processes

MDPI

Article

A Hybrid Recommendation Approach for Medical Services That Incorporates Knowledge Graphs

Chao Ma [1,*], Qi An [1], Zhenguo Yang [1], Hongguo Zhang [1] and Jiaxing Qu [2]

[1] School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China; anan19971109@163.com (Q.A.); yangzhenguo852@163.com (Z.Y.); zhg07@163.com (H.Z.)
[2] Heilongjiang Province Cyberspace Research Center, Harbin 150090, China; smilingqu@126.com
* Correspondence: machao8396@163.com

Abstract: At present, there are a large number of growing medical applications in the application market. It is difficult for users to find satisfactory medical services conveniently and efficiently. The classical collaborative filtering algorithm has some problems, such as cold start, unsatisfactory recommendation results, and so on. This paper proposes a hybrid medical service recommendation approach based on knowledge graph to solve the above problems. This approach introduces the open knowledge graph and establishes the semantic link relationship between the mobile application and the knowledge graph entity. It not only enhances the semantic feature of single application for improving the accuracy of recommendation results, but also realizes the in-depth analysis of the semantic relationship among multiple application entities in the knowledge graph through the TransHR model which can alleviate the cold start problem. Then we design a hybrid recommendation algorithm based on multi-dimensional similarity fusion. This algorithm uses the entropy method to organically integrate the calculation results of multi-dimensional semantic similarity, such as feature vector similarity, entity relation similarity, and user rating similarity. It is convenient and efficient to recommend satisfactory medical application services to target users. Finally, we test and analyze the accuracy and effectiveness of our proposed approach by experiment.

Keywords: hybrid recommendation; knowledge graph; mobile application feature; multi-dimensional similarity

Citation: Ma, C.; An, Q.; Yang, Z.; Zhang, H.; Qu, J. A Hybrid Recommendation Approach for Medical Services That Incorporates Knowledge Graphs. *Processes* **2022**, *10*, 1500. https://doi.org/10.3390/pr10081500

Academic Editor: Lei Wang

Received: 16 June 2022
Accepted: 27 July 2022
Published: 29 July 2022

1. Introduction

At present, mobile Internet technology is developing rapidly, and there is an increasing demand for electronic devices such as smartphones and tablet computers. At the same time, more and more mobile applications appear in the view of users. The users put forward higher requirements for mobile applications in terms of functional and non-functional requirements. So how to recommend satisfactory mobile applications, improve the utilization rate of mobile applications, and reduce the unloading rate of mobile applications is significant for users and mobile application developers. Especially in the field of medical services, due to the shortage of offline medical services, users more and more tend to look for medical applications on the mobile Internet. They acquire medical knowledge through medical applications and purchase its supporting medical diagnostic equipment to obtain continuous health monitoring and medical diagnostic services. However, at present, there are many medical-related mobile applications in the application market with different functions and uneven performance. For the users who lack medical knowledge, how to easily and efficiently obtain the medical services they need is an urgent problem to be solved.

In academia and industry, the recommendation system is one of the best solutions to solve the above problems. The recommendation system can help users accurately obtain

satisfactory application services from hundreds of mobile applications at the minimum time and energy. The recommendation system began with the email recommendation system Tapestry [1] designed and implemented by Goldberg et al. in 1992. The purpose of recommendation system is to find what users want among many products. Classical service recommendation methods have three categories: content-based recommendation (CB) [2,3], collaborative filtering (CF) [4–6], and hybrid recommendation method [7]. The CF is the earliest and the most famous recommendation algorithm. It is also one of the most widely used and mature recommendation technologies [8,9]. Collaborative filtering algorithms have two types: user-based collaborative filtering algorithms and item-based collaborative filtering algorithms. The classic collaborative filtering carries out recommendation by using similar users or items. However, the new applications often have very few user ratings or none, or the new users do not rate any applications or only rate few applications. This phenomenon will lead to cold start problems, which makes the recommendation unable to be carried out or the recommendation results unavailable.

In recent years, many scholars have conducted in-depth research to solve the cold start problem. Tan et al. [10] first proposed the similarity measure method based on the principle of physical resonance. This method alleviates the data sparsity and cold start problems of collaborative filtering recommendation system. However, this method mainly improves the calculation of similarity, it does not consider the context information in service recommendations. Hu et al. [11] incorporated time information into similarity calculation and service quality prediction. They designed a hybrid personalized random walk algorithm to infer the similarity between indirect users and services. This method reduces the problem of data sparsity and alleviates the cold start problem. Zhang et al. [12] mine neighbor users of social networks by user embedding. They design a two-level model based on the Markov chain. This method dynamically models user preferences to alleviate the cold start problem of implicit feedback recommendation systems.

In addition to the cold start problem mentioned above, another problem of the classical collaborative filtering recommendation method is the unideal accuracy of the recommendation results. For this reason, some scholars try to combine knowledge graphs with service recommendations, which emerged as a recommended method based on knowledge graphs. Wang et al. [13] combined knowledge graphs with user-item graphs and proposed a knowledge graph attention network. This method explicitly models the high-order connectivity in knowledge graphs in an end-to-end way and improves the accuracy of service recommendations based on the knowledge graph. Zhang et al. [14] constructed a travel spatio-temporal knowledge graph and modeled the scenic spot feature and spatio-temporal semantics in the knowledge graph. Finally, it recommends suitable tourist attractions for the target users. This method improves the accuracy of recommendations based on the knowledge graph. Wang et al. [15] use Microsoft satori to construct a news-related knowledge graph. They extract the structural information from the knowledge graph through representation learning and replace the corresponding entities in the news headlines. In this way, news recommendations can find more connections at the knowledge level, mine potential connections between news, and provide users with personalized news recommendations. From the above research, we found that the current methods to alleviate the cold start problem are to improve the collaborative filtering algorithm itself. The application of knowledge graphs in the field of service recommendation is only for service recommendation, and it is rarely combined with collaborative filtering algorithms.

To sum up, on the one hand, there are a large number of growing medical applications in the application market. It is impossible for users to find satisfactory mobile applications conveniently and efficiently to obtain much-needed medical services. This is a hot issue that users, medical devices, and application developers are concerned about. On the other hand, the classical Collaborative Filtering algorithm has some problems such as cold start and unsatisfactory recommendation results, which urgently need to be improved. Although the recommendation method based on knowledge graph is rising gradually, and there are commercial knowledge graphs in medical fields such as Baidu Lingyi (Baidu

Lingyi: https://01.baidu.com (accessed on 19 July 2022)), Ping an Intelligent Medical (Ping an Intelligent Medical: https://yun.pingan.com/ssr/solutions/medical (accessed on 19 July 2022)), and Ali Medical knowledge Graph (Ali Medical knowledge Graph: https://www.doctoryou.ai (accessed on 19 July 2022)), the open medical knowledge graph is still difficult to obtain.

This paper proposes a medical service hybrid recommendation method based on Knowledge Graph to solve the above problems. This method is based on the application feature set obtained from the mobile application market through natural language processing technology. First, we introduce the open knowledge graph and establish the semantic link relationship between the mobile application and the knowledge graph entity. It not only enhances the single application semantics feature and improves the accuracy, but also realizes the in-depth analysis of the semantics relationship among multiple application entities in the knowledge graph through the TransHR model which can alleviate the cold start problem. Then, we design a hybrid recommendation algorithm based on multi-dimensional similarity fusion. This algorithm uses the entropy method to organically integrate the calculation results of multi-dimensional semantic similarity, such as feature vector similarity, entity relation similarity, and user rating similarity. It is convenient and efficient to recommend satisfactory medical application services to target users. Finally, we test and analyze the accuracy and effectiveness of this method by experiment.

