information

Article

Medical Knowledge Graph Completion Based on Word Embeddings

Mingxia Gao [1,*], Jianguo Lu [2] and Furong Chen [3]

[1] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
[2] School of Computer Science, University of Windsor, Windsor, ON N9B 3P4, Canada; jlu@uwindsor.ca
[3] TravelSky Technology Limited, Beijing 101300, China; frchen@travelsky.com.cn
* Correspondence: gaomx@bjut.edu.cn; Tel.: +86-136-8108-2184

Abstract: The aim of Medical Knowledge Graph Completion is to automatically predict one of three parts (head entity, relationship, and tail entity) in RDF triples from medical data, mainly text data. Following their introduction, the use of pretrained language models, such as Word2vec, BERT, and XLNET, to complete Medical Knowledge Graphs has become a popular research topic. The existing work focuses mainly on relationship completion and has rarely solved entities and related triples. In this paper, a framework to predict RDF triples for Medical Knowledge Graphs based on word embeddings (named PTMKG-WE) is proposed, for the specific use for the completion of entities and triples. The framework first formalizes existing samples for a given relationship from the Medical Knowledge Graph as prior knowledge. Second, it trains word embeddings from big medical data according to prior knowledge through Word2vec. Third, it can acquire candidate triples from word embeddings based on analogies from existing samples. In this framework, the paper proposes two strategies to improve the relation features. One is used to refine the relational semantics by clustering existing triple samples. Another is used to accurately embed the expression of the relationship through means of existing samples. These two strategies can be used separately (called PTMKG-WE-C and PTMKG-WE-M, respectively), and can also be superimposed (called PTMKG-WE-C-M) in the framework. Finally, in the current study, PubMed data and the National Drug File-Reference Terminology (NDF-RT) were collected, and a series of experiments was conducted. The experimental results show that the framework proposed in this paper and the two improvement strategies can be used to predict new triples for Medical Knowledge Graphs, when medical data are sufficiently abundant and the Knowledge Graph has appropriate prior knowledge. The two strategies designed to improve the relation features have a significant effect on the lifting precision, and the superposition effect becomes more obvious. Another conclusion is that, under the same parameter setting, the semantic precision of word embedding can be improved by extending the breadth and depth of data, and the precision of the prediction framework in this paper can be further improved in most cases. Thus, collecting and training big medical data is a viable method to learn more useful knowledge.

Keywords: medical knowledge graph completion; word embeddings; RDF triples

check for updates

Citation: Gao, M.; Lu, J.; Chen, F. Medical Knowledge Graph Completion Based on Word Embeddings. *Information* **2022**, *13*, 205. https://doi.org/10.3390/info13040205

Academic Editor: Ryutaro Ichise

Received: 14 February 2022
Accepted: 28 March 2022
Published: 18 April 2022

1. Introduction

Medical Knowledge Graphs are widely used in automatic diagnosis systems [1], professional medical search engines [2], answering of medical questions [3], and other applications. However, most Medical Knowledge Graphs on existing networks, such as PHARE [4], are manually established by domain experts. This work is expensive and laborious. Most of the acquired Knowledge Graphs have only classes, and lack entities and the RDF triples of entities. Thus, in recent years, Medical Knowledge Graph Completion (MKGC) has been widely used to extend existing Knowledge Graphs. The goal of MKGC is to automatically predict one of three parts (head entity, relationship, tail entity) in RDF triples from medical data, mainly text data.

Most of the early methods for MKGC, such as [5], are based on rules. Following their introduction, the use of pretrained language models, such as Word2vec [6], ELMo [7], BERT [8], RoBERTa [9], and XLNET [10], to complete Knowledge Graphs [11,12], including Medical Knowledge Graphs [13–16], has become a popular research topic. The biggest limitation of these existing works is that they focus on completing relationships or classes, and rarely involve directly obtaining entities and RDF triples from unstructured text.

Our work focused on predicting entities about a semantic relationship for MKGC based on word embeddings. PTMKG-WE, a framework for predicting new triples for Medical Knowledge Graphs based on word embeddings, is proposed in this paper. The framework first acquires existing samples for a given relation from the Medical Knowledge Graph. Then, it trains word embeddings according to these samples through Word2vec from medical text. Third, it can obtain candidate triples from word embeddings based on analogies with existing samples. From the point of view of a given relationship, the analogy equation can express the embedding feature about the specific relationship. In order to refine a semantic relation, the paper proposes two ways to improve the relation feature. One is used to refine the relational semantics by clustering triple samples. The other is used for the accurate embedding expression of the relation feature through means of samples. Finally, in this study, PubMed data and NDF-RT were collected, and a series of experiments were conducted to evaluate the PTMKG-WE framework.

