. 2.

A person may like many dishes, and a recipe can be liked by various people. With so many choices in life nowadays, recommending a new recipe to customers is quite difficult. It is relatively easy for the RcpRS, if a person's demands are clear. However, in many cases, users often face ambiguous choices or scenarios, and they do not know what they need most. Therefore, the premise that a recipe can be accepted by a user is that he considers many factors. As shown in the upper part of Fig. 3, a customer goes through many steps to choose a certain recipe he/she likes. The user's preferences often follow a hierarchical model, which is recognized by researchers, i.e., people's preferences are diverse and hierarchical. For example, some people select a dish based on its nutritional value, while others choose that dish based on its bright color. Another relatively common factor is the cost effectiveness of the dish. In our daily online ordering, users may quickly skip some

[1] Long-tail items mean majority of the data belongs to a small number of categories, and most other categories have only a few samples.

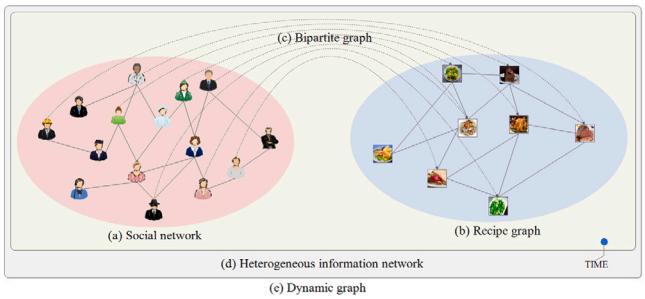

Fig. 2. The inclusion relationship of various graphs.

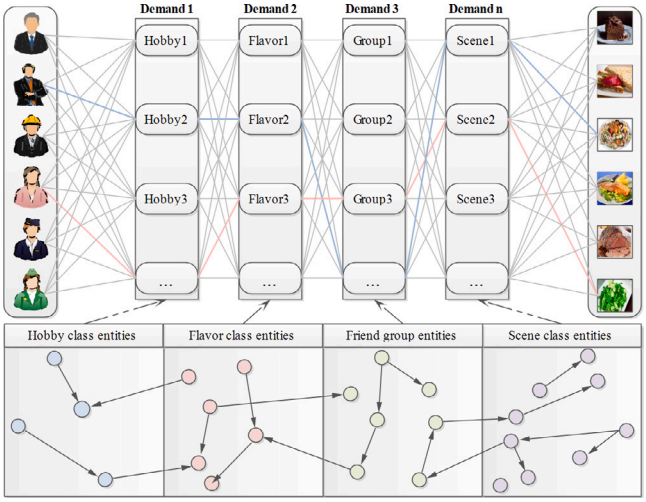

Fig. 3. Recipe knowledge graph construction with different demands (RcpKG). The upper part is the process of choosing the recipe according to different user needs; the lower part is the recipe knowledge graph generated based on user's requirements.

recipes they do not like based on their own experience and directly buy the ones they like more. Typically, customers keep adding products to shopping carts and take the final decision at checkout. In addition, recommendations are effective only in specific scenarios. For example, a customer searches for tomato dishes in cold weather, the system should be able to suggest dishes served in cold weather, such as stirred egg with tomato instead of tomato salad. Therefore, if the context of a user query is not considered in a specific scenario, it will result in a poor user experience. In many previous works, user stratification, diverse interests, multi-factor influences, and scenarios have not been well utilized. Moreover, most of the existing knowledge graphs are constructed using only one aspect.

In order to consider a variety of potential fine-grained factors, inspired by Bosselut et al. (2019) and Luo et al. (2020), we construct a hierarchical recipe knowledge graph with more specific demands, called RcpKG. It can explicitly express user's requirements as nodes in the graph. Constructing a graph centered on user demand nodes can better link the graph network of user-profiles, item characteristics, and other interactions. RcpKG and existing knowledge graphs both mine deep-level relationships between users, and the most significant difference with them is that we have introduced multimedia storage. The image and video data are represented in our graph by index. By referring to the definition of the specific requirements, our most significant advantage is to more directly display the relationships between users and items from multiple aspects. As shown in the lower part of Fig. 3, the demand-based recipe knowledge graph is constructed hierarchically according to the corresponding factors above. In this paper, we construct it according to six different perspectives, including user hobbies, real intentions, group influence, recipe characteristics,

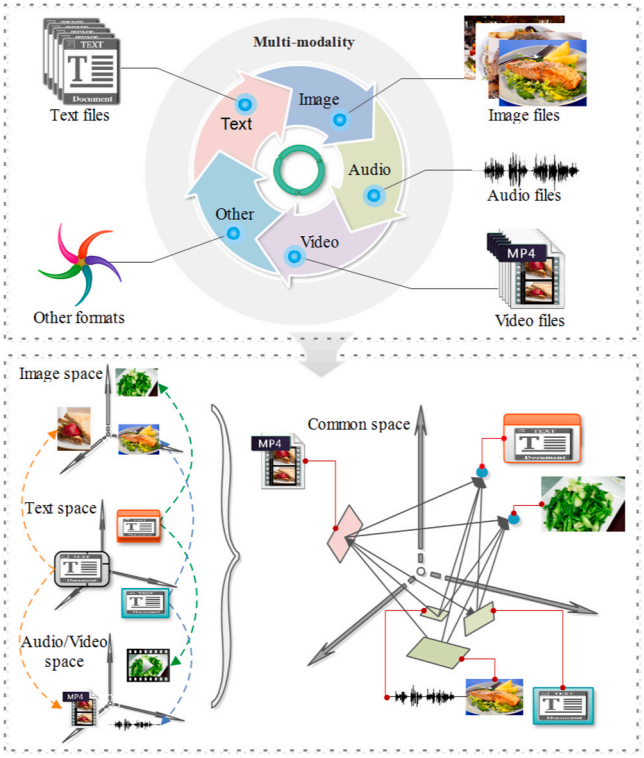

Fig. 4. Multi-modality and their corresponding representations in different spaces.

location, and occasions. Regarding entity naming recognition and relationship extraction, we use a combination of manual and automated extraction methods. More detailed information can be referred to as part of the above two existing works.

Based on the constructed RcpKG, this section discusses specific methods. First, we introduce the proposed overall recipe fusion framework based on multimedia data. Then, the methods used in this paper are described from two aspects, i.e., improving the recommendation performance and generating explanations.

3.2. Proposed overall framework

Recommender systems usually contain multimedia data of highly personalized interactions about user–item pairs in different scenarios, especially in recipe recommendation. As shown in the upper part of Fig. 4, the data exists in text, image, audio, video, or other forms. Employing only a single medium of data to model the complex relations will not provide the best advice to customers as important information may be lost during sampling. Moreover, it usually leads to a non-optimal performance in practical applications (Wang et al., 2020e). Therefore, recommending recipes in a multi-modal form seems to be the right solution. Multi-modal RL aims to narrow the heterogeneity gap between different modalities, and plays an indispensable role in the use of ubiquitous multimedia data. Multi-modal representation learning can make it possible for users to select the best model via the user's specific historical data. The lower part of Fig. 4 shows how multimedia data can uniformly represent recipe information in a common space (Guo et al., 2019).

Based on the existence of multimedia data in the field of recipe recommendation, we propose an extensible three-modal recipe fusion framework with knowledge graphs as shown in Fig. 5. (a) can be understood as a combination using two mediums, i.e., text and image. First, the text features are extracted through Doc2vec (Luo et al., 2020) or FastText (Qiao et al., 2020). The image features are extracted through deep convolutional neural networks (CNN) models, such as

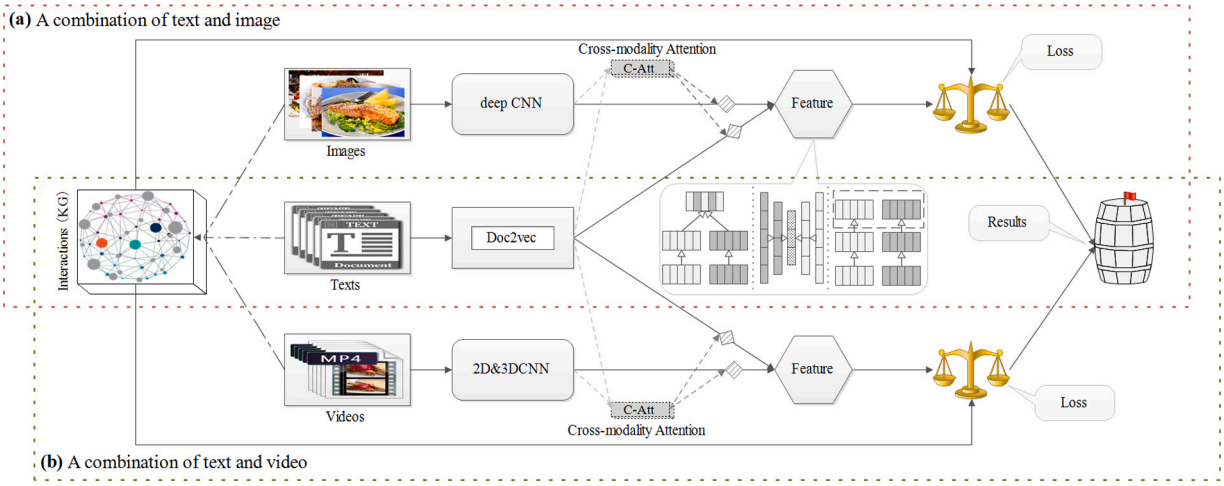

Fig. 5. An extensible three-modal recipe fusion framework with knowledge graphs.