The main contributions of this paper are as follows: (1) we establishes the semantic link relationship between the knowledge graph and the mobile application feature set. On the one hand, it expands the semantic features of a single mobile application. On the other hand, it embeds the semantics feature into the knowledge graph, which can supplement the relationship semantics between multiple mobile applications, improve the semantic representation ability of the mobile application feature set as a whole, and improve the final recommendation results. (2) We use the entropy method to organically integrate the calculation results of multi-dimensional semantic similarity, such as application feature vector similarity, knowledge graph entity relation similarity, and user application rating matrix similarity. It is convenient and efficient to recommend satisfactory medical application services to target users. Finally, we test and analyze the accuracy and effectiveness of this method by experiments.

The main structure of this paper is as follows: (1) in the second section of this paper, we briefly introduce the knowledge graph and describe the recommendation method based on the knowledge graph. (2) In the third section of this paper, the extracted application feature is fused with the corresponding application entity feature in the knowledge graph to complete the knowledge graph and calculate application similarity from multiple dimensions. We proposed a hybrid similarity calculation method of the fusion knowledge graph, which uses the entropy method to determine the weight of each similarity to recommend. (3) In the fourth section of this paper, we experiment according to the previous recommendation algorithm, which proves the algorithm can improve the cold start problem to a certain extent.

2. Knowledge Graph and Recommendation Method Based on Knowledge Graph

Knowledge graph [16] is a graph-based network data structure, which is composed of multiple nodes and edges. Among them, entities are nodes, and the relationships between entities are edges. In the knowledge base, the knowledge graph is stored in the form of multiple triples of <head entity, relationship, tail entity>, and the triple can be expressed in the form of G = (h, r, t), where h represents the set of head entities in the knowledge graph, h = {h1, h2, . . . }; r represents the set of relations between all entities in the knowledge graph, r = {r1, r2, . . . }; t represents the set of tail entities in the knowledge graph, t = {t1, t2, . . . }.

Recommendation methods based on knowledge graphs can be divided into two categories [17], which are embedding-based methods and path-based methods. In this paper, we mainly use embedding-based methods. The embedding-based methods mainly characterize entities and relationships by the method of graph embedding, so as to realize

the expansion of the original application's characteristic information, and then apply it to the recommendation system. The presentation learning models of the knowledge graph include translation models [18], distance models [19], simple neural network models [20], and energy models [21,22]. The Trans series methods are a typical method of representing entities and relationships in the knowledge graph. These methods embed existing semantic data into a low-dimensional semantic space, and in the low-dimensional space, the similarity between two applications can be observed more intuitively. The Trans series methods are widely used in the representation of knowledge graphs.

The basic idea of the TransE method is to make the sum of the head entity vector h and the relation vector r as close to the tail entity vector t as possible, that is, $h + r \approx t$, as shown in Figure 1. It is assumed here that the L1 or L2 norm is used to measure "closeness" on the semantic level. Therefore, for a triple with a correct relationship, the value of the distance d $(h + r, t)$ should be as small as possible; in contrast, for a triple with a wrong relationship, the value of the distance d $(h + r, t)$ is larger. Therefore, the objective function is shown in Equation (1):

$$L = min \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S} \left[\gamma + d(h+r,t) - d\left(h'+r,t'\right)\right]_+ \quad (1)$$

$$\|h\| \le 1, \|r\| \le 1, \|t\| \le 1$$

where $d = ||h + r - t||$ is the L1 or L2 norm, and γ represents the maximum distance between positive and negative samples, which is a constant.

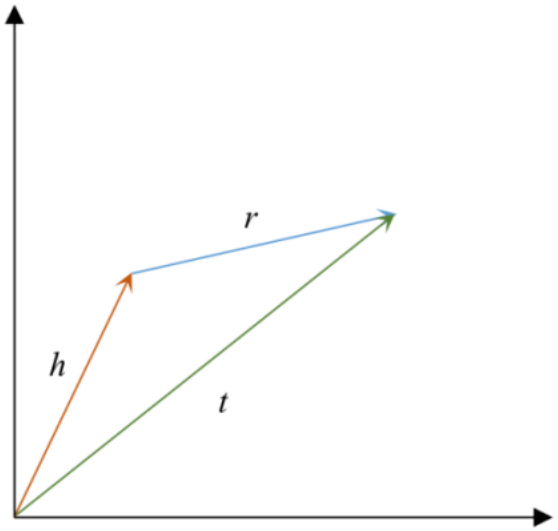

Figure 1. Schematic diagram of atlas embedded TransE method.

3. Hybrid Recommendation Method Fused with Knowledge Graph

3.1. Expansion of Application Feature Sets Based on Knowledge Graph

Under the research background of this paper, the mobile application characteristic information mainly includes application type, release area, application rating, application function classification, application introduction, application update information, etc. The description of the content of each mobile application characteristic information is mainly shown in the following Table 1.

Table 1. Mobile application characteristic information description table.

Information Item	Information Item Description
Application Type	Word, including 16 categories: games, practical tools, audiovisual, chat and social networking, book reading, learning and education, efficient office, fashion shopping, home life, travel and transportation, photography and videography, medical and health, sports, news and information, entertainment, financial management.
Release Area	Word, currently includes five release regions: China, India, Indonesia, Russia, Spain.
Application Rating	Number, reflect the comprehensive evaluation of this application by all users.
Application Function Classification	Words, classification items that conform to the main functions of the application. If the application conforms to multiple classifications, there may be multiple functional classification information.
Application Introduction	Long sentence description, introducing the application functions and highlights, through the application introduction can reflect some of the characteristics of the application, users can understand the general role of the application through this item.
Application Update Information	Long sentence description, reflecting the newly added features of the application.

The above information is all semi-structured information provided by the application market. From the perspective of the service recommendation system, the information in the application type, release area, application rating, and application function classification of the mobile application can be directly used as the feature information of the application after being crawled, and stored in the feature word set of the application. However, content such as application introduction and application update information contains a lot of invalid information, which cannot be directly used as application feature information. Therefore, natural language processing methods must be used to process this type of information to convert it into usable application feature information.

This paper uses the Language Technology Platform (LTP) of the Harbin Institute of Technology (HIT) to process the application introduction and update information. LTP is an open Chinese natural language processing system developed by Social Computing and Information Retrieval (SCIR) of HIT which integrates word segmentation, part-of-speech tagging, named entity recognition, dependency parsing, and other NLP technologies. Word Segmentation (WS) refers to segmenting Chinese character sequences into word sequences. Part-of-speech Tagging (POS) is giving each word in a sentence a part-of-speech category, which includes nouns, verbs, adjectives, or others. Named Entity Recognition (NER) locates and identifies entities such as people's names, place names, and organizations in the word sequence of a sentence. Dependency Syntactic Parsing (DSP) reveals syntactic structure by analyzing the dependency relationship between sentence components, identifying the grammatical components such as "subject-predicate-object" and "attribute-adverbial-complement" in the sentence, and analyzing the relationship between these components.