The experimental results yield the following conclusions. The first is that the data-rich model can achieve higher accuracy, in addition to the parameters during the training process. Thus, collecting and training big medical data is a viable method of learning more useful knowledge. The second conclusion is that PTMKG-WE can be used to predict entities for a given relationship in the Medical Knowledge Graph when medical data are sufficiently abundant and the Medical Knowledge Graph has appropriate existing samples.

This paper is organized as follows. The related works about KGC are introduced in Section 2. Section 3 explains the proposed method, including the overall architecture and implementation details. In Section 4, the experimental research, datasets, the results of the experiment are introduced. Finally, we conclude the paper in Section 5.

2. Related Works

The goal of MKGC is to solve the problems of incompleteness and sparsity caused by missing instances or links in existing Medical Knowledge Graphs. It is an important means of discovering new medial knowledge.

Most of the early methods for MKGC, such as [5], are based on rules. This kind of method has poor scalability. Following their introduction, the use of pretrained language models, such as Word2vec [6], ELMo [7], BERT [8], RoBERTa [9], and XLNET [10], to complete Knowledge Graphs [11,12], including Medical Knowledge Graphs [13–16], has become a popular research topic. The literature [11] proposes a method based on XLNET and a classification model to verify whether the triples of a Knowledge Graph are valid for relation completion. Although the quality of embedding learning based on XLNET is high, the method cannot be directly used to extend new triple knowledge from unstructured text. A method for leveraging pretrained transformer language models to perform scholarly Knowledge Graph Completion, named exBERT, is presented in [12]. Similar to [11], this method also performs relationship completion through two classifications. Another study [13] only applies Word2vec to the identification of relationships from unstructured text. However, compared with other published results, the results of this method are very limited. This process in [14] only uses distance similarities of word embeddings learned based on Word2vec to improve disease classes. Although these studies [13,14] use Word2vec, they can only deal with relationships and classes, and cannot supplement new entities. Another study [15] proposes a method to use BERT-enhanced entity representation or path representation to improve relationships in Medical Knowledge Graphs. This method can extend some specific relationships for a given Medical Knowledge Graph, but it cannot extend entities from unstructured text. An approach to model subgraphs

in a Medical Knowledge Graph is also proposed in [16]. Then, the learned subgraphs knowledge is integrated with BERT to perform entity typing and relation classification. In addition to the classification relationship, this work has begun to involve entities, but it still cannot expand new entities from unstructured text, and can only learn upper types for entities.

Certain studies [11,12] deal not only with non-medical Knowledge Graphs, but also with relational classification. Even if other studies can complete Medical Knowledge Graphs, they only expand relationships and types (or classes) at present. The biggest advantage of this work is that it can complement various entities from unstructured data.

The embeddings in this paper are word embeddings learned by Word2vec, which cannot solve the problem of polysemy, so it affects accuracies of this framework. The latest works, such as these studies [17,18], study methods of learning contextual embeddings in medical texts from the latest language models such as BERT. Next, the framework proposed in this paper can be naturally extended to contextual embeddings to improve accuracies.

3. Methods

3.1. Problem Statement

Generally, a Knowledge Graph includes different elements, such as entities, relationships, RDF triples, and so on. Knowledge Graph Completion (KGC) aims to complete the structure of Knowledge Graphs by predicting the missing entities or relationships in RDF triples from medical data, mainly text data. Therefore, it can be divided into three sub tasks: head entity prediction; relationship prediction; tail entity prediction. Relationships are closely related to specific areas. Domain experts are generally required to manually assist in relationship prediction. PTMKG-WE in this paper focuses on head-entity prediction and tail-entity prediction.

The work-flow of PTMKG-WE is shown in Figure 1. PTMKG-WE can be divided into three major steps, namely, (1) prior knowledge acquisition; (2) word embeddings training; (3) triples predicting. The first one refers to acquiring special types of prior knowledge through parsing a given Medical Knowledge Graph. As shown in Figure 1, some triples about "may_treat", such as (*glyburide, may_treat, pulmona*) and (*glipizide, may_treat, diabetes*), are acquired from a Medical Knowledge Graph named NDF-RT. These triples can be seen as existing samples. The second one is training word embeddings by Word2vec from medical data such as Figure 1. The third one is obtaining candidate triples from word embeddings based on an analogy relationship with prior knowledge. For an existing head entity "carvedilol", a new triple *(carvedilol, may_treat, X)* can be predicted. Through the analogy relationship $v(glipizide) - v(diabetes) == v(carvedilol) - v(X)$ some candidates for $v(X) == v(carvedilol) - (v(glipizide) - v(diabetes))$. can be obtained from word embeddings. Among others, the word "hypertension" can be seen as a tail entity in the new triple (*carvedilol*, *may_treat*, *hypertension*). Next, we discuss these steps in detail.