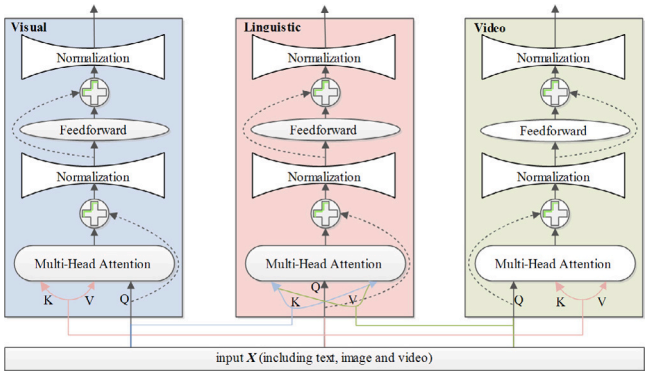

Fig. 6. Three-mediums BERT fusion.

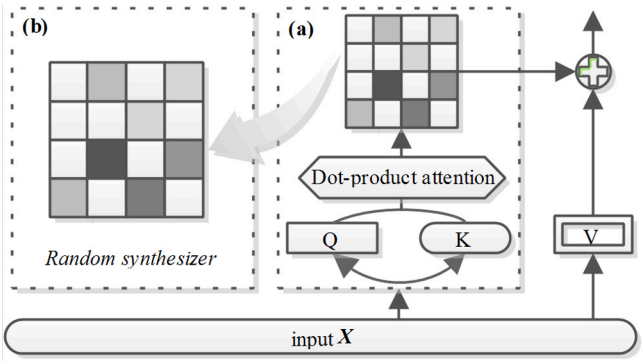

Fig. 7. Random synthesizer attention.

FasterRCNN, ResNet-*n* (Li et al., 2019; Su et al., 2019; Xu et al., 2018). Then, features from the two mediums are merged through the attention mechanism or other fusion ways. The fusion way can be summarized into three forms, i.e., joint, encoder–decoder and coordination (Guo et al., 2019). The fused features will be combined with the KG for the final recommendation or generation. (b) is the fusion of two mediums of text and video. Similarly, the process is similar to (a). Due to the existence of a large number of high-order *lookup* features, item *id* features, and discretized numerical *id* features, we will perform normalization when embedding. If there is no normalized feature, the convergence of the network will not be guaranteed. It is worth noting that the method shown in Fig. 5 can be extended. In other words, our proposed framework has good scalability. In specific experiments, we can select appropriate algorithms based on different field data for better recommendation or generation results. For example, video coding can use the 2D&3DCNN (Zhang et al., 2020b) for feature extraction, of course, it can also use S3D algorithm (Sun et al., 2019a, 2019b) or ResNet-*n* to extract features.

During the encoding and feature fusion, our proposed framework can fuse features of the two modalities separately, and then merge the fused features obtained to attain the final recommendation or generation results via attention-based mechanisms. Inspired by Lu et al. (2019), we propose a BERT-based feature fusion way of the three mediums (Zhang et al., 2020a), as shown in Fig. 6, which learns interaction relationships from the three separate mediums to get a unified expression, so that it can capture more comprehensive fusion information. In the text encoding and feature extraction, we assign different weights to words as the semantic information of the same word or the same sentence may differ in various contexts. When users face many items, the importance of different scenarios is also not the same. For example, when facing a local snack, a user may focus on characteristics rather than cost-effectiveness and high nutritional value. Similarly, when the item is expressed in different modalities, the importance of its different modal features is also different. Most people may be attracted by images of that local snack first, and then read the text introduction about the snack. If there is a related video, the user will make a final decision after watching the video without knowing whether to purchase the snack. Therefore, it is very important to learn a multi-modal relationship between users and items. On the other hand, under the premise that a user's preferences for various items may be different, the representations given by different modes are also different. Therefore, learning about a customer or product from a single medium of information and presenting it in the same medium will be flawed and cannot fully capture the overall characteristics of user preferences and items.

The attention in the standard Transformer (Gong et al., 2019; Vaswani et al., 2017) requires pairwise interaction between tokens, as shown in Fig. 7 (a). Although interactive information can be obtained, the attention score strongly depends on examples. Moreover, it is difficult to ensure that the model learns more generalized features, and its calculation may be unnecessary. In this paper, we adopt another method, i.e., random synthesizer attention proposed by Tay et al. (2020), as shown in Fig. 7 (b). The core idea is to replace the dot product operator with low-complexity attention calculation and compute the attention score solely through a simple feed-forward neural network, eliminating the dot product interaction between tokens. This method does not rely on the interaction between token pairs or the

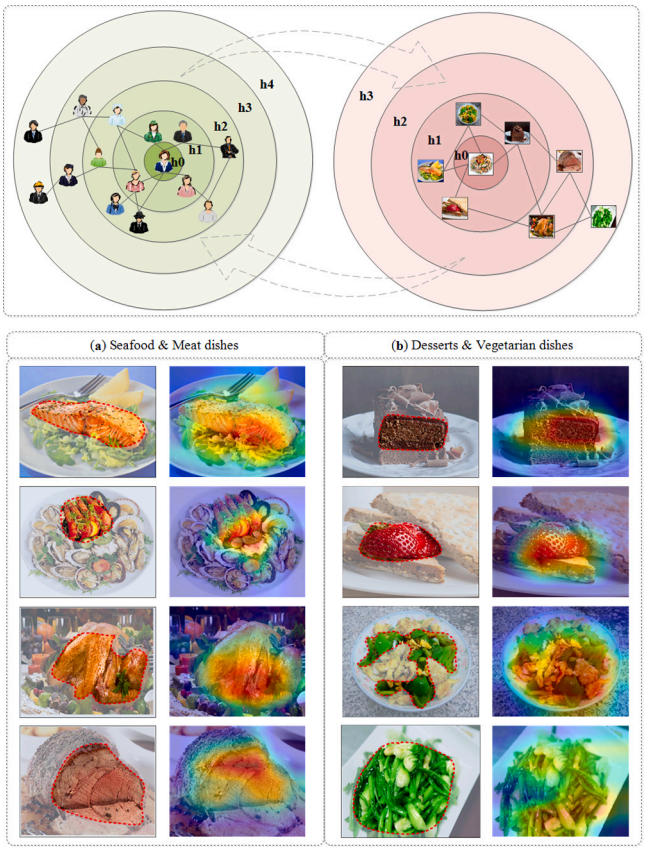

Fig. 8. Hierarchical users or items and different concerns for users. The upper left is a graph of the user's social network with hierarchy, and the upper right is a recipe graph with hierarchy. The lower is a concern example of different users on various recipes.

information of any single token, but learns a task-specific alignment that can effectively cross-instance. Specifically, Query and Key are removed, and a matrix having the number of rows and columns of the sequence length is used to represent the relationship between tokens, i.e., $\mathrm{softmax}(QX \cdot KX)(VX) \Rightarrow \mathrm{softmax}(R)G(X)$. Here R represents a random initialization matrix, and $G(X)$ can be analogized to V in the standard Transformer.

3.3. Improving recommendation performance

In recent years, knowledge graphs have attracted increasing attention due to the successful application of their rich connection information in the context of recommender systems. However, increasing the number of views or multi-scale encoding may not improve the recipe recommendation performance, because of the hierarchy of question in building a RcpKG. We need to analyze the graph's high-order neighbor encoding and graph diffusion (Liu et al., 2020). In addition, users often have many points of interest at a certain moment when they "visit" the RS. If a recommender system does not identify and utilize different points that users pay attention to, then it will not satisfy the users. Conversely, if the recommender system can cover most of the user's concerns, it will increase the probability of a successful transaction. Therefore, we need to identify different user concerns in an effective way. Next, we will specifically introduce two aspects of the problem and then propose the corresponding solutions.

Graph hierarchy. Whether it is a social network or a recipe graph, there will be hierarchical information. We cannot simply think that a user chooses a certain recipe due to he really likes it, because in most cases users' choices are influenced by other users or other recipes. High-level social relations such as friends of friends and colleagues of colleagues play a vital role to a large extent, as shown in the upper left part of Fig. 8. Therefore, we should focus on modeling the indirect effects of high-order neighbors in social networks, such as through multi-hop (Liu et al., 2020; Saxena et al., 2020; Wang et al., 2019a, 2019b, 2018a), by following the multiple-layer links of the graph to discover the potential interest of users. It is like sometimes we wonder why I chose this recipe in the end. The root cause is that customers decisions are mainly influenced by other people or objects. The recipe graph is shown on the upper right-side of Fig. 8. In the same way, there is a high-order relationship among recipes. If there are no apparent reasons why a recipe is related to another recipe, we cannot simply think that it is unreasonable, perhaps because of their high order influence. While considering the hierarchical structure between people or among recipes, the degree of their influence is also different. This is because the ability of people (or recipes) to influence others varies in some way. If a user has a greater influence on an item, the corresponding probability, such as ordering will also be greater. For example, in deciding what feast meal a family should eat at a restaurant, if the father has been to a certain seafood restaurant many times, then the father will have more power to decide whether to eat seafood, which is more in line with people. Influential individuals have mainstream tastes and are highly distracted by other people's tastes. The same can be deduced for recipe relationships, cuisines similar to a certain popular recipe will have a high level of acceptance. Therefore, the degree of its influence in the hierarchy should be considered in the recipe recommendation.

Multiple concerns. A recipe may be liked by different people for different reasons. For example, a steamed seafood dish was ordered by different users. It may be that some users have taken a fancy to the shrimp in this dish, while others want to taste the crab in the dish. Another dish, fried fish salad, is liked by people maybe because some persons like fried fish while others like salad. Therefore, if the recommender does not consider different characteristics of a dish from customer perspectives, then the recommendation will not satisfy the user. As shown in the lower of Fig. 8, we use *Grad-CAM* (Dong et al., 2020) to find the attention map showing different points of concern. (a) shows four popular dishes with higher nutritional content in meat and seafood, while (b) is concerned with lighter ingredients in desserts and vegetarian dishes. According to different user information, if our recommendation can cover most of the user's concerns, it will increase the probability that the user is suggested the recipe he is interested in, so that the recommendation can be more accurate and successful.

3.3.1. Proposed recipe recommendation method (RcpMKR)

Based on the above two aspects, we propose a novel multi-modal recipe recommendation method via the knowledge graph (RcpMKR) shown in Fig. 9. RcpMKR can be divided into four parts: feature extraction based on the basic interactions between users and recipes, feature information extraction of graph nodes, multi-relational graph structure extraction, and recipe feature extraction based on different user focuses.