First, use LTP to segment the input application information, make the complete sentence segmented into individual words. Then, use the part-of-speech tagging function to mark each word obtained after word segmentation. Finally, perform named entity recognition based on the results of part-of-speech tagging, and apply the recognition results to the subsequent dependency syntax analysis. Through dependency syntax analysis, the relationship between each word can be expressed, and the relationship between entities can be extracted, and then a ternary relationship group can be formed. The schematic diagram of the results obtained after the sentence is processed by LTP is shown in Figure 2.

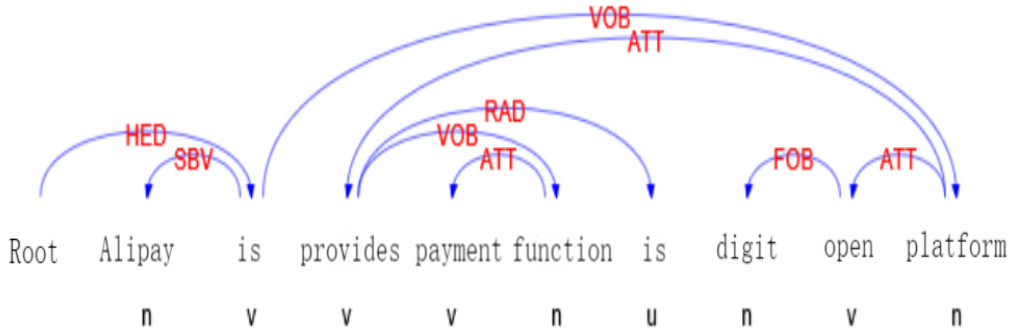

Figure 2. LTP processing result of application information.

In the background of this paper, the application feature information is generally included in the subject-verb relationship (SBV), verb-object relationship (VOB), and attribute relationship (ATT) which are in the application information. Extract these three kinds of relations in the LTP processing result, and form the ternary relationship. In the process of relation extraction, anaphora resolution is carried out on the entities that refer ambiguously. As shown in Figure 2 above, the entity "platform" and the entity "Alipay" are both Alipay, so the final ternary relationship extracted is (Alipay, is, Open platform) and (Alipay, provides, Payment function). All the ternary relationships of an application together form the ternary relationship group of the application.

Taking the application information of application, a∈A in the application service set A = {a1, a2, . . . , an} as input after the above steps are processed, the ternary relationship group Ga = (a, rj, tj), j = 1, 2, ... can be obtained. In this ternary relationship group, tj contains the feature words of application a, and the full set of tj is taken as the feature word set CFa = {CFa1, CFa2, . . . , CFan} of application a.

The importance of the extracted feature information of each application is different, some features are directly related to the application, and some features are not very relevant to the application. Therefore, consider using the TF-IDF algorithm to calculate the word frequency weight of the feature words, and retain the more important feature words as the extraction result. For application a, a∈A, feature word set CFa, CFa = {CFa1, CFa2 . . . , CFan}, the number of times the feature word CFai appears in the content feature text is nai, and in the content feature text total number of words is Σknak, then the TF value of the word CFai is shown in the following Equation (2):

$$TFa_i = \frac{na_i}{\sum_k na_k} \tag{2}$$

The total number of documents in the feature text corpus is D, and the number of documents containing CFai is Dai, then the IDF value of CFai is shown in the following Equation (3):

$$IDFa_i = \log\left(\frac{D}{Da_i + 1}\right) \tag{3}$$

In the content feature text of application a, the TF-IDF value of the word CFai is shown in Equation (4):

$$TF\text{-}IDFa_i = TFa_i \times IDFa_i \tag{4}$$

Perform weight sorting on the calculated TF-IDF values of the feature words, and select the top ω as the feature word set CFa' of application a, CFa' = {CFa1', CFa2', . . . , CFaω'}. $\omega = \left\lceil \frac{f}{\sigma} \right\rceil$, where $\lceil . \rceil$ expresses rounded up to an integer, f is the number of features in the feature set CFa, and σ is the scale factor. Compared with the original CFa, the new feature word set CFa' obtained by calculating the TF-IDF value of the feature words reduces some unnecessary feature words. At the same time, remove these unnecessary feature words

and their relationships from the ternary relationship group, thereby updating the ternary relationship group. The updated ternary group is recorded as G'a = (a, r'j, t'j), j = {1, 2, . . . , *n*}.

However, there is some unstructured or semi-structured information in the mobile service description information, so there are some features that cannot be extracted during text feature extraction. Therefore, consider fusing the extracted feature word set CFa' with the application features in the general knowledge graph G to supplement the application features and the knowledge graph.

In the process of project feature fusion and complement knowledge graph, there may be cases where the extracted application feature information is inconsistent with the node names in the knowledge graph. That is, the expression of the same word in the application feature information and the knowledge graph is inconsistent. In order to eliminate this ambiguity and make effective matching, the updated ternary relationship group G'a is linked to the entities in the knowledge graph using the DeepType model, taking the entities in the ternary relationship group G'a and the general knowledge graph G as input. The DeepType model first generates candidate entities from the general knowledge graph G according to the entities and relationships in G'a, and then generates matching results through its entity matching module to complete entity links.

For project a, the final project feature set Fa may have the following two results after entity linking:

(1) There are entities corresponding to applications in the knowledge graph.

In this case, there is an entity corresponding to application a in the knowledge graph. Through the entity linking the ternary relationship group, G'a is matched with the entity in the knowledge graph, and the attribute of applications that does not exist in the knowledge graph is supplemented. After the link is taken, the attributes of a in the knowledge graph constitute the EFa set. Take EFa as the feature set of applications, that is Fa = EFa.

(2) There is no entity corresponding to application a in the knowledge graph.

In this case, there is no entity corresponding to application a and no features related to application a in the knowledge graph, so all the ternary relationship groups G'a of a are added to the knowledge graph. The feature set Fa of application is CFa'.

Since some new entities and attributes are added to the knowledge graph, to find the potential directed edges (relationships) in the existing knowledge graph, the TransE method mentioned in Section 2 above is used to complement the static knowledge graph relationship. After the knowledge graph is modeled by TransE, the embedded vector of each entity and the relation can be obtained. By using the computability of the embedded vector and by calculating the similarity between t-h and relation vector, the potential implicit relationship between the two entities can be found and the potential relationship in the knowledge graph can be complemented.

The application feature fusion process based on the knowledge graph described in this section is shown in Figure 3.

The left column of Figure 3 contains mobile application characteristics information. The application introduction and update information are generally stored in an unstructured form, which cannot be directly used as application feature information, so it needs to convert into feature information by natural language processing. The middle column of Figure 3 shows how to process application feature information using natural language processing methods. First, the input application information is segmented and part-of-speech tagging is performed, and then named entity recognition is performed according to the result of the part-of-speech tagging, and the recognition result is applied to the subsequent dependency syntax analysis. Then, the TF-IDF algorithm is used to calculate the weight of the feature words, and the more important feature words are reserved as the extraction result. The right column of Figure 3 shows the fusion process of mobile application features and knowledge graph. First, the ternary relationship group is entity-linked with the existing general knowledge graph; then, the relationship is completed on the static knowledge graph according to the knowledge graph, and the completed knowledge graph is obtained.