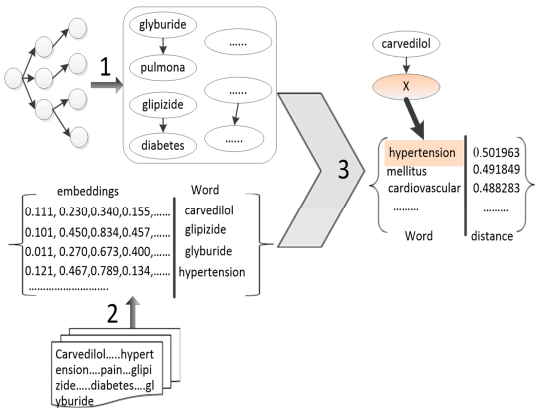

Figure 1. The workflow of PTMKG-WE.

3.2. Prior Knowledge Acquiring

For a special relationship in a given Knowledge Graph, we can acquire some triples including the relationship as prior knowledge or existing samples by parsing the Knowledge Graph. The paper uses Apache Jena (Jena for short) to parse a given OWL Knowledge Graph.

For a special relationship, the set of existing triples (head entity, relationship, and tail entity) is prior knowledge. For complex Knowledge Graphs, existing samples need some preprocessing, such as name segmentation, semantic merging, and so on. Special details will be presented in experiments.

3.3. Word Embeddings Trained by Word2vec

In statistical language modeling, word embedding (or a word vector) is a k-dimensional vector in R^k to represent each word. Recently, some neural network models such as [6–10] for training word embeddings have been proposed. Word2vec is one of the common tools. Thus, it is used in this study to learn word embeddings from medical data. There are two models, such as Skip-gram and CBOW, and two training methods, such as the hierarchical softmax and Negative sampling in Word2vec. Skip-gram is slow and good at infrequent words. CBOW is fast and good at common words. The hierarchical softmax is good at infrequent words and Negative sampling is good at common words and low-dimension vectors. Compared to the whole dataset, these words in a given Knowledge Graph, such as the drug list and drug characteristics in NDF-RT, are infrequency. Thus, Skip-gram and the hierarchical softmax are used in this study. Other parameters, such as -window (window widths), -sample (sampling threshold), and -size (dimensions of vector) may affect results. In order to select appropriate parameters, different word embeddings with different -window, -size, and -sample will be evaluated in this work.

After the model is trained, the distance similarity and analogy relationship based on word embeddings can be used to explain semantic relationships. The distance similarity computes the value of cosine between two word embeddings. The analogy relationship satisfies a semantic equivalence for two pairs of words based on embeddings; for example, $v(ribavirin) - v(hepatitis) = v(cefotaxime) - v(pneumonia)$, where $v(w)$ is the embedding of a word w.

3.4. Triples Predicting

For a given relationship, once we have acquired a suitable set of triples as prior knowledge and trained word embeddings from medical data, we can get a set of tail/head entities candidates for a given head/tail entity to combine new triples (head entity, Relationship, tail entity). From the feature point of view, the embeddings offset between the head entity word and the tail entity word in a triple, that is $v(w_i) - v(w_j)$, represents the embedding feature of the particular relationship in the triple. In theory, the embeddings offsets between entities in any two triples including the same relationship satisfies semantic equivalence. For example, there are two triples (*cycloserine*, *may_treat*, *tuberculosis*) and (*praziquantel*, *may_treat*, *schistosomiasis*) in a given Medical Knowledge Graph. The equation about entities in the triples based on embeddings holds; that is $v(cycloserine) - v(tuberculosis) = v(praziquantel) - v(schistosomiasis)$, where $v(w)$ is the embedding of a word w. Thus, if a given relationship R and an entity C and a reference triple (A_j, R, B_j), an unknown entity X used to compose a new triple (C, R, X) can be obtained from embeddings according to the analogy relationship with the reference triple. The solution for $v(X)$ is: $v(X) = v(C) - (v(A_j) - v(B_j))$ based on the analogy equation $v(C) - v(X) = v(A_j) - v(B_j)$. Then, we can compute cosine similarities between $v(X)$ and each word embedding in a model and rank these words according to these similarities. Finally, the top N words are selected as candidates of X. The specific implementation method is Algorithm 1 in Section 3.4.

Although every triple in existing samples can be seen as a reference triple for predicting new triples alone, not every triple is a positive sample. Table 1 shows three samples about

"may_treat" in NDF-RT. The first two are positive samples and the last one is a negative sample. There are many reasons for generating negative samples. Knowledge Graphs are incorrect or there are multiple semantics for a given relationship. Due to the small amount of data, the training model is not accurate enough. These problems can be improved at the experimental data level. Now, we are more concerned about the inherent semantic problems of relationships and corresponding solutions.