Regarding the basic interactive extraction of graphs, we use the most basic embedding method for feature extraction, which will not be described in detail here. Regarding the feature information extraction of nodes, we use high-order neighborhood encoding and graph diffusion to achieve node-level and graph-level multi-aspect representation (Hassani & Khasahmadi, 2020). Graph diffusion is used to generate a structural view of the sample graph, which is sampled and passed to graph neural network (GNN) or graph convolutional network (GCN) (Berg et al., 2017; Jin et al., 2020; Wu et al., 2020) for feature extraction like regular views. Graph attention networks (GAT) (Wang et al., 2019a) manipulate the structure space expansion and diffusion matrix on the graph structure by adding or removing connectivity, subsampling, or using the shortest distance to generate a global view. We convert the adjacency matrix to a diffusion matrix and treat these two

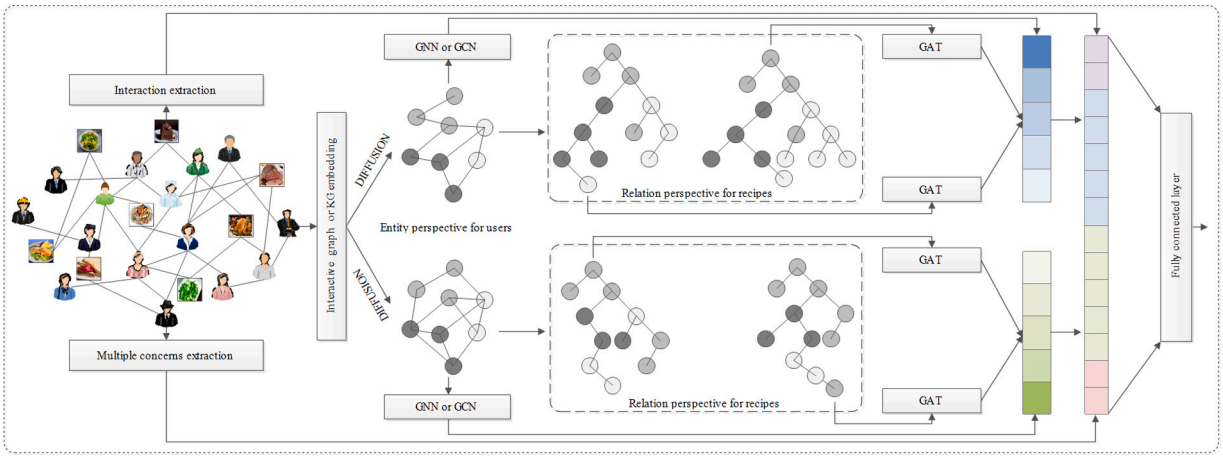

Fig. 9. A novel multi-modal KG-based recipe recommendation method (RcpMKR).

matrices as two congruent views of the same graph structure. Since the adjacency matrix and the diffusion matrix, respectively, provide local and global views of the graph structure, the maximum consistency between the representations learned from these two views allows the model to simultaneously encode rich local and global information. The key of diffusion can be expressed as $\sum_{k=0}^{\infty} \Theta_k T^k$. Here, Θ means the weighting coefficient, which controls the proportion of local and global information. And T represents the transition matrix. To train the rich nodes and graph-level representations that are unknown in the learning graph, we use the *deep InfoMax* method to maximize the MI by comparing the node representation of one view with the structural representation of another view. The objective can be defined as follows:

$$\max_{\theta,\omega,\phi,\psi} \frac{1}{|\mathcal{G}|} \sum_{g\in\mathcal{G}} \left(\frac{1}{|g|} \sum_{i=1}^{|g|} \left(\mathrm{MI}\left(\vec{h}_i^{\alpha}, \vec{h}_g^{\beta}\right) + \mathrm{MI}\left(\vec{h}_i^{\beta}, \vec{h}_g^{\alpha}\right) \right) \right) \tag{1}$$

where θ, ω, ϕ and ψ represent the encoder and projection parameters. Note that $|\mathcal{G}|$ represents the number of graphs. $\vec{h}_i^{\alpha}$ and $\vec{h}_g^{\beta}$ denote representations of node i and graph g, which are encoded from the α and β aspects respectively.

To fully extract the relational structure information, we perform pruning operations on the obtained graphs (Feng et al., 2020). In the recipe recommendation, the user may select the same recipe by different actions such as online clicking or buying operation. Obviously, various behaviors symbolize the user's different preferences for recipes. Specifically, we will learn the different relationships between entities and the rich structural information in KG by constructing a relation-aware extraction layer. The relation-aware extraction is to effectively and comprehensively extract the structural connection information in the relation subgraph. In addition, we regard a relationship as the central point to aggregate its neighbor information in the relationship subgraph. Multi-relational graph structure extraction can be achieved by:

$$\alpha^{(l)}(h,r,t) = \frac{\exp\left(x_r W_\alpha f\left(x_h^{(l)} \bowtie x_t^{(l)}\right)\right)}{\sum_{(r',t')\in\mathcal{N}_h^{(l)}} \exp\left(x_{r'} W_\alpha f\left(x_h^{(l)} \bowtie x_{t'}^{(l)}\right)\right)} \tag{2}$$

where $x_h^{(l+1)} = x_h^{(l)} \bowtie \sum_{(r,t)\in N_h^{(l)}} \alpha^{(l)}(h,r,t) x_t^{(l)}$. Note that $\mathcal{N}_h^{(l)} = \{(r,t) \mid (h,r,t) \in \mathcal{G}_{ui}\}$ and $x_h^{(l)}$ represent the neighbors and the representation of entity h in lth layer, respectively. And $f(x)$ means the perceptron and W_α is the attention matrix. Aiming at the feature extraction of the user's different focuses, we mainly use the *Grad-CAM* algorithm to calculate the user's concerns, and then perform the corresponding feature fusion.

When the RcpMKR method extracts structural information, the graph will depend on the context, which will cause different layers to have various structures. If ordinary static convolution operation is used, a lot of key information will be lost. Here, we will apply a dynamic convolution structure to deal with the above issue, as shown in Fig. 10. The design of dynamic convolution can increase the representation capacity of the model without increasing the depth or width of the network. The basic idea is to adjust the convolution parameters adaptively according to the different feature structures of the input. Dynamic convolution does not use a single convolution kernel on each layer, but dynamically aggregates multiple parallel convolution kernels based on attention. Attention will dynamically adjust the weight of each convolution kernel according to the input, thereby generating adaptive dynamic convolution. Since attention is a function of input, dynamic convolution is no longer a linear function. Attention will have a stronger representation ability by superimposing the convolution kernel in a non-linear way. The dynamic perceptron can be redefined as follows (Chen et al., 2020a):

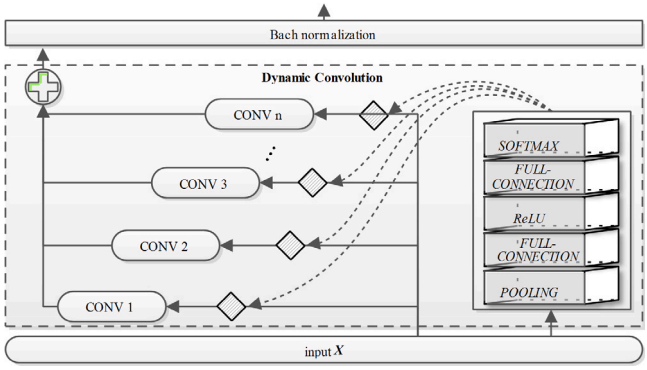

Fig. 10. Dynamic convolution structure.

$$\triangle = g\left(\tilde{W}^T(x)x + \tilde{b}(x)\right) \tag{3}$$

where $\tilde{W}(x) = \sum_{k=1}^{K} \pi_k(x)\tilde{W}_k$ and $\tilde{b}(x) = \sum_{k=1}^{K} \pi_k(x)\tilde{b}_k$ $(0 \le \pi_k(x) \le 1, \sum_{k=1}^{K} \pi_k(x) = 1)$.

3.4. Generating explanations

Generally, the recommended results using deep learning are more accurate than the results of traditional methods. However, DL has always been used as a black box in laboratory research, and no explanation has been given about the reasons for how to obtain these results and how to determine the parameters that make the results better. At the same time, when there is an error in the result, it is impossible to explain why the error occurred and even how to solve the error.

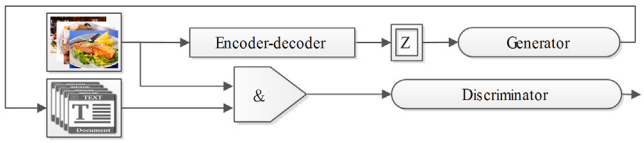

Fig. 11. A GNN structure to generate text interpretation.

Therefore, it is essential that a recommender system using DL be trusted by users through reasonable explanations. This paper will try to use multi-modal approaches to generate the corresponding interpretation for suggested results. It is mainly divided into two directions, one is to generate explanations from images, and the other is to use videos to produce corresponding interpretations.

3.4.1. Image to text generation

We use two kinds of methods to generate textual interpretations via images, that is, using generative adversarial network (GAN) as shown in Fig. 11, and the BERT-based models as shown in Figs. 12 and 13. Their detailed introduction is given below.

GAN is an emerging deep learning technology, which can generate high-quality new samples according to the distribution of training data (Guo et al., 2019). Recently, generative adversarial learning strategies have been further extended to multi-modal cases, such as text-to-image synthesis, visual captioning, cross-modal retrieval, and multi-modal feature fusion. This paper uses a GAN to generate text explanation with images (imgExpGAN). The structure used is shown in Fig. 11.