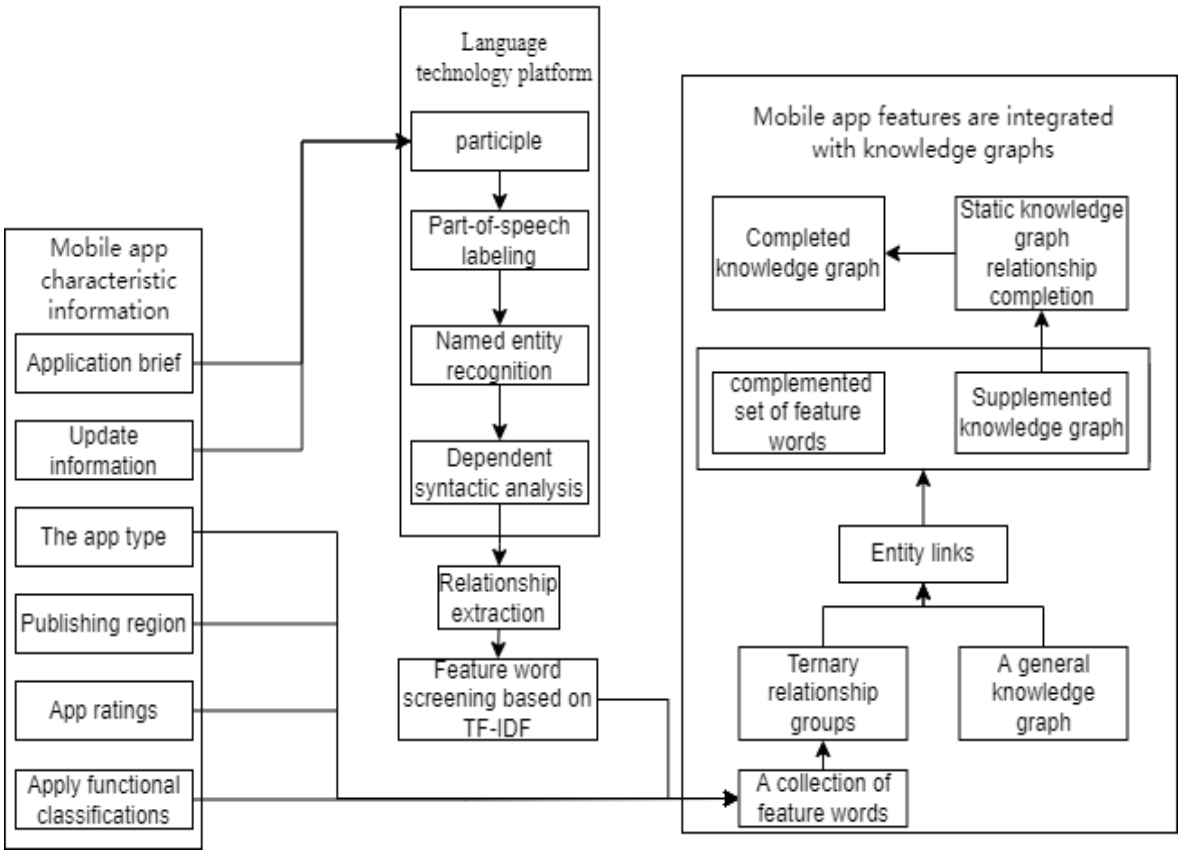

Figure 3. Application feature fusion flowchart based on knowledge graph.

3.2. Computation of Multidimensional Application Similarity

3.2.1. Application Similarity Based on Application Feature Vector

In the previous content, each application is represented in the form of the application feature set Fa. In the feature set Fa, the feature information of the application is contained. Now improve the representation form of the application feature. Each feature j in Fa is transformed into a corresponding word vector $\vec{Vaj} = (Vaj_1, Vaj_1, \ldots)$ through Word2vec, and the word vector module of feature j is calculated, as shown in Equation (5):

$$\|\vec{Vaj}\| = \sqrt{Vaj_1{}^2 + Vaj_2{}^2 + \ldots + Vaj_n{}^2} \tag{5}$$

After the above changes, the feature set Fa can be changed into the vector representation form $F'_a = (\left|\left|\vec{Va}1\right|\right|, \left|\left|\vec{Va}2\right|\right|, \ldots, \left|\left|\vec{Van}\right|\right|)$. Representing all applications as the above vectors and combining all application vectors into set F = (F′1, F′2, . . . , F′n). By comparing, the smallest application vector dimension m in F is obtained. Principal component analysis (PCA) is used to reduce the dimension of vectors in F whose dimension is higher than m, and the application vector set F′ = (I1, I2, . . . , In) with uniform dimension is obtained.

Using Pearson correlation coefficient to calculate the similarity between applications, which is shown in Equation (6):

$$sim_f(x,y) = \frac{N\sum I_x \cdot I_y - \sum I_x \cdot I_y}{\sqrt{N\sum I_x{}^2 - (\sum I_x{}^2)}\sqrt{N\sum I_y{}^2 - (\sum I_y{}^2)}} \tag{6}$$

3.2.2. Knowledge Graph Entity Similarity

There are large one-to-many, many-to-one, many-to-many, and reflexive relation triples in the knowledge graph involved in this paper, which is difficult to be solved by using the TransE method mentioned in Section 2 above. Therefore, in the process of calculating the similarity of the knowledge graph, this paper adopts the improved model of the TransE—TransHR [23] model, which can effectively solve the problem of difficult solutions of multiple relationships between application entities. The principle of TransHR is shown in Figure 4.

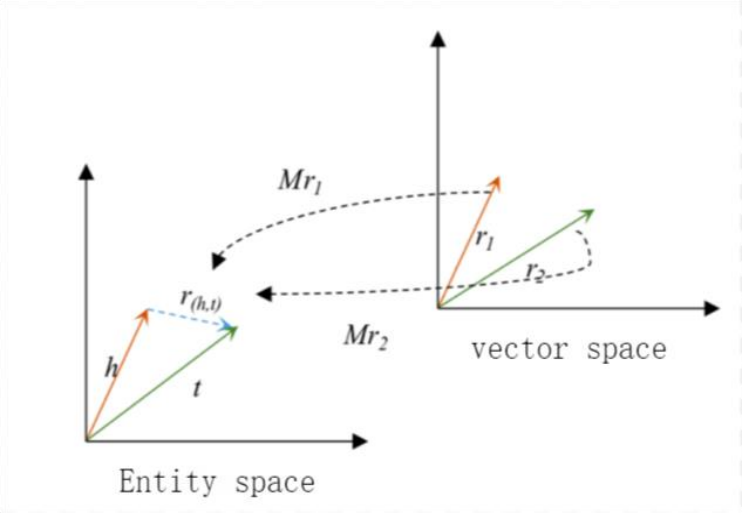

Figure 4. TransHR model.