Table 1. Examples of embedding analogy for NDF-RT.

No.	Examples
1	*v(fentanyl) − v(pain) = v(polymyxin) − v(urinary)*
2	*v(interferon) − v(purpura) = v(rifampin) − v(leprosy)*
3	*v(fluocinolone) − v(facial)/ = v(topiramate) − v(spasms)*

The most obvious problem is that semantics of a relationship in existing samples are not single in a given Knowledge Graph. For example, the head entity and tail entity of "may_treat" are "Pharmaceutical Preparations" and "Diseases, Manifestations or Physiologic States", respectively. The semantic range of this relationship contains two parallel concepts: "Diseases" and "Manifestations or Physiologic States". From this point of view, this relationship can be further refined. The clustering is a more common method of refinement. Given word embeddings and a set of triples including a same relationship R, that is $\{(A_1, R, B_1), \ldots, (A_k, R, B_k)\}$, each embeddings offset between two entities in the same triple $v(A_j) - v(B_j)$ represents the embedding feature of the relationship R for a corresponding sample. Thus, the embeddings offset can be seen as clustering features. Then, all triples can be clustered according to similarities of embeddings offsets between every two pairs of triples. In this study, cosine is used to compute similarities between any two embedding offsets and K-means is used to cluster. The number of clusters and the number of iterations (two user parameters) are closely related to experimental data. PTMKG-WE using the cluster method to refine a relationship is named PTMKG-WE-C in this paper. The specific implementation method is Algorithm 1 in Section 3.4.

Algorithm 1 Predicting Triples Algorithm

Input: a given entity C; word embeddings R^k; a set of triples related to a relationship R $T = \{(A_1, R, B_1), \ldots, (A_k, R, B_k)\}$; the number of candidate words *TopN*; the number of clusters CN; the number of iterations IN;
Output: a set of candidate words $\{w_1, \ldots, w_{TopN}\}$;
*** 1. Refining relationships by clustering ***
if $T \neq$ *null*
{**for** *(j = 1, j + +, k)*
{Acquiring embeddings from $R^k A_j$ for and B_j;
$R_j = v(A_j) - v(B_j)$;}
K-means ($\{R_1, \ldots, R_k\}$, CN, IN);}
*** 2. Refining relationships by mean ***
for *(j = 1, j + +, CN)*
{**if** $R_j \neq$ null
{$n = |R_j|$;
$R = \frac{1}{n}\sum_{j=1}^{n} R_j$;
Acquiring $v(C)$ from R^k;
$v(X) = v(C) - R$;
*** 3. Acquiring candidates ***
Acquiring the top *TopN* words $\{w_1, \ldots, w_{TopN}\}$ from R^k;}}
*** clustering by K-means ***
Input: a set of clustering elements R; the number of clusters CN; the number of iterations IN;
Output: clustering results $\{R_1, \ldots, R_{CN}\}$ doing K-means;

Even by refinement, each finer relationship will also have multiple triples as samples.

In order to accurately express the feature of this relationship, computing the mean of all samples is a better method. A given set of triples related to a relationship R, that is $\{(A_1, R, B_1), \ldots, (A_k, R, B_k)\}$, the embedding feature of the relationship can be expressed in $v(R) = \frac{1}{k} * \sum_{j=1}^{k} (v(A_j) - v(B_j))$. Thus, the solution for $v(X)$. is: $v(X) = v(C) - v(R)$ based on the mean of all samples. PTMKG-WE and PTMKG-WE-C using the mean of samples to represent the embedding feature of a relationship are named PTMKG-WE-M and PTMKG-WE-C-M, respectively. The specific implementation method is Algorithm 1 in Section 3.4.

4. Experiments

4.1. Medical Knowledge Graphs and Preprocessing

NDF-RT is produced and maintained by the U.S. Department of Veterans Affairs, Veterans Health Administration (VHA). NDF-RT is a formal representation of the drug list and drug characteristics, including ingredients, chemical structures, dose forms, physiologic effect, mechanism of actions, pharmacokinetics, and related diseases. New versions of NDF-RT are released every year. There are 43,474 entities, 30 object properties, 39 data properties, and 945,542 triples in NDF-RT (2015 version). There are plenty of triples about a special relationship, such as "may_treat" in NDF-RT. Thus, NDF-RT is used as a Medical Knowledge Graph to evaluate the methods proposed by this paper.