A GAN consists of two parts, the generative network G and the discriminatory network D. They compete with each other. The G is responsible for generating new samples based on the learned data distribution. The aim of D is to distinguish the difference between the examples generated by G and the items extracted from the training set. Their optimization objective can be defined as:

$$\Gamma = \min_{G} \max_{D} V(G, D) \tag{4}$$

where $V(G, D)$ denotes the cross-entropy loss, i.e., $V(G, D) = E_{x \sim p_{\text{data}}(x)}[\log D(x)] + E_{z \sim p_z(z)}(1 - \log D(G(z)))$. The cross-modal adversarial model contains an encoder network that transforms one modality into another. For example, given a pair of inputs $(t; v)$, the encoder maps v to a representation vector, and then the decoder (as a generator) maps the vector to the reproduction sample $\vec{t}$. The generated sample $\vec{t}$ is expected to be sufficiently similar to t so that they are regarded as a true pair by the discriminator.

This paper uses a multi-modal two-stream BERT model to generate explanations with images (imgExp2SBERT), which supports two streams of input (Li et al., 2019; Lu et al., 2019; Sun et al., 2019b). It pre-processes the visual and textual inputs in the two streams separately, and integrates them in the joint attention layer shown in Fig. 12. There are three tasks in the pre-training stage, including masking language modeling, masked object label prediction, and visual-text matching.

The first two tasks allow the model to learn content-related representations from a joint token based on textual and visual content inputs. The latter task attempts to predict whether an image matches a text description. The joint probability distribution can be defined as:

$$P(x \mid \theta) = \frac{1}{Z(\theta)} \prod_{l=1}^{L} \phi_l(x \mid \theta) \propto \exp\left(\sum_{l=1}^{L} \log \phi_l(x \mid \theta) \right) \tag{5}$$

Here $Z(\theta)$ represents the partition function and $\phi_l(x)$ means the lth potential function. Since BERT is a heavyweight model, we can try to use the knowledge distillation technology.

Another multi-modal BERT-based model we use is called imgExpRoIsBERT in the paper, which takes the region of interest (RoIs) in the text sentence or in the image as the feature input (Su et al., 2019; Tan & Bansal, 2019). In model training, each element can adaptively aggregate information from all other elements according to its content, location, category, and other properties. After stacking multiple layers of the attention module, its feature representation will have higher ability to aggregate and align visual and textual cues. In order to better model the general vision-text representation, we pretrain the imgExpRoIs-BERT in a large-scale vision-text recipe corpus. In BERT pre-training, we introduce the masked language modeling, and use a number of different methods to extract feature information of the image, such as CNN, ResNet-n (n = 145 is best) and FasterRCNN. In the training of the masked language modeling of BERT, the joint probability distribution Eq. (6) is optimized ($x_{\setminus i} = \{x_1, \ldots, x_{i-1}, [\text{MASK}], x_{i+1}, \ldots, x_N\}$).

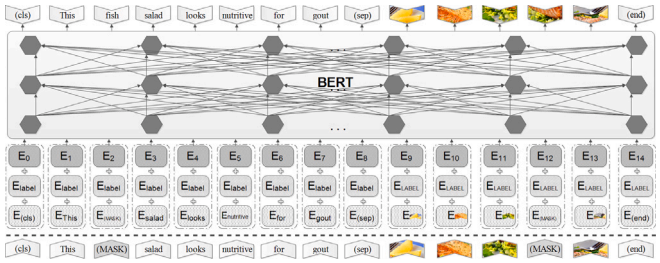

Fig. 12. A multi-modal two-stream BERT model for generating explanation.

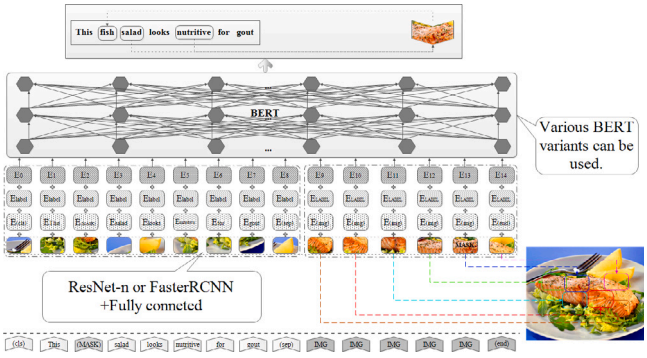

Fig. 13. A multi-modal BERT model based on the region of interest for explanation generation.

$$\log P(x \mid \theta) = \frac{1}{Z(\theta)} \sum_{i=1}^{N} x_i^T f_i\left(x_{\setminus i} \mid \theta\right)_i \tag{6}$$

3.4.2. Video to text generation

Using video to generate text explanation is like video description, or it can be described as talking through videos. Unlike the single static information in images, video data can combine spatiality, time, and language information at the same time. It requires not only the full extraction of the key visual information in the video, but also the construction of a mapping relationship from vision to text. This paper uses a fairly simple but efficient fusion method to generate text interpretation through video (Sun et al., 2019a, 2019b; Zhang et al., 2020b; Zhou et al., 2018), which is called viExpCBTBERT as shown in Fig. 14. The pre-training of the model is for encoding and feature extraction, then it is generated through cross-modal conversion, and finally through the loss function to determine whether it is correct.

$$\mathcal{L} = w_1 L_{\text{text}} + w_2 L_{\text{video}} + w_3 L_{\text{cross}} \tag{7}$$

where $L_{\text{text}} = -E_{\mathbf{y} \sim D} \sum_{t=1}^{T} \log P\left(y_t \mid y_{-t}\right)$, $L_{\text{video}} = -E_{\mathbf{x} \sim D} \sum_t \log \frac{\exp(\mathbf{e}_t^T \hat{\mathbf{e}}_t)}{\exp(\mathbf{e}_t^T \hat{\mathbf{e}}_t) + \sum_{j \in \text{neg}(t)} \exp\left(\mathbf{e}_j^T \hat{\mathbf{e}}_t\right)}$ and $L_{\text{cross}} = -E_{(\mathbf{x}, \mathbf{y}) \sim D} \log$

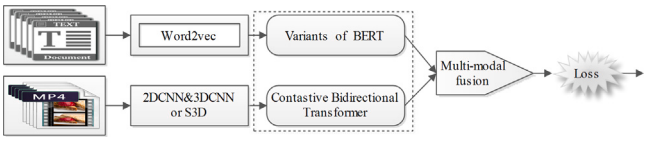

Fig. 14. A multi-modal video-text fusion model to produce explanation.

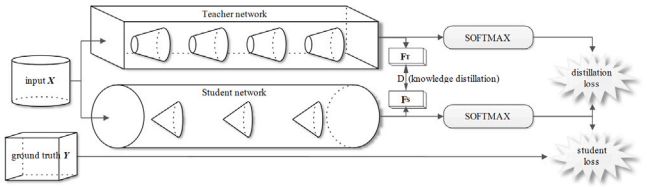

Fig. 15. A knowledge distillation structure.

$\frac{f\left(\mathbf{h}^{xy}_{1:T+T'}\right)}{f\left(\mathbf{h}^{xy}_{1:T+T'}\right)+\sum_{y'\in \mathrm{Neg}(y)} f\left(\mathbf{h}^{xy'}_{1:T+T'}\right)}$. In video feature processing, we adopt the local and global object relationship graph model (Zhang et al., 2020b). The object relationship graph model uses GCN to build a relationship graph network between objects, and enriches the visual representation of the video through the process of relational reasoning. The local relationship graph is used to model the relationships among objects from a single frame. The aim of the global relationship graph is to associate the objects in the entire video, so as to obtain long-term dependencies between objects.

In generating text descriptions of videos, feature extraction pre-training can use complex as well as simple methods. We will use the knowledge distillation method based on them, whose structure is shown in Fig. 15. It mainly uses the joint learning way between *Teacher* network and *Student* network (Tang & Wang, 2018; Zhang, 2020), where the *Teacher* and the *Student* share the feature embedding layer. For the *Teacher* network, it follows the normal training process that uses cross-entropy as its loss function. For the *Student* network, the loss function consists of two parts: one sub-item is cross entropy, which prompts the *Student* network to fit the training data. The other sub-item forces the *Logits* output of the *Student* to fit the *Logits* output of the *Teacher*. The so-called distillation is reflected in the second sub-item.

$$\pounds = H(y, f(x)) + \lambda \cdot \left\| Lgt_T - Lgt_S \right\|^2 \tag{8}$$

where $H(*)$ presents the cross-entropy loss function, $f(x)$ denotes the mapping function of the *Student* network and y means the ground truth. Note that Lgt_T and Lgt_S represent the *Logits* of *Teacher* and *Student* networks, respectively. Here λ is used to adjust the influence of distillation loss. In this way, the *Teacher* network can enhance the model generalization ability of the *Student* network. Distillation can be used here in the dashed box in Fig. 14. When performing distillation, we adopt deep self-attention distillation (Wang et al., 2020d).

4. Experiments

In this section, we describe the experiment in detail. Our work conducted extensive experiments on a real-world dataset to validate our proposed methods. The experiments are carried out from two aspects, the performance of recipe recommendation and the quality of interpretation generated to explain recommended recipes. Specifically, three research questions will be answered as follows:

- **RQ1:** Does RcpMKR outperform the recommendation performance of existing methods?
- **RQ2:** Can the methods adopted or fused generate more reasonable explanations compared with baseline methods?
- **RQ3:** How is the satisfaction of recommendation performance and generated interpretations?

Table 1
Details of dataset collected for this work.

Dataset	Text	Image	Video
# users	54,026		
# recipes	330,102		
# interactions	814,928		
# entities	141,800	26,057	5309
# entity types	21		
# relation types	40		
# triplets	414,602		

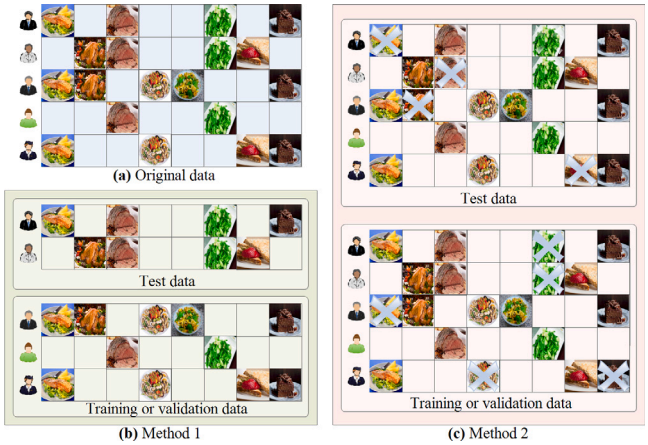

Fig. 16. Two ways of using data in the experiments.