The TransHR method differs from TransE in that TransHR embeds relationship vectors into a specific relationship space. Assuming that there are v relations between two application entities, ri is used to represent the i-th relation, and then all the relations between them are stored in a separate matrix space, which is denoted as Mr. Then, the relation vector of TransHR is shown in Equation (7):

$$r'_{(h,t)} = r_i M_{r_i},\ i = 1, 2, \ldots, v \tag{7}$$

Through the relation mapping of matrix space, the multiple relations between entities are expressed in the relation matrix M_{r_i}, and then the vector $r(h,t)$ is formed by the mapping, so as to realize the relation link between the head and tail entities. In the TransHR model, the head-to-tail entity mapping is shown in Equation (8):

$$h + r_{(h,t)} \approx t \tag{8}$$

In addition to representing each application entity, the TransHR method preservers the multiple relations within it. When calculating similarity in the knowledge graph, the similarity between applications can be calculated according to the weighting of multiple relations. Since the TransHR method stores the multiple relationships between application entities in a separate matrix space M_{r_i}, the calculation is more complex, so this paper replaces the matrix space M_{r_i} with the diagonal matrix A_{r_i}, aiming to reduce the complexity of the relational matrix. Thus, the vector mapping should be as shown in Equation (9):

$$r_{(h,t)} = r_i A_{r_r},\ i = 1, 2, \ldots, v \tag{9}$$

In the research scenario of this paper, the relationships stored in the knowledge graph are determined according to the types of medical applications, users' rating after using medical applications, medical application introduction, etc., of the mobile application entity. In the knowledge graph, the more the total number of the same relationships

between two medical application entities, the higher the similarity between the two medical application entities. For example, Xiaohe Health and Jianke Doctor have more relationships in the knowledge graph than Xiaohe Health and Chunyu Doctor. Therefore, the similarity calculation results of Xiaohe Health and Jianke Doctor are relatively large. In the scenario of this paper, the improved TransHR model is adopted, and the training loss function is shown in Equation (10):

$$L = \sum_{(h,r,t)\in S} \sum_{(h',r',t')\in S} \left[\gamma + d(h+r,t) - d\left(h'+r',t'\right)\right]_+ \tag{10}$$

where (h, r, t) represents a set of triples with the correct relationship, and (h', r', t') represents a set of triples with the wrong relationship. γ is the maximum distance between positive and negative samples, which is a constant, and $d(.)$ is the Euclidean distance.

By embedding the mobile application entity into a specific relationship space, the correctness of the entity-relationship in the triplet can be more clearly expressed. The triples with the right relationship can make condition $h + r \approx t$ true, while the wrong relationship in the wrong triples will make the triples far apart. Multiple relationships between mobile application entities are stored in the relationship matrix A_{r_i}.

When the application entity is embedded into the knowledge graph, the application entity is represented as a d-dimensional vector. The applied entity I_i is represented by a vector shown in Equation (11):

$$I_i = (E_{1i}, E_{2i}, \ldots, E_{di})^T \tag{11}$$

where, E_{p_i} represents the entity's corresponding value on the p-th dimension. Then, the Euclidean distance is used to calculate the distance between the above applied entity vectors which is shown in Equation (12):

$$d\left(I_i, I_j\right) = \sqrt{\sum_{k=1}^{d}\left(E_{ki} - E_{kj}\right)^2} \tag{12}$$

According to the distance between the applied entity vectors, the similarity between the two applied entities can be obtained. The similarity is shown in Equation (13):

$$sim_{sg}\left(I_i, I_j\right) = \frac{1}{1 + d\left(I_i, I_j\right)} = \frac{1}{1 + \sqrt{\sum_{k=1}^{d}\left(E_{ki} - E_{kj}\right)^2}} \tag{13}$$

As described above, in practice, the same relationship between two entities stored in the knowledge graph may have many situations, and the more the relationship between two entities, the higher the similarity. In the inter-entity relationship, there are two kinds of direct relationship and indirect relationship. In general, the influence of the direct relationship is greater than that of the indirect relationship. Therefore, $C_1(I_i, I_j)$ is defined as the number of direct relationships between I_i and I_j in the knowledge graph, and $C_2(I_i, I_j)$ is defined as the number of indirect relationships between I_i and I_j. Then, the similarity between I_i and I_j after adding weight factors is shown in Equation (14).

$$sim_{kg}\left(I_i, I_j\right) = \frac{1}{1 + \sqrt{\sum_{k=1}^{d}\left(E_{ki} - E_{kj}\right)^2}}\left[\frac{1}{x}C_1\left(I_i, I_j\right) + xC_2\left(I_i, I_j\right)\right] \tag{14}$$

We finally randomly selected 20 groups of applications from the mobile app market, each group consisting of 100 applications, and calculated their entity similarity based on the knowledge graph entity similarity. Due to space limitation, only the similarity calculation results of four applications are presented here with three decimal places. The similarity matrix between the obtained entities is shown in Table 2.

Table 2. Mobile application entity similarity matrix.

Mobile Application	WeDoctor	Xiaohe Health	Chunyu Doctor	Jianke Doctor
WeDoctor	0	2.245	1.760	2.311
Xiaohe Health	2.245	0	1.290	2.955
Chunyu Doctor	1.760	1.290	0	1.113
Jianke Doctor	2.311	2.955	1.113	0

3.2.3. Application Similarity Based on User-App Rating Matrix

In the traditional application-based collaborative filtering recommendation algorithm, there is a nearest neighbor algorithm idea which presumes that users tend to like similar applications, and then look for its nearest neighbor applications to recommend based on the applications that users have rated. In the recommendation process, first assume that there are m users and n applications, where $U = (U_1, U_2, \ldots, U_m)$, $I = (I_1, I_2, \ldots, I_n)$, and then input an user-app rating matrix $R_{m\times n}$ as a data set. The matrix $R_{m\times n}$ is shown in Equation (15).

$$R_{m\times n} = \begin{bmatrix} R_{11} & R_{12} & \ldots & R_{1j} & \ldots & R_{1n} \\ R_{21} & R_{22} & \ldots & R_{2j} & \ldots & R_{2n} \\ \ldots & \ldots & & \ldots & & \ldots \\ R_{i1} & R_{i2} & \ldots & R_{ij} & \ldots & R_{in} \\ \ldots & \ldots & & \ldots & & \ldots \\ R_{m1} & R_{m2} & \ldots & R_{mj} & \ldots & R_{mn} \end{bmatrix} \tag{15}$$

where, R_{ij} is the rating given by user U_{i} to application I_j, which represents the degree of user's preference for the application. The method of cosine similarity calculation is used to measure the similarity between matrix elements, that is, to calculate the cosine of the included angle between two vectors. Suppose there are two application rating vectors I_i and I_j, both of m-dimension, respectively $I_i = \{S_{1,i}, S_{2,i}, \ldots, S_{m,i}\}$ and $I_j = \{S_{1,j}, S_{2,j}, \ldots, S_{m,j}\}$, then the cosine similarity between vectors I_i and I_j is shown in Equation (16).

$$sim'_{\cos}(I_i, I_j) = \cos(I_i, I_j) = \frac{\sum S_{u,i} \cdot S_{u,j}}{\sqrt{\sum S^2{}_{u,i}} \cdot \sqrt{\sum S^2{}_{u,j}}} \tag{16}$$

$$sim'_{\cos}(I_i, I_j) \in [0, 1]$$

In Equation (16), the greater the result value, the greater the similarity between applications. When the value of $sim_{\cos}(I_i, I_j)$ is 0, it means that the two applications are completely dissimilar.

However, in practical applications, it is necessary to consider the difference in users' scoring preferences. It often happens in the rating of the same type of application. Some users may give too high or too low scores to the applications with average scores. In order to solve the problem of different users' rating criteria, the average rating factor should be

considered when calculating cosine similarity. The improved cosine similarity is shown in Equation (17):

$$sim_{\cos}(I_i, I_j) = \frac{\sum U_{i,j}(S_{u,i} - \overline{S}_u) \cdot (S_{u,j} - \overline{S}_u)}{\sqrt{\sum U_i (S_{u,i} - \overline{S}_u)^2} \cdot \sqrt{\sum U_j (S_{u,j} - \overline{S}_u)^2}} \tag{17}$$

In the Equation (17), $\overline{S_u}$ represents the average rating of the user.