For a given relationship, for example "may_treat" in NDF-RT, a set of triples including the relationship is acquired by parsing the Knowledge Graph. Because the concept granular in these triples is too fine, multiple triples may express the same semantic. For example, (*EDETATE DISODIUM*, *may_treat {NDFRT}*, *Manganese Poisoning [Disease/Finding]*) and (*EDETATE DISODIUM 150MGML INJ [VA Product]*, *may_treat {NDFRT}*, *Manganese Poisoning [Disease/Finding]*) express the same semantic. Triples with the same semantic are merged by ignoring detailed descriptions, mainly including brackets and descriptions, punctuation, and dose. To simplify experimental processes, we only consider two types of triples. The first one, named single-word triples, only includes single-word entities. The second one, named double-word triples, only includes single-word or double-word entities. Table 2 shows the sums of triples, head entities, and tail entities corresponding to two relationships, "may_treat" and "may_prevent", in NDF-RT. Among these, (*X*, *may_treat*, *Y)* presents that (*drug X may treat disease/symptom Y*) and (*X*, *may_prevent*, *Y*) presents that (*drug X may prevent disease/condition Y*).

Table 2. Sums of triples, head or tail entities corresponding to two relationships in NDF-RT.

Relationship	Type of Triples	Triples	Head	Tail
may_treat	original triples	51,248	11,655	956
	single word	1915	930	334
	double word	8677	2566	680
may_prevent	original triples	6292	4076	215
	single word	262	216	70
	double word	1094	742	162

As shown in Table 2, the number of head entities is greater than the number of tail entities; that is, the number of drugs is greater than the number of diseases/symptoms/conditions. Thus, "may_treat" and "may_prevent" can not only represent one-to-one relationships but also represent many-to-many relationships. Figures 2 and 3 present the statistical information of "may_treat" and "may_prevent", respectively. Among these, Figures 2a,c and 3a,c show the Figures distribution of drugs that can treat different numbers of diseases and 2b,d and 3b,d represent the distribution of diseases that can be treated with different numbers of drugs. Figure 2a describes the distribution of drugs in single-word triples on "may_treat" in detail. The horizontal axis indicates the number of diseases that a drug can treat, and

the vertical axis represents the total number of drugs that can treat the same number of diseases. The sum of drugs that can treat a disease is more than 450. The total number of diseases that can be treated by one drug is up to 12, and there is only one such drug. As the total number of diseases that can be treated by one drug increases, the total number of such drugs also decreases sharply. This trend is in line with our actual cognition. For the relationship "may_prevent", the distribution trend is similar, as shown in Figure 3a, but the distribution is simpler.

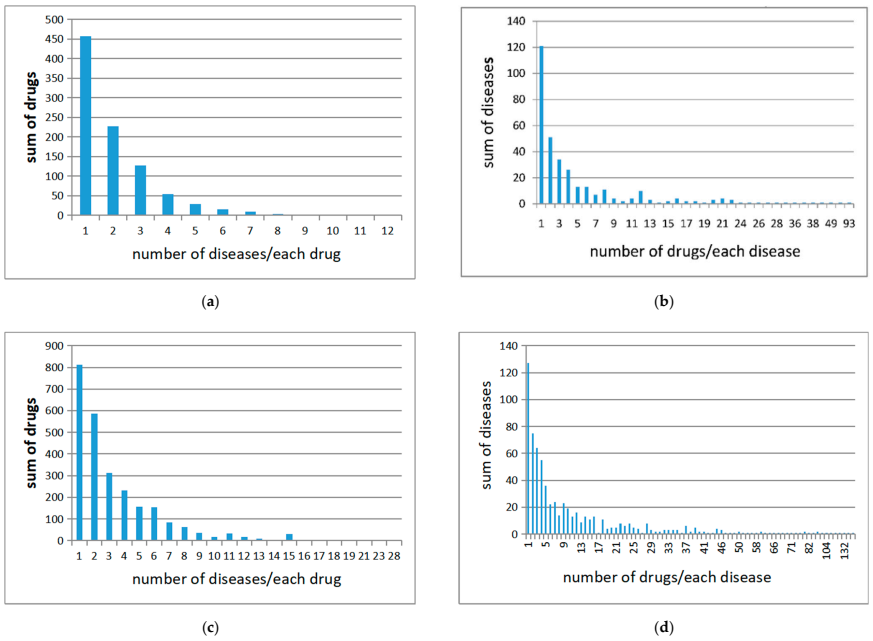

Figure 2. The distribution of drugs and diseases for "may treat". Among these, (**a**,**c**) show the distribution of drugs that can treat different numbers of diseases and (**b**,**d**) represent the distribution of diseases that can be treated with different numbers of drugs. (**a**) Single-word triples; (**b**) Single-word triples; (**c**) Double-word triples; (**d**) Double words triples.