4.1. Datasets and data split

The data used in these experiments mainly comes from two parts: popular recipe websites and published articles such as Zhang et al. (2020b). These websites mainly include allrecipes.com, netflix.com, food.com, yummly.com, finecooking.com, meishij.net, and Recipe1M.[2] For health-related data, we mainly use common medical knowledge and some recognized experience. There are three types of data we crawled, text, image, and video formats. The first three rows in Table 1 are mainly for recommendations, and the following four rows are for building a knowledge graph based on user requirements. The user's interactions are mainly reflected by their historical behaviors, such as user play, like, favorite, follow, share, comment, etc.

In traditional machine learning, we split the original data according to Fig. 16(b) to create a training set, a validation set and a test set. However, this will not work for the recommendation model if we train the data on a single user group and verify other data on another user group. For the recommender system, we randomly mask some known recipe elements in the data matrix, as shown in Fig. 16(c) (Parul, 2019). Then we use recommendation algorithms to predict these masked elements. Finally, we compare the predicted elements with the actual recipes to see if they are consistent. In order to have more credibility, we will perform six runs to report their means as our final experimental results.

4.2. Evaluation metric

The pros and cons of a recommendation algorithm can be directly reflected through evaluation indicators. Generally speaking, according to the different recommendation tasks, the most commonly used metrics can be divided into three categories: the evaluation of the predicted score, which is suitable for scoring the prediction task; the evaluation

[2] http://im2recipe.csail.mit.edu/dataset.

of the predicted item set, which is suited to the Top-n recommendation task (generally, $n = 10$); the evaluation of the recommendation effect weighting based on a rank list, which is fit for the above both recommendation tasks. In this paper, we will use three commonly used metrics, **precision**, **recall** and **F1** to evaluate the recommendation performance. There is currently no unified evaluation indexes to evaluate the quality of generation interpretation. Here we follow the previous related work (Zhao et al., 2020), and indirectly evaluate interpretation generated by using the above three metrics. For example, according to the explanation generated for an image, we compare the key fields in the generated explanation with the ground truth. Its idea is similar to the verification of recommendation performance, i.e., matching the key fields in the generated explanation to recommended items. Using the above three indicators to evaluate the generation of interpretation may be failed if such a situation occurs: the generated text is very different from the ground truth, but their meanings are very close, or the generated interpretation is very similar to the ground truth, but their meanings are vastly different. Therefore, we will use **similarity** to verify models (Vedula et al., 2020; Zhou et al., 2020b). Specifically, the method is to convert the generated interpretation into a vector and compare it with the vector corresponding to the ground truth.

On the other hand, the best way to evaluate any recommender systems is to test them in the real world. Techniques like A/B testing (Feng et al., 2020; Wang et al., 2020b) are the best because you can get actual feedback from real users. In order to verify our model more realistically and conveniently, we invite real users to give us the satisfaction of recommendation performance and generated explanations (Mao et al., 2016). Satisfaction will be reflected by MOS, which is divided into 5 levels. 1 score means very dissatisfied, and 2 score presents slightly dissatisfied. 3 score denotes average satisfied and 5 score represents completely satisfied. 4 score means relatively satisfied, which is between the 3 and 4 scores.

4.3. Experiment setup

In order to verify the effectiveness of our methods, we compare with a number of relatively new baselines. For the recommended performance, we compare with libFM (Rendle, 2012), CKE (Zhang et al., 2016), RippleNet (Wang et al., 2018a) and KGAT (Wang et al., 2019a). For the generation of explanation, we compare with LXMERT (Tan & Bansal, 2019), End2End (Zhou et al., 2018) and CBT (Sun et al., 2019a). In the recommender system, users and items are often represented by low-dimensional vectors, and the number of items and users can often reach hundreds of millions. This causes the parameters of the underlying embedding layer to be the majority of the parameters in the network, and even over-fitting. Most of the researchers generally use regularization techniques, including *L1* and *L2* regularization, *dropout*, and so on. In this paper, we use stochastic shared embedding (SSE) (Wu et al., 2019) to overcome over-fitting. SSE can be well combined with these algorithms of SGD, that is, a SSE layer can be introduced after the embedding layer. It can be expressed as $\sum_i \sum_{k \in I} P(j^i, k|\Phi) l_{oss}(E[k]|\Theta)$. Here i represents the ith training sample, and j and k are the indexes of embedding. Note that Φ is the replacement probability parameter between two embeddings. P denotes the probability of replacing embedding j with embedding k, and $E[k]$ represents the operation of taking out embedding k. It is worth noting that two embeddings are randomly replaced during training, and the replacement operation is turned off during validating and testing.

4.4. Results and analysis

4.4.1. Recommendation performance (to RQ1 & RQ3)

The RcpMKR is composed of multiple modules. We have generated different combinations of these modules and compare their results. The knowledge graph module is either implemented as a basic interactive KG (**KG**) or the more detailed variant called demand-based recipe knowledge graph (**RcpKG**). Similarly, the feature extraction component is either implemented as **GNN** or **GCN**. Finally, we tested our recommender system with and without taking user concerns (**Cons**) into consideration. For example, RcpMKR+KG+GNN shows that the recommender system uses a basic knowledge graph and GNN, while RcpMKR+RcpKG+Cons+GNN shows that the recommender system uses a demand-based knowledge graph as well as takes user concerns into as input. The abbreviations of other methods are similar to them. The specific experimental results are shown in Table 2. **Bold** represents the best performance. Underline represents the second-best result. Dashedunderline indicates that the same results are happening. Their average results are shown in **Average** row and ↑↑↑ represents the percentage that our proposed method improves compared with the best baseline.

Table 2
Results comparison of recommended performance on three evaluation metrics.

Methods	Precision	Recall	F1
libFM (Rendle, 2012)	0.0772	0.0603	0.0677
CKE (Zhang et al., 2016)	0.0796	0.0629	0.0703
RippleNet (Wang et al., 2018a)	0.0822	0.0656	0.0730
KGAT (Wang et al., 2019a)	0.0831	0.0672	0.0743
RcpMKR+KG+GNN	0.0829	0.0658	0.0734
RcpMKR+RcpKG+GNN	0.0866	0.0673	0.0757
RcpMKR+RcpKG+Cons+GNN	0.0887	0.0694	0.0779
RcpMKR+KG+GCN	0.0832	0.0657	0.0734
RcpMKR+RcpKG+GCN	0.0865	0.0686	0.0765
RcpMKR+RcpKG+Cons+GCN	**0.0896**	**0.0703**	**0.0788**
Average	0.0840	0.0663	0.0741
↑↑↑ (%)	+7.82%	+4.61%	+6.06%

Table 2 gives experimental results of different methods on the recommended performance. It can be seen that the best results are achieved in our proposed method. Specifically, RcpMKR+RcpKG+Cons+GCN achieved 0.0896, 0.0703, 0.0788 on precision, recall and F1, respectively. The results show that the proposed approach improves precision by 7.82% as compared to the best performing. From the proposed methods we listed, only the method RcpMKR-KG+GNN did not outperform KGAT, while the other methods surpassed it, respectively. It shows that using the demand-based knowledge graph we constructed is of great value. In other words, its introduction greatly improves the recommendation performance. Judging from the methods of adding concerns, the performance has improved to different degrees, which shows that feature extraction about concerns is effective in our proposed method. In addition, the use of GNN and GCN also showed different effects on the results. Overall, methods using GCN perform better than methods using GNN. It may be that our methods are based on the graph structure. When extracting information from the graph, GCN is indeed stronger than general GNN. The result of the best combination method our proposed is much higher than the average result. It can also reflect the feasibility of our proposed method from the side.

Fig. 17 shows the satisfaction of different recommendation methods in terms of performance. Fig. 17(a) is the specific result of individual MOS satisfaction for each method. It can be seen that the broken lines of 1-score and 2-score are downward trends, while 3-score, 4-score, and 5-score are upward trends. 1-score and 2-score indicate the percentage of dissatisfaction with the recommended results. The lower the value, the fewer the proportion of dissatisfaction, i.e., the fewer the number of objections to the method. Similarly, the higher the value of 3-score, 4-score and 5-score, the greater the proportion of satisfaction, that is, the greater the degree of acceptance of the method. Overall, the proposed approaches have achieved a higher level of user satisfaction than the baseline approaches. For example, all our proposed methods are higher than the reference methods from the 3-score broken line. In order to show the superiority of our proposed method more clearly, we give the overall satisfaction for each method, as shown in Fig. 17(b). The