3.3. Hybrid Recommendation Algorithm Based on Multidimensional Similarity Fusion

Combining the application similarity based on application vector, the application similarity based on knowledge graph and the improved rating similarity, the calculation method is shown as follows Equation (18):

$$sim_m(I_i, I_j) = W_{sim_f(I_i,I_j)} sim_f(I_i, I_j) + W_{sim_{kg}(I_i,I_j)} sim_{kg}(I_i, I_j) + W_{sim_{\cos}(I_i,I_j)} sim_{\cos}(I_i, I_j) \tag{18}$$

where $W_{sim_f(I_i,I_j)}$, $W_{sim_{kg}(I_i,I_j)}$, and $W_{sim_{\cos}(I_i,I_j)}$ are the weights of their corresponding similarity degrees. Now the three similarity degrees need to be weighted. The entropy weight method, as an objective weighting method, is widely used in the calculation process of determining the weights of different indexes at present. Therefore, it is considered to use the entropy weight method to determine the weights of each similarity. The specific process of using the entropy weight method to determine weight is shown in Equations (19)–(22):

$$x'_{m,sim_t(I_i,I_j)} = \frac{\left| x_{m,sim_t(I_i,I_j)} - \min(x_{sim_t(I_i,I_j)}) \right|}{\max(x_{sim_t(I_i,I_j)}) - \min(x_{sim_t(I_i,I_j)})} \tag{19}$$
$$t \in [f, kg, \cos], m \in (0, M]$$

$$y_{m,sim_t(I_i,I_j)} = \frac{x'_{m,sim_t(I_i,I_j)}}{\sum_{m=1}^{M} x'_{m,sim_t(I_i,I_j)}} \tag{20}$$

$$e_{sim_t(I_i,I_j)} = -K \sum_{m=1}^{M} y_{m,sim_t(I_i,I_j)} \ln y_{m,sim_t(I_i,I_j)},\ k = \frac{1}{\ln M} \tag{21}$$

$$W_{sim_t(I_i,I_j)} = \frac{1 - e_{sim_t(I_i,I_j)}}{\sum_{sim_t(I_i,I_j)} 1 - e_{sim_t(I_i,I_j)}} \tag{22}$$

In the above Equation, M is the number of data under index $W_{sim_f(I_i,I_j)}$, $W_{sim_{kg}(I_i,I_j)}$, and $W_{sim_{\cos}(I_i,I_j)}$. Taking the mixed similarity calculated by the above method as the weight, the user's prediction score for the unrated service is obtained through the weighted calculation. Assume that $pred(u, p)$ is the prediction score of user u for service p, $sim(i, p)$ is the mixed similarity between the rated service i and the predicted service p, $r_{u,i}$ is the rating of user u on the rated service i, and $ratditems$ is a collection of rated services. The predicted score is shown in Equation (23).

$$pred(u, p) = \frac{\sum\limits_{i \in rateditems} sim(i, p) * r_{u,i}}{\sum\limits_{i \in rateditems} sim(i, p)} \tag{23}$$

Based on the above algorithm ideas, this paper proposes a hybrid recommendation algorithm fused with knowledge graph (HRA-KG). The algorithm is shown as Algorithm 1 below:

Algorithm 1 Design of Hybrid Recommendation Algorithm Fused with Knowledge Graph HRA-KG

Input: target user u, similarity matrix, similarity matrix, similarity matrix, service number num, service set $Service_{set}$.
Output: The recommended service set $Toplist_{set}$ for the target user u
Begin
1: for I = 1 to num do
2: for j = 1 to num do
3: if $s_i \neq s_j$ then
4: CALCULATE_SIMILARITY(); //Calculate according to Equation (2)
5: end for
6: end for
7: $Toplist_{set} \leftarrow$ TOP-N($pred(u, s)$); //Calculate according to Equation (7)
8: return $Toplist_{set}$;

4. Experimental Results and Analysis

4.1. The Evaluation Index of the Experiment

In this chapter, precision, recall, and F-measure method were selected as the evaluation indexes of the experiment. Precision is a kind of representation of how many predicted positive samples are really positive samples, reflecting the level of recommendation results of the recommendation algorithm, and the level of correctly recommending the applications favored by users to users. The calculation of precision is shown in Equation (24):

$$Precision = \frac{TP}{TP + FP} \tag{24}$$

In the Equation (24), TP refers to the samples that users are interested in and have done some behavior, and FP refers to the samples that users are interested in but have not done any behavior.

Recall reflects the proportion of the recommendation results obtained through the recommendation algorithm in the total number of applications that users really like. The calculation of recall is shown in Equation (25):

$$Recall = \frac{TP}{TP + FN} \tag{25}$$

In the Equation (25), FN represents the sample of applications that the user is not interested in but has acted.

Choose F1-measure as the evaluation index of experimental results. F1-measure is the weighted harmonic average of precision and recall. The F1-measure considers both the accuracy rate and recall rate of the algorithm, and the evaluation results are more objective. The calculation of F-measure is shown in Equation (26).

$$F1 - measure = \frac{2 \times P \times R}{P + R} \tag{26}$$

In the Equation (26), P represents the Precision and R represents the Recall.

4.2. Data Source and Processing

The experimental data in this paper come from the Xiaomi Mobile App Store, and the experimental data we used are a subset of the mobile application service and user information data of the website. In terms of the choice of the knowledge graph, this paper selects OwnThink, which integrates over 25 million entities and hundreds of millions of relationships. The data formats in the knowledge graph mainly include the following two forms: (entity, relationship, entity) and (entity, attribute, value).

Through knowledge graph extraction, 20 groups of randomly selected mobile application knowledge base, including mobile application name, developer and type, etc., were finally obtained. Each group included 100 mobile application services, of which 70% were training set and 30% were test set.

4.3. Experimental Results and Analysis

Experimental 1: Determine the value of x by comparing the precision and recall under different undetermined coefficients x. When considering the two influencing factors of the direct and indirect relationship between the two entities in the knowledge graph existing in the actual situation, the two factors are integrated and use the undetermined coefficient x as the weight factor of the two influencing factors. The value of x is within the range of 0 to 1. With the change of the value of x, the values of precision and recall in this experiment will also change accordingly. The experimental results obtained are shown in Figures 5 and 6.