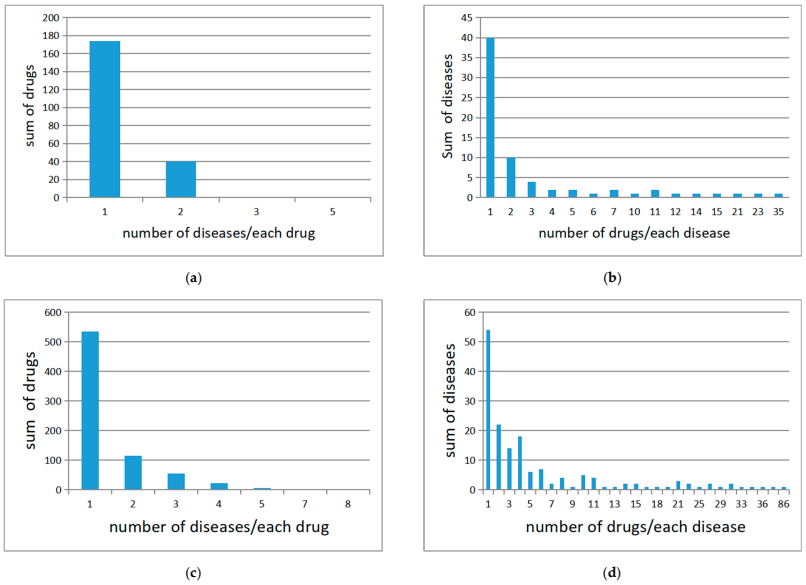

Figure 3. The distribution of drugs and diseases for "may prevent". Among these, (**a**,**c**) show the distribution of drugs that can treat different numbers of diseases and (**b**,**d**) represent the distribution of diseases that can be treated with different numbers of drugs. (**a**) Single-word triples; (**b**) Single-word triples; (**c**) Double-word triples; (**d**) Double-word triples.

4.2. Data Collecting and Preprocessing

The data in this paper are mainly abstracts from PubMed in 2016 and articles from PubMed Central Open Access Subset (PMCOAS) from October 2016. PubMed comprises more than 26 million citations for biomedical literature from MEDLINE, life-science journals, and online books in 2016. Some citations may include links to full-text content from PubMed Central (PMC) and publisher websites. PMC is a free full-text archive of biomedical and life sciences journal literature at the U.S. National Institutes of Health's National Library of Medicine (NIH/NLM). PMCOAS is a part of the total collection of articles in PMC. As of 2016, there were over 1.3 million articles available in this collection. Table 3 is the details of datasets used in experiments. Data1 and data2 have the same source. Data1 includes titles and abstracts and data2 increases the bodies of papers. Data2 can be regarded as a deep extension of data1. Data3 and data4 are from the same source. Data3 includes only titles and data4 includes titles and abstracts. Similarly, Data4 can be regarded as a deep extension of data3. Both data1 and data4 include titles and abstracts, but data4 adds 24 M new articles in addition to articles in data1. Therefore, data4 can be regarded as a breadth expansion of data1.

Table 3. The details of datasets.

Name	Source	Articles	Size	Title	Abstract	Body
data1	PMCOAS	1.3 M	1.21 G	Y	Y	N
data2			19.8 G	Y	Y	Y
data3	PubMed	26 M	2.2 G	Y	N	N
data4			15.12 G	Y	Y	N
data5			22.06 G	Y	Y	N
C34dd1	PMCOAS	1.3 M	1.21 G	Y	Y	N
C34dd2			19.8 G	Y	Y	Y
C34dd3	PubMed	26 M	2.2 G	Y	N	N
C34dd4			15.12 G	Y	Y	N
C38dd1	PMCOAS	1.3 M	1.21 G	Y	Y	N
C38dd2			19.8 G	Y	Y	Y
C38dd3	PubMed	26 M	2.2 G	Y	N	N
C38dd4			15.12 G	Y	Y	N

Data1, data2 and data4 include punctuation, so data1, data2 and data4 are prepossessing through some nature language methods, such as deleting marks or stopping words. Data3 only includes titles and may have very little punctuation without prepossessing. To verify different forms of corpus, data5 is the original format of the data4 without prepossession. To build double-word data for double-word triples, what needs extra treatments is the recognition and connection of two words in data1 to data4. For a given relationship, a recognition method based on double-word dictionaries is used in this paper. For example, the double-word dictionary generated in NDF-RT is used to recognize "may" and "treat" for "may_treat". The link symbol between two words is "_". c34dd1-4 and c38dd1-4 in Table 3 are double-word data attained through the recognition and connection of two words in data1-4 for the tworelationships "may_treat" and "may_prevent". Their source and distribution are the same as data1-4.

4.3. Experimental Setup

The experimental environment is CPU: 32*Intel® Xeon® CPU E5-2620 v4@2.10 GHz and memory 131900072KB, Operating system: Linux 4.15.0-128-generic Ubuntu 16.04.6 LTS server.