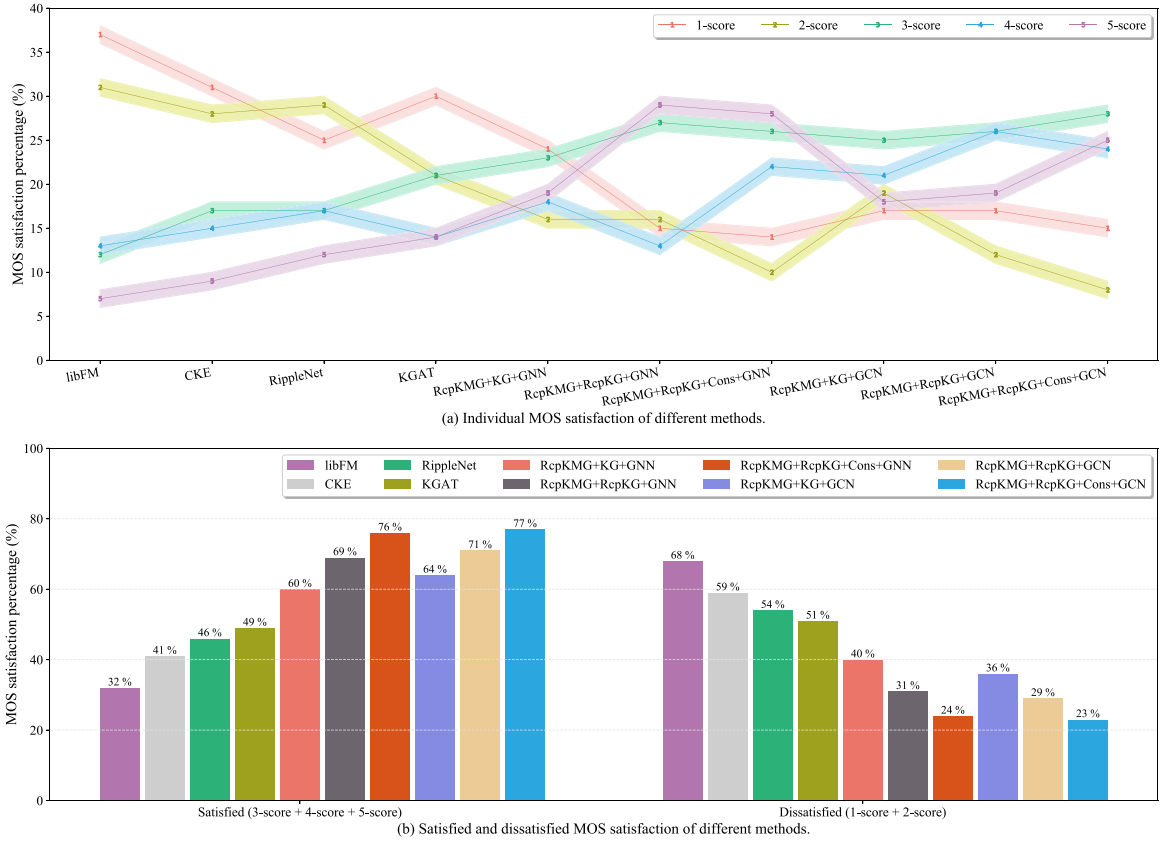

Fig. 17. Satisfaction comparison of different methods for recommended performance.

left is the sum of satisfaction (2-score, 3-score and 4-score) for each method, and the right is the sum of dissatisfaction (1-score and 2-score) for each method. It can be clearly seen from the left part that all the proposed methods have achieved a higher satisfaction level than the baseline methods. Specifically, the method of introducing the demand-based knowledge graph is indeed much better than using ordinary interactive data. Furthermore, the method of considering concerns is obviously higher than the method of not introducing Cons. On the other hand, we can also conclude that the difference in satisfaction between the method using GCN and the method using GNN is very small, which may be that there are a lot of the same recommendation results in both methods. In general, the satisfaction result of using GCN is slightly higher than that of GNN, and this result is consistent with the conclusion drawn from Table 2. In the same way, the same conclusion can be given from Fig. 17(b) on the right. For example, the dissatisfaction value of our proposed methods, especially the RcpMKR+RcpKG+Cons+GNN and the RcpMKR+RcpKG+Cons+GCN, is less than that of the baselines methods, indicating that our proposed methods have received few objections.

4.4.2. Interpretation generation (to RQ2 & RQ3)

Table 3 gives results evaluation of the generated interpretation. On one hand, from the explanation of the Text&Image generation, the methods we applied (except for imgExpGAN) are generally better than the reference method. No single method has the best performance on all evaluation metrics as each method extract a different set of features from the input data. imgExpGAN does not outperform LXMERT method as a shallow network has been used, which may have not extracted enough features. The imgExp2SBERT method achieves the best results on the F1 metric. The best results on other indicators are imgExpRoISBERT+FasterRCNN and imgExpRoISBERT+ResNet-*n*. We can see from the precision and similarity indicators that the

Table 3
Comparison of experimental results for generating explanation.

	Methods	Precision	Recall	F1	Similarity
Text&Image	LXMERT (Tan & Bansal, 2019)	0.1722	0.1247	0.1447	0.6794
	imgExpGAN	0.1483	0.0934	0.1146	0.5731
	imgExp2SBERT	0.1834	0.1249	**0.1486**	0.6819
	imgExpRoIsBERT+CNN	0.1805	0.1193	0.1437	0.6773
	imgExpRoIsBERT+FasterRCNN	0.1831	**0.1251**	**0.1486**	0.6818
	imgExpRoIsBERT+ResNet-*n*	**0.1836**	0.1246	0.1485	**0.6895**
Text&Video	End2End (Zhou et al., 2018)	0.1509	0.0948	0.1164	0.5885
	CBT (Sun et al., 2019a)	0.1520	0.0951	0.1170	0.5873
	viExpCBTBERT+S3D	0.1543	0.0955	0.1180	0.5910
	viExpCBTBERT+2D3DCNN	**0.1608**	**0.1073**	**0.1287**	**0.5981**
	Average	0.1669	0.1105	0.1329	0.6348
	↑↑↑ (%)	+6.62% +5.79%	+0.32% +12.83%	+2.70% +10.00%	+1.49% +1.63%

imgExpRoISBERT+ResNet-*n* method has achieved better results, indicating that it is more suitable for the imgExpRoISBERT framework. In addition, we can see that it is difficult to get better results using ordinary CNN alone in the application of imgExpRoISBERT. Both imgExp2SBERT and imgExpRoISBERT methods have shown good performance. Especially from the perspective of the improvement, the imgExpRoISBERT+ResNet-*n* method increases by 6.62%. Therefore, we can show that the use of both imgExp2SBERT and imgExpRoISBERT is very feasible in our recipe interpretation generation. On the other hand, judging from Text&Video generation interpretation, their results are not as good as using images to generate interpretation. This may be because the amount of sample data we used is relatively small and no more training is performed. Another reason may be that their feature extraction mechanisms are different from the Text&Image methods. But only from the results of the interpretation generated by the video, the method we used is also better than baselines. For example, the

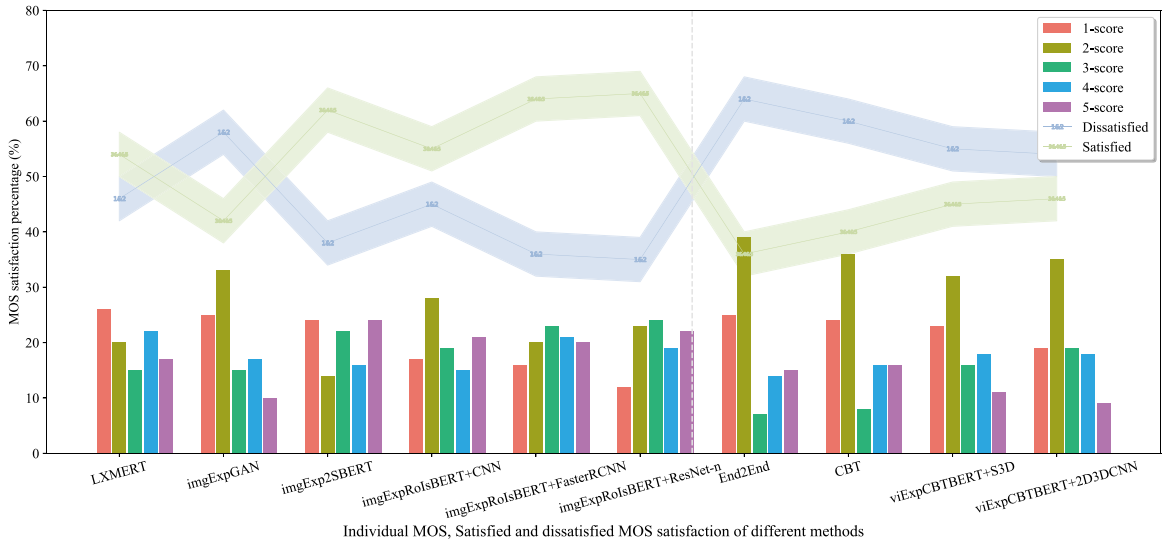

Fig. 18. Satisfaction comparison of different methods for generating interpretation using images or videos.

viExpCBTBERT+2D3DCNN method achieved the best results on four indicators. Even, its recall and F1 indicators have increased by more than 10%, which is not easy to get. Therefore, we can conclude that the method we used is very suitable whether it is using images or using video to generate explanations in the field of recipe recommendation.

Fig. 18 shows user satisfaction levels for explanations generated by each method. It can be seen from the two broken lines that the methods we adopted are better than the baselines method, regardless of image generation interpretation or generating explanation by videos. From the Satisfied broken line, the imgExpGAN method is at the lowest point. The imgExp2SBERT and imgExpRoIsBERT methods are higher than LXMERT. When the general CNN is used alone, the imgExpRoIsBERT method is not as good as the imgExp2SBERT method, which shows that the imgExp2SBERT also has a strong advantage in the interpretation generation. The results of viExpCBTBERT+2D3DCNN and viExpCBTBERT+S3D are very close, but the former is slightly higher. This may be because viExpCBTBERT+2D3DCNN considers deep features from two different perspectives. The conclusion drawn here about satisfaction is basically consistent with Table 3. Similar results can be obtained from the histogram. For example, 1-score can be intuitively seen as a downward trend, which shows that the methods we adopted are indeed supported by more people. In other words, the explanation generated by the framework we used is more interpretable and correct than the reference methods. From the perspective of the 5-score, there is no certain upward trend. This may be because everyone believes that the generated explanation can only be given a 5 score if they are 100% satisfied. (This is impossible because everyone's understanding is different even if they face the same interpretation.) Therefore, compared to the baseline, the method we used in this paper is feasible in terms of recipe interpretation generation.