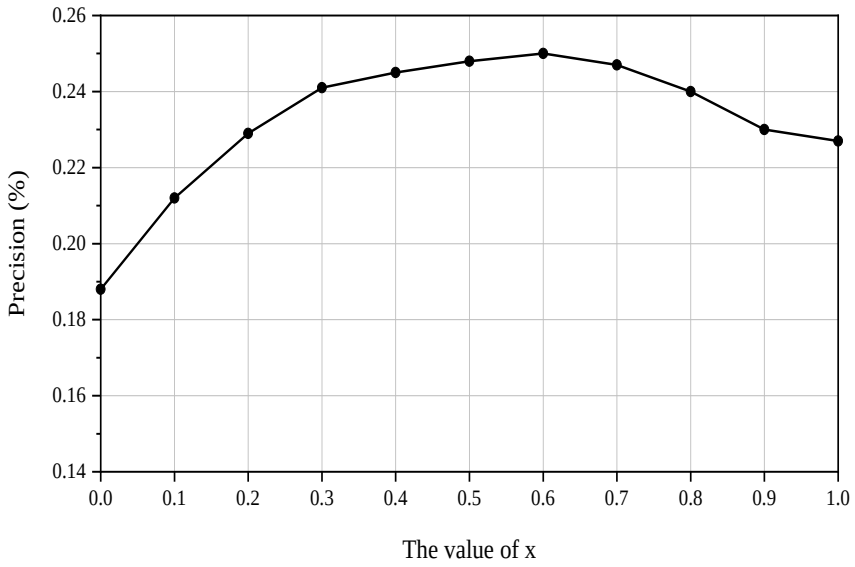

Figure 5. Precision under different values of x.

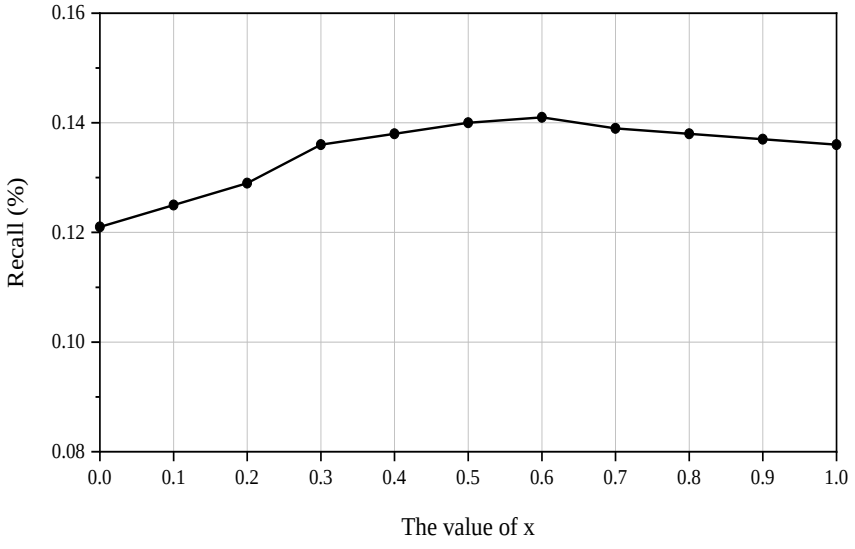

Figure 6. Recall under different values of x.

As shown in Figure 5, both the precision and recall rate increase with the increase of the undetermined coefficient and reach a maximum. Then they decrease with the increase

of x. Among them, when x is 0, the corresponding experimental results show that its precision and recall are the lowest values. The higher the precision and recall, the better the recommendation effect. As shown in Figure 6, when the x is less than 0.6, the precision and recall gradually increase as x increases; when the x is greater than 0.6, the precision and recall gradually decrease as x increases; the highest precision and recall occurs when the x is 0.6. The experimental results show that when the value of x is 0.6, the recommendation result is the most accurate. In this scenario, the direct relationship between the two entities in the knowledge graph is more influential than the indirect relationship.

Experiment 2: The proposed algorithm is compared with the collaborative filtering algorithm and HRACFE. In order to verify the effectiveness of the recommendation algorithm proposed in this paper, the recommendation effect of the proposed algorithm is compared with that of collaborative filtering recommendation algorithm (CF) [1] and the Hybrid Recommendation Algorithm base of Content Feature Extraction (HRACFE) proposed by reference [24]. The experimental results obtained are shown in Figures 7 and 8.

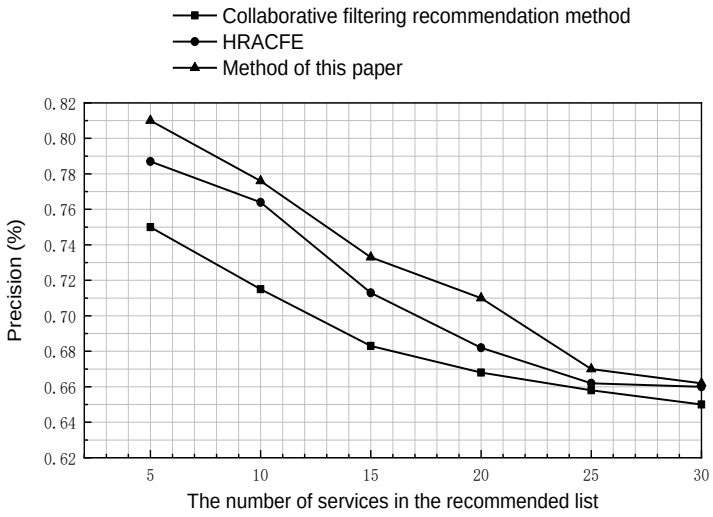

Figure 7. Precision comparison of different recommendation methods.

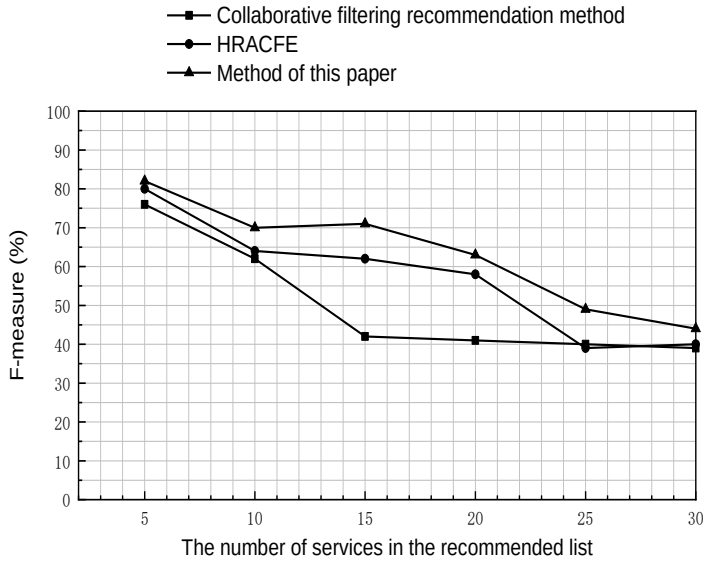

Figure 8. F-measure comparison of different recommendation methods.

It can be seen from the experimental results in the Figure 7, as the number of recommendation service increases from 5 to 30, the precision and F-measure value continue to decrease. When the number of recommendation service is 5, the precision and F-measure

value of the three methods are the highest. Subsequently, with the continuous increase of the number of recommendation services, the precision and F-measure value of the recommendation results of the three recommendation methods all show a downward trend. All things considered, the precision and F-measure value of the recommendation method of the fusion knowledge graph proposed in this paper are higher than the other two methods, and the advantage of our method is obvious when the number of recommendations services is less than 25. It shows that our method achieves the completion of the characteristics of mobile applications and effectively improves the precision of the recommendation results through the fusion knowledge graph. Moreover, our method alleviates the item cold-start problem.