The essence of evaluating PTMKG-WE and its improving methods is to verify whether a triple removed from a Knowledge Graph can be retrieved from word embeddings using PTMKG-WE and its improving methods. For example, a triple (*fludarabine*, *may_treat*,

leukemia) is removed from NDF-RT. Another triple (*benzocaine*, *may_treat*, *otitis*) is used as a reference triple to find *fludarabine/leukemia* from word embeddings trained from medical data for a given entity, *leukemia/fludarabine*. A set of top *TopN* (a user parameter) candidate words *W* is acquired from word embeddings using PTMKG-WE or its improving methods. For a given entity *leukemia/fludarabine*, if *fludarabine/leukemia* or its synonyms (only considering synonyms presented by "perperty:synonyms" in NDF-RT) can be found from *W*, the triple that is removed can be seen as being retrieved. The paper only considers predicting unknown tail entities for a given head entity. The more candidates there are, the more opportunities there are to obtain the tail entity that has been removed. Similarly, experiments take more time. For balance, *TopN* was set to 20 during the experiments.

In order to ensure entities in triples are included in each model, the original triples in Table 2 are not used in these experiments. By filtering out tail entities whose word frequency in data is less than a threshold, we obtain experimental RDF triples. In experiments, the threshold was set to five.

The accuracy is the ratio of the number of triples retrieved and the number of triples removed. The accuracy is the overall statistical indicator and cannot measure the specific ranking. The mean reciprocal rank (MRR) is used to evaluate sorting results. The reciprocal rank in this paper calculates the reciprocal of the rank of the first retrieved tail entity in a candidate queue. If the tail entity is not found in the candidate queue, we assume that the reciprocal rank is zero. The mean reciprocal rank is the average of the reciprocal ranks of the results with multiple triples removed. The reciprocal rank of a tail-entity candidate in a special case; that is, a drug may treat different diseases in a given Knowledge Graph, need extra considerations. For example, a given drug "ceftriaxone" may treat three diseases such as "pneumonia", "septicemia", and "meningitis" in NDF-RT. Thus, there are three triples about the drug, that is *(ceftriaxone*, *may_treat*, *pneumonia)*, *(ceftriaxone*, *may_treat*, *septicemia)*, and *(ceftriaxone*, *may_treat*, *meningitis).* If the first triple is removed, a candidate queue is retrieved from embeddings by the methods in this paper for unknown tail entities. If "septicemia", "meningitis", and "pneumonia", respectively ranks one, two, and three in the candidate queue. The rank of "pneumonia" was adjusted from three to one by ignoring these tail-entity words of existing samples in the candidate queue, such as"septicemia" and "meningitis". In experiments, MRR obtained by recalculating RR is named MRR+.

To obtain the top *TOPN* candidate words from word embeddings, word embeddings need to be traversed once when retrieving each removed triple. When the number of triples is relatively large, the experimental time cannot be tolerated. In order to improve the time efficiency of experiments, the number of traversal embeddings must be reduced. Each experiment is performed using random sampling 100, when the number of samples is greater than 100. The final accuracy is the mean of three random experiments.

4.4. Experimental Results

4.4.1. Models Trained by Word2vec

To train word embeddings, Word2vec in C language is used in this study. To match single-word triples and double-word triples from NDF-RT, single-word embeddings and double-word embeddings must be separately trained by Word2vec. As Section 3.3 denotes, we have trained a series of word embeddings using different parameters (-window, -sample, -size) in Word2vec. The effect of -sample is not very obvious. A slight advantage setting 1×10^{-3} is chosen. -window and -size have a relatively large impact on the experimental results. The higher the -size is, the better the result is. However, the higher the -size is, the larger embeddings are, and the more time experiments take. Thus, -size is usually set 400 during training. Data3/c38dd3/c34dd3 only includes titles. Generally, the number of words in titles is less than 10. So, smodel3/c38dmodel3/c34dmodel3 is trained from data3/c38dd3/c34dd3 through -window = five. In addition to data3/c38dd3/c34dd3, a sentence in other data may include diseases and drugs. Most of the names of drugs and diseases are multi-words entities. The number of words in a sentence may be more than 10. Figure 4 shows accuracy comparison of different settings for -window. When

-window = 15, PTMKG -WE and PTMKG -WE-M get the best accuracy. Therefore, other models are trained from data with -window = 15, as shown in Table 4.