4.4.3. Ablation discussion

In our proposed approach, we used a combination of different modules constructed by various methods, and employed two important technologies, namely dynamic convolution (DynConv) and random synthesizer attention (RanSynAtt). In order to verify the rationality of the two methods, we give an ablation discussion of related experiments as shown in Table 4. From the results of recommended performance, the methods using ordinary attention are worse than those using the RanSynAtt on three indicators. Furthermore, the way of using DynConv is better than that of regular convolution. On one hand, in the case of using ordinary convolution, adopting RanSynAtt has achieved slightly better results than ordinary attention mechanism for most of the method. Except that the precision of using RanSynAtt is the same as using regular attention in the RcpMKR+RcpKG+GCN method, most of the other methods using RanSynAtt are slightly higher than them using ordinary attention. On average, we can get the same conclusion on recall and F1 indexes, that is, the methods using RanSynAtt are better than those using ordinary attention. In the case of DynConv, the gap between using RanSynAtt and using ordinary attention is more obvious. For example, the average values of those methods using the RanSynAtt on three indicators are 0.0863, 0.0679 and 0.0760. Specifically, the precision of the RcpMKR+RcpKG+Cons+GCN method via RanSynAtt reaches 0.0896, while its precision via ordinary attention is only 0.0891. On the other hand, when the RanSynAtt is used, the result of using DynConv is higher than that of the ordinary convolution. From the F1 index, the result 0.0734 of RcpMKR+KG+GCN via DynConv is significantly higher than the result 0.0729 of using ordinary convolution. When general attention is used, although some values do not have good results on certain metrics, most of the results of using DynConv are still higher than those of using ordinary convolution. Therefore, it can be said that the application of dynamic convolution technology is effective in our proposed methods, which may be due to them involving multiple convolution operations. In addition, we can also conclude that using random synthesizer attention technology also has a positive effect on our proposed methods because they involve different types of attention.

From the results of the generated interpretation, we can also get the same conclusion, that is, the method of using dynamic convolution is better than that of ordinary convolution, and the result of using random synthesizer attention is better than that of using ordinary attention. It is worth pointing out that the imgExpGAN method is basically ineffective when the two technologies are used. This may be because the feature extraction or other operations in the imgExpGAN method do not involve convolution and attention mechanisms. From the perspective of the Text&Image methods, the results obtained by using dynamic convolution are slightly better than using ordinary convolution. For example, the similarity 0.6895 of imgExpRoIsBERT+ResNet-*n* with dynamic convolution is higher than the result 0.6890 obtained by using ordinary convolution. Moreover, its 0.6890 with random synthesizer attention is higher than its 0.6884 with regular attention. From the Text&Video methods, we can also get similar conclusions. Although the results of some methods are not consistent, most of them can draw the conclusions we stated. Therefore, the use of dynamic convolution and random synthesizer attention techniques is a successful application in improving the recipe recommendation performance and generating reasonable explanations.

Table 4
Comparison results of different methods on recommendation performance and generation interpretation through DynConv and RanSynAtt.

Conv.	Att.	Recommendation performance				Explanation generation				
		Methods	Precision	Recall	F1	Methods	Precision	Recall	F1	Similarity
Ordinary Convolution	General Attention	RcpMKR+KG+GNN	0.0826	0.0649	0.0727	imgExpGAN	0.1482	0.0929	0.1142	0.5729
		RcpMKR+RcpKG+GNN	0.0857	0.0683	0.0760	imgExp2SBERT	0.1828	0.1242	0.1479	0.6883
		RcpMKR+RcpKG+Cons+GNN	0.0883	0.0696	0.0778	imgExpRoIsBERT+CNN	0.1791	0.1182	0.1424	0.6763
		RcpMKR+KG+GCN	0.0827	0.0649	0.0727	imgExpRoIsBERT+FasterRCNN	0.1826	0.1245	0.1481	0.6808
		RcpMKR+RcpKG+GCN	0.0858	0.0683	0.0761	imgExpRoIsBERT+ResNet-*n*	0.1825	0.1242	0.1478	0.6884
		RcpMKR+RcpKG+Cons+GCN	0.0884	0.0698	0.0780	viExpCBTBERT+S3D	0.1532	0.0940	0.1165	0.5901
		Average	0.0856	0.0676	0.0756	viExpCBTBERT+2D3DCNN	0.1601	0.1065	0.1279	0.5976
						Average	0.1698	0.1121	0.1350	0.6421
	RanSynAtt	RcpMKR+KG+GNN	0.0827	0.0649	0.0727	imgExpGAN	0.1482	0.0934	0.1146	0.5731
		RcpMKR+RcpKG+GNN	0.0856	0.0682	0.0759	imgExp2SBERT	0.1829	0.1242	0.1479	0.6884
		RcpMKR+RcpKG+Cons+GNN	0.0885	0.0699	0.0781	imgExpRoIsBERT+CNN	0.1795	0.1184	0.1427	0.6767
		RcpMKR+KG+GCN	0.0829	0.0650	0.0729	imgExpRoIsBERT+FasterRCNN	0.1826	0.1247	0.1482	0.6810
		RcpMKR+RcpKG+GCN	0.0858	0.0684	0.0761	imgExpRoIsBERT+ResNet-*n*	0.1832	0.1242	0.1480	0.6890
		RcpMKR+RcpKG+Cons+GCN	0.0886	0.0698	0.0781	viExpCBTBERT+S3D	0.1533	0.0946	0.1170	0.5907
		Average	0.0857	0.0677	0.0756	viExpCBTBERT+2D3DCNN	0.1603	0.1070	0.1283	0.5978
						Average	0.1700	0.1124	0.1353	0.6424
Dynamic Convolution	General Attention	RcpMKR+KG+GNN	0.0828	0.0650	0.0728	imgExpGAN	0.1482	0.0930	0.1143	0.5730
		RcpMKR+RcpKG+GNN	0.0856	0.0679	0.0757	imgExp2SBERT	0.1832	0.1246	0.1483	0.6885
		RcpMKR+RcpKG+Cons+GNN	0.0881	0.0693	0.0776	imgExpRoIsBERT+CNN	0.1801	0.1190	0.1433	0.6771
		RcpMKR+KG+GCN	0.0828	0.0652	0.0730	imgExpRoIsBERT+FasterRCNN	0.1828	0.1250	0.1485	0.6813
		RcpMKR+RcpKG+GCN	0.0861	0.0684	0.0762	imgExpRoIsBERT+ResNet-*n*	0.1833	0.1244	0.1482	0.6893
		RcpMKR+RcpKG+Cons+GCN	0.0891	0.0701	0.0785	viExpCBTBERT+S3D	0.1541	0.0950	0.1175	0.5910
		Average	0.0858	0.0677	0.0756	viExpCBTBERT+2D3DCNN	0.1606	0.1070	0.1284	0.5980
						Average	0.1703	0.1126	0.1356	0.6426
	RanSynAtt	RcpMKR+KG+GNN	0.0829	0.0658	0.0734	imgExpGAN	0.1483	0.0934	0.1146	0.5731
		RcpMKR+RcpKG+GNN	0.0866	0.0673	0.0757	imgExp2SBERT	0.1834	0.1249	0.1486	0.6819
		RcpMKR+RcpKG+Cons+GNN	0.0887	0.0694	0.0779	imgExpRoIsBERT+CNN	0.1805	0.1193	0.1437	0.6773
		RcpMKR+KG+GCN	0.0832	0.0657	0.0734	imgExpRoIsBERT+FasterRCNN	0.1831	0.1251	0.1486	0.6818
		RcpMKR+RcpKG+GCN	0.0865	0.0686	0.0765	imgExpRoIsBERT+ResNet-*n*	0.1836	0.1246	0.1485	0.6895
		RcpMKR+RcpKG+Cons+GCN	0.0896	0.0703	0.0788	viExpCBTBERT+S3D	0.1543	0.0955	0.1180	0.5910
		Average	0.0863	0.0679	0.0760	viExpCBTBERT+2D3DCNN	0.1608	0.1073	0.1287	0.5981
						Average	0.1706	0.1129	0.1358	0.6418

5. Conclusion and future work

The emergence of enormous multimedia data and the diversification of user needs have increased the difficulty of successfully suggesting recipes in recipe recommender systems. In particular, user's implicit demands may cause a sense of distrust towards the recommender system in the absence of a reasonable explanation for suggested recipes. In this work, we proposed an extensible multi-modal recipe fusion framework, which provides guidelines for improving performance of recipe recommendation and generating reasonable and acceptable explanations derived from images or videos related to the recommended results. In order to address the multi-interest and the diversity of user needs, we first adopt the combination of multi-modality and hierarchical ideas to construct a recipe knowledge graph centered on considering multiple factors. It not only takes into account the underlying demands of users, but also mines the in-depth relationship among users, among recipes and among users and recipes. Then, we proposed a novel multi-modal recipe recommendation method based on node representation of multiple aspects and multi-relational graph structure extraction via demand-based knowledge graph. For generating interpretation, we applied multiple BERT-based multi-modal fusion models and generative adversarial networks. Moreover, the use of dynamic convolution and random synthesizer attention further improved the performance. The experimental results on real data set show that the work done in this paper is feasible and effective in the field of recipe recommendation. When performing multi-modal fusions, we did not overthink the effectiveness of their complementary information, which will be the direction for our work in the future.

CRediT authorship contribution statement

Zhenfeng Lei: Work concept or design, Data collection, Drawing, Experiments, Draft paper, Making important changes to the paper. **Anwar Ul Haq:** Work concept or design, Drawing, Experiments, Draft paper, Making important changes to the paper. **Adnan Zeb:** Draft paper, Making important changes to the paper. **Md Suzauddola:** Drawing, Making important changes to the paper. **Defu Zhang:** Work concept or design, Device support, Making important changes to the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is partially supported by the National Nature Science Foundation of China under Grant 61672439.

References

Adaji, I., Sharmaine, C., Debrowney, S., Oyibo, K., & Vassileva, J. (2018). Personality based recipe recommendation using recipe network graphs. In *Proceedings of the international conference on social computing and social media* (pp. 161–170).