5. Results

In this paper, we introduce the open knowledge graph into the field of medical mobile application recommendation. Through establishing the semantic link relationship between the mobile application and the knowledge graph entity, we not only enhance the single application semantics feature, but also enhance the semantics relationship among multiple application entities, then improve the semantic representation ability of the mobile application feature set as a whole, on this basis, we designed and implemented a medical service hybrid recommendation method incorporating the open knowledge graph. In this paper, we use LTP to extract the semantic feature set of mobile applications in the application market. Through the introduction of the OwnThink knowledge graph, we enhance the semantics of the application feature set and use the entropy method to organically integrate the calculation results of multi-dimensional semantic similarity, such as feature vector similarity, and entity relationship similarity, and user rating similarity. Under the help of our proposed recommendation algorithm, users can obtain their expected medical mobile applications conveniently and efficiently, then through these medical applications and their supporting medical diagnostic equipment, users can access to the satisfying continuous health monitoring and medical diagnostic services. Because we establish semantic links between mobile applications and knowledge graph entities. On the one hand, our method finds more similar relations between mobile applications in the completed knowledge graph through the TransHR model, which alleviates the cold start problem. On the other hand, we can obtain better feature vector similarity by the application feature set after semantic addition. We integrate multi-dimensional application similarity by entropy weight method, which improves the accuracy and effectiveness of recommendation results.

On the one hand, our method can help users to obtain satisfactory medical mobile applications conveniently and efficiently. On the other hand, it also makes developers pay more attention to the reasonable description and effective promotion of medical services when designing, developing, testing, deploying, and operating medical services and medical applications. Through the full and clear description of the function, type, and other characteristic information, these medical services, and supporting medical applications can be better found and recommended to the target users, so as to get higher downloads and attention. In the follow-up, we will conduct more in-depth research on the construction and promotion of knowledge graphs in the medical field to recommend better medical applications and provide better medical services for users

Author Contributions: Methodology, Z.Y.; writing—original draft preparation, J.Q.; writing—review and editing, Q.A.; supervision, H.Z.; project administration, C.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation, China (No.62172123), the special projects for the central government to guide the development of local science and technology, China (No. ZY20B11).

Conflicts of Interest: The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

1. Goldberg, D. Using collaborative filtering to weave an information tapestry. *Commun. ACM* **1992**, *35*, 61–70. [CrossRef]
2. Liu, L.; Lecue, F.; Mehandjiev, N. Semantic Content-based Recommendation of Software Services Using Context. *ACM Trans. Web* **2013**, *7*, 1–22. [CrossRef]
3. Hwang, W.S.; Lee, H.J.; Kim, S.W.; Won, Y.; Lee, M.S. Efficient Recommendation Methods Using Category Experts for a Large Dataset. *Inf. Fusion* **2016**, *28*, 75–82. [CrossRef]
4. Silva, E.Q.D.; Camilo, G.; Camilo-Junior, A.; Pascoal, L.M.L.; Rosa, T.C. An Evolutionary Approach for Combining Results of Recommender Systems Techniques Based on Collaborative Filtering. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 959–966.
5. Carmel, Y.; Patt-Shamir, B. Comparison-based Interactive Collaborative Filtering. *Theor. Comput. Sci.* **2016**, *628*, 40–49. [CrossRef]
6. Wang, H.Y.; Yang, W.B.; Wang, S.C. Service recommendation method based on trusted alliance. *Chin. J. Comput.* **2014**, *37*, 301–311.
7. Zhou, C.H.; Shen, J.J.; Li, Y.; Guo, X.F. Review of Classical Recommendation Algorithms. *Comput. Sci. Appl.* **2019**, *9*, 1803–1813.
8. Xu, H.L.; Wu, X.; Li, X.D.; Yan, B.P. Comparative Study of Internet Recommender Systems. *J. Softw.* **2009**, *20*, 350–362. [CrossRef]
9. Deshpande, M.; Karypis, G. Application-based top-N recommendation algorithms. *ACM Trans. Inf. Syst. (TOIS)* **2004**, *22*, 143–177. [CrossRef]
10. Tan, Z.; He, L. An efficient similarity measure for user-based collaborative filtering recommender systems inspired by the physical resonance principle. *IEEE Access* **2017**, *5*, 27211–27228. [CrossRef]
11. Hu, Y.; Peng, Q.; Hu, X.; Yang, R. Time aware and data sparsity tolerant web service recommendation based on improved collaborative filtering. *IEEE Trans. Serv. Comput.* **2014**, *8*, 782–794. [CrossRef]
12. Zhang, Y.J.; Shi, Z.K.; Zuo, W.L.; Yue, L.; Liang, S.N.; Li, X. Joint Personalized Markov Chains with social network embedding for cold-start recommendation. *Neurocomputing* **2020**, *386*, 208–220. [CrossRef]
13. Wang, X.; He, X.N.; Cao, Y.X.; Liu, M.; Chua, T.-S. KGAT: Knowledge Graph Attention Network for Recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 950–958.
14. Zhang, W.T.; Gu, T.L.; Sun, W.P.; Phatpicha, Y.; Liang, C.; Bin, C.Z. Travel Attractions Recommendation with Travel Spatial-Temporal Knowledge Graphs. In *Proceedings of the International Conference of Pioneering Computer Scientists, Engineers and Educators, Zhengzhou, China, 21–23 September 2018*; Springer: Singapore, 2018; pp. 213–226.
15. Wang, H.; Zhang, F.; Xing, X.; Guo, M. DKN: Deep Knowledge-Aware Network for News Recommendation. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1835–1844.
16. Xu, Z.L.; Sheng, Y.P.; He, L.R.; Wang, Y.F. Review on Knowledge Graph Techniques. *J. Univ. Electron. Sci. Technol. China* **2016**, *45*, 589–606.
17. Wang, H.; Zhang, F.; Wang, J.; Zhao, M.; Li, W.; Xie, X.; Guo, M. Exploring High-Order User Preference on the Knowledge Graph for Recommender Systems. *ACM Trans. Inf. Syst. (TOIS)* **2019**, *37*, 1–26. [CrossRef]
18. Lu, B.S. Research and Application of Collaborative Filtering Algorithm Based on Restricted Boltzmann Machine. Master's Thesis, Xi'an University of Technology, Xi'an, China, 2017.
19. Nickel, M.; Tresp, V.; Kriegel, H.P. A Three-Way Model for Collective Learning on Multi-Relational Data. In Proceedings of the 28th International Conference on International Conference on Machine Learning, Bellevue, WA, USA, 28 June–2 July 2011.
20. Bordes, A.; Weston, J.; Collobert, R.; Bengio, Y. Learning Structured Embeddings of Knowledge Bases. In Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 7–11 August 2011.
21. Socher, R.; Chen, D.; Manning, C.D.; Ng, A.Y. Reasoning with Neural Tensor Networks for Knowledge Base Completion. In *Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013*; Curran Associates Inc.: Red Hook, NY, USA, 2013; pp. 926–934.
22. Bordes, A.; Glorot, X.; Weston, J. A Semantic Matching Energy Function for Learning with Multi-relational Data. *Mach. Learn.* **2014**, *94*, 233–259. [CrossRef]
23. Zhou, M. The Research and Development of Question Answering System Based on Knowldege Graphs. Master's Thesis, Beijing University of Posts and Telecommunications, Beijing, China, 2017.
24. Ma, C.; Sun, Y.G.; Yang, Z.G.; Huang, H.; Zhan, D.Y.; Qu, J.X. Content Feature Extraction-Based Hybrid Recommendation for Mobile Application Services. *Comput. Mater. Contin.* **2022**, *71*, 6201–6217. [CrossRef]