Berg, R. v. d., Kipf, T. N., & Welling, M. (2017). Graph convolutional matrix completion. arXiv preprint arXiv:1706.02263.

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., & Choi, Y. (2019). COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th conference of the association for computational linguistics* (pp. 4762–4779).

Chari, S., Gruen, D. M., Seneviratne, O., & McGuinness, D. L. (2020). Directions for explainable knowledge-enabled systems. arXiv preprint arXiv:2003.07523.

Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., & Liu, Z. (2020). Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 11030–11039).

Chen, M., Jia, X., Gorbonos, E., Hong, C. T., Yu, X., & Liu, Y. (2019). Eating healthier: Exploring nutrition information for healthier recipe recommendation. *Information Processing & Management*, Article 102051.

Chen, H., Yin, H., Sun, X., Chen, T., Gabrys, B., & Musial, K. (2020). Multi-level graph convolutional networks for cross-platform anchor link prediction. arXiv preprint arXiv:2006.01963.

Cheng, K., Wang, N., Shi, W., & Zhan, Y. (2020). Research advances in the interpretability of deep learning. *Journal of Computer Researchand Development*, *57*(6), 1208–1217.

Dong, M., Yuan, F., Yao, L., Wang, X., Xu, X., & Zhu, L. (2020). Trust in recommender systems: A deep learning perspective. arXiv preprint arXiv:2004.03774.

Fadhil, A. (2018). Can a chatbot determine my diet?: Addressing challenges of chatbot application for meal recommendation. arXiv preprint arXiv:1802.09100.

Feng, Y., Hu, B., Lv, F., Liu, Q., Zhang, Z., & Ou, W. (2020). ATBRG: Adaptive target-behavior relational graph network for effective recommendation. arXiv preprint arXiv:2005.12002.

Gao, X., Feng, F., He, X., Huang, H., Guan, X., Feng, C., Ming, Z., & Chua, T.-S. (2019). Hierarchical attention network for visually-aware food recommendation. *IEEE Transactions on Multimedia*, *22*(6), 1647–1659.

Gong, Y., Zhu, Y., Duan, L., Liu, Q., Guan, Z., Sun, F., Ou, W., & Zhu, K. Q. (2019). Exact-k recommendation via maximal clique optimization. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 617–626).

Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., & Yang, G. (2019). XAI - Explainable artificial intelligence. *Science Robotics*, *4*(37).

Guo, W., Wang, J., & Wang, S. (2019). Deep multimodal representation learning: A survey. *IEEE Access*, *7*, 63373–63394.

Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., & He, Q. (2020). A survey on knowledge graph-based recommender systems. arXiv preprint arXiv:2003.00911.

Hassani, K., & Khasahmadi, A. H. (2020). Contrastive multi-view representation learning on graphs. arXiv preprint arXiv:2006.05582.

Jin, Y., Zhang, W., He, X., Wang, X., & Wang, X. (2020). Syndrome-aware herb recommendation with multi-graph convolution network. In *Proceedings of the 36th international conference on data engineering* (pp. 145–156).

Kim, Y., Kim, K., Park, C., & Yu, H. (2019). Sequential and diverse recommendation with long tail. In *Proceedings of the 28th international joint conference on artificial intelligence* (pp. 2740–2746).

Li, L. H., Yatskar, M., Yin, D., Hsieh, C.-J., & Chang, K.-W. (2019). Visualbert: A simple and performant baseline for vision and language. arXiv preprint arXiv:1908.03557.

Liu, G. (2020). Manual control of the recommended system. https://mp.weixin.qq.com.

Liu, Y., Chen, L., He, X., Peng, J., Zheng, Z., & Tang, J. (2020). Modelling high-order social relations for item recommendation. arXiv preprint arXiv:2003.10149.

Lu, J., Batra, D., Parikh, D., & Lee, S. (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proceedings of the advances in neural information processing systems* (pp. 13–23).

Luo, X., Liu, L., Yang, Y., Bo, L., Cao, Y., Wu, J., Li, Q., Yang, K., & Zhu, K. Q. (2020). AliCoCo: Alibaba E-commerce cognitive concept net. In *Proceedings of the 2020 ACM SIGMOD international conference on management of data* (pp. 313–327).

Mao, X., Yuan, S., Xu, W., & Wei, D. (2016). Recipe recommendation considering the flavor of regional cuisines. In *Proceedings of the 2016 international conference on progress in informatics and computing* (pp. 32–36).

Parul, P. (2019). Recommendation systems in the real world. https://towardsdatascience.com.

Paul, C., Jay, A., & Emre, S. (2016). Deep neural networks for YouTube recommendations. In *Proceedings of the 10th ACM conference on recommender systems* (pp. 191–198).

Qiao, Z., Zhou, Y., Yang, D., Zhou, Y., & Wang, W. (2020). SEED: Semantics enhanced encoder-decoder framework for scene text recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 13528–13537).

Rendle, S. (2012). Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *3*(3), 1–22.

Saxena, A., Tripathi, A., & Talukdar, P. (2020). Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 4498–4507).

Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., & Dai, J. (2019). Vl-bert: Pre-training of generic visual-linguistic representations. In *Proceedings of the 8th international conference on learning representations* (pp. 1–16).

Subramaniyaswamy, V., Manogaran, G., Logesh, R., Vijayakumar, V., Chilamkurti, N., Malathi, D., & Senthilselvan, N. (2019). An ontology-driven personalized food recommendation in IoT-based healthcare system. *The Journal of Supercomputing*, *75*(6), 3184–3216.

Sun, C., Baradel, F., Murphy, K., & Schmid, C. (2019). Learning video representations using contrastive bidirectional transformer. arXiv preprint arXiv:1906.05743.

Sun, C., Myers, A., Vondrick, C., Murphy, K., & Schmid, C. (2019). Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE international conference on computer vision* (pp. 7464–7473).

Tan, H., & Bansal, M. (2019). Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 5099–5110).

Tang, J., & Wang, K. (2018). Ranking distillation: Learning compact ranking models with high performance for recommender system. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2289–2298).

Tay, Y., Bahri, D., Metzler, D., Juan, D., Zhao, Z., & Zheng, C. (2020). Synthesizer: Rethinking self-attention in transformer models. arXiv preprint arXiv:2005.00743.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the advances in neural information processing systems* (pp. 5998–6008).

Vedula, N., Lipka, N., Maneriker, P., & Parthasarathy, S. (2020). Open intent extraction from natural language interactions. In *Proceedings of the web conference 2020* (pp. 2009–2020).

Wang, P., Fan, Y., Xia, L., Zhao, W. X., Niu, S., & Huang, J. (2020). KERL: A knowledge-guided reinforcement learning model for sequential recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 209–218).

Wang, X., He, X., Cao, Y., Liu, M., & Chua, T.-S. (2019). Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 950–958).

Wang, X., He, X., Wang, M., Feng, F., & Chua, T.-S. (2019). Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 165–174).

Wang, M., Lin, Y., Lin, G., Yang, K., & Wu, X.-m. (2020). M2GRL: A multi-task multi-view graph representation learning framework for web-scale recommender systems. arXiv preprint arXiv:2005.10110.

Wang, X., Wang, R., Shi, C., Song, G., & Li, Q. (2020). Multi-component graph convolutional collaborative filtering. In *Proceedings of the 34th innovative applications of artificial intelligence conference* (pp. 6267–6274).

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. arXiv preprint arXiv:2002.10957.

Wang, X., Xu, Y., He, X., Cao, Y., Wang, M., & Chua, T.-S. (2020). Reinforced negative sampling over knowledge graph for recommendation. In *Proceedings of the web conference 2020* (pp. 99–109).

Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., & Guo, M. (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* (pp. 417–426).

Wang, H., Zhang, F., Xie, X., & Guo, M. (2018). DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference* (pp. 1835–1844).

Wu, J., He, X., Wang, X., Wang, Q., Chen, W., Lian, J., Xie, X., & Zhang, Y. (2020). Graph convolution machine for context-aware recommender system. arXiv preprint arXiv:2001.11402.

Wu, L., Li, S., Hsieh, C.-J., & Sharpnack, J. L. (2019). Stochastic shared embeddings: Data-driven regularization of embedding layers. In *Proceedings of the advances in neural information processing systems* (pp. 24–34).

Xu, W., He, H., Tan, M., Li, Y., Lang, J., & Guo, D. (2020). Deep interest with hierarchical attention network for click-through rate prediction. arXiv preprint arXiv:2005.12981.

Xu, Q., Shen, F., Liu, L., & Shen, H. T. (2018). Graphcar: Content-aware multimedia recommendation with graph autoencoder. In *Proceedings of the 41st international ACM SIGIR conference on research & development in information retrieval* (pp. 981–984).

Zhang, J. (2020). Application of knowledge distillation in recommendation system. https://zhuanlan.zhihu.com.

Zhang, J., Bai, B., Lin, Y., Liang, J., Bai, K., & Wang, F. (2020). General-purpose user embeddings based on mobile app usage. arXiv preprint arXiv:2005.13303.

Zhang, Z., Shi, Y., Yuan, C., Li, B., Wang, P., Hu, W., & Zha, Z.-J. (2020). Object relational graph with teacher-recommended learning for video captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 13278–13288).

Zhang, F., Yuan, N. J., Lian, D., Xie, X., & Ma, W.-Y. (2016). Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 353–362).

Zhao, K., Wang, X., Zhang, Y., Zhao, L., Liu, Z., Xing, C., & Xie, X. (2020). Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 239–248).

Zhou, B., Cui, Q., Wei, X.-S., & Chen, Z.-M. (2020). BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 9719–9728).

Zhou, Y., Mishra, S., Verma, M., Bhamidipati, N., & Wang, W. (2020). Recommending themes for ad creative design via visual-linguistic representations. In *Proceedings of the web conference 2020* (pp. 2521–2527).

Zhou, L., Zhou, Y., Corso, J. J., Socher, R., & Xiong, C. (2018). End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8739–8748).