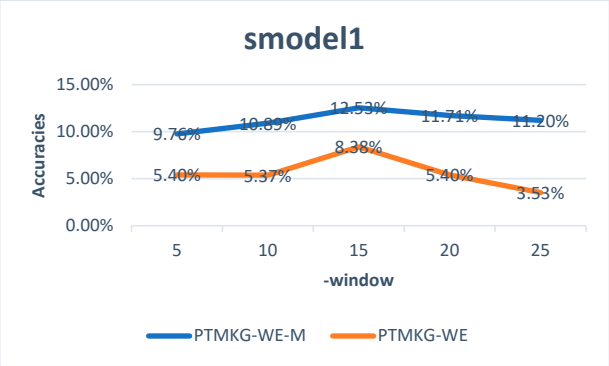

Figure 4. Accuracy comparison of different settings for -window on data1.

Table 4. Embeddings in experiments.

Name	Source	-Windows	Size (G)	Words
smodel1	data1	15	0.5	155,407,624
smodel2	data2	15	2.09	2,634,263,510
smodel3	data3	5	1.20	307,063,987
smodel4	data4	15	2.14	1,924,072,604
smodel5	data5	15	6.30	3,390,842,792
c34dmodel1	c34dd1	15	0.5	155,192,660
c34dmodel2	c34dd2	15	2.09	2,631,795,611
c34dmodel3	c34dd3	5	1.2	155,192,660
c34dmodel4	c34dd4	15	2.14	2,631,795,611
c38dmodel1	c38dd1	15	0.5	155,192,660
c38dmodel2	c38dd2	15	2.09	2,631,795,611
c38dmodel3	c38dd3	5	1.2	155,192,660
c38dmodel4	c38dd4	15	2.14	2,631,795,611

When other settings such as methods are the same, the accuracy of smodel4 is generally better than that of smodel5, as shown in Table 5, and the time efficiency of smodel4 is far higher than that of smodel5. Therefore, it is necessary to reprocess the corpus, even if it is simple to remove the punctuation. Smodel5 will be not used in future experiments.

Table 5. Accuracy comparison of different models.

Models	Methods	Accuracies
smodel4	PTMKG-WE	10.14%
	PTMKG-WE-M	16.33%
smodel5	PTMKG-WE	5.66%
	PTMKG-WE-M	9.67%

4.4.2. Experimental Results for Single Words

Figure 5a,b shows accuracies of four algorithms proposed in this paper on the same model for "may_treat" and "may_prevent". The accuracies of two clustering methods denote the final comprehensive accuracy in Table 6. By comparing the accuracies of PTMKG-WE and PTMKG-WE-M on different models, it can be found that the accuracy of PTMKG-WE-M is nearly one time higher than that of PTMKG-WE for the same model. The accuracy of PTMKG-WE-C-M is also close to one time greater than PTMKG-WE-C for the same model. This means that the embedding feature of a given relationship represented by the mean of samples, as performed in PTMKG-WE-M and PTMKG-WE-C-M, is very effective

for all models. The accuracy increases of PTMKG-WE-C/PTMKG-WE-C-M relative to PTMKG-WE/PTMKG-WE-M will be affected by the specific model and data. As shown in Figure 5a, for "may_treat", the accuracy of PTMKG-WE-C/PTMKG-WE-C-M increases much more on smodel2 than on smodel3. For "may_prevent" in Figure 5b, the increment of accuracy of PTMKG-WE-C/PTMKG-WE-C-M in smodel1 is much larger than that in smodel3. However, the increasing trend of the accuracy of PTMKG-WE-C/PTMKG-WE-C-M compared with that of PTMKG-WE/PTMKG-WE-M is obvious. This trend clearly shows that it is effective to obtain more accurate sub-relations through clustering used in these methods as PTMKG-WE-C and PTMKG-WE-C-M. When clustering and mean effect are superimposed on one method, the accuracy of this method is greatly improved. As shown in Figure 5, PTMKG-WE-C-M is the superposition of four methods and obtains the best accuracy.

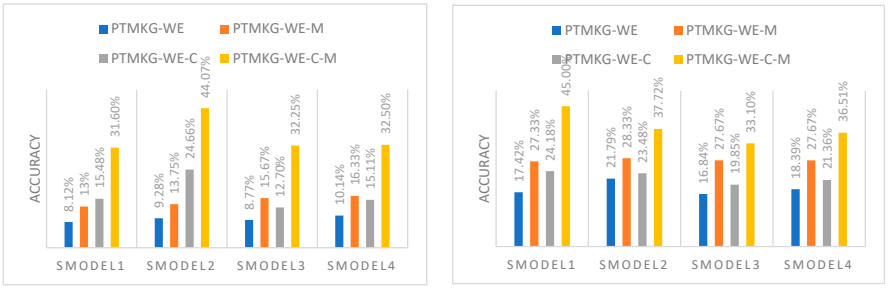

Figure 5. Accuracy comparison of four algorithms on a same model. (**a**,**b**) show accuracies of four algorithms on a same model for "may_treat" and "may_prevent"; (**a**) may_treat; (**b**) may_prevent.

Table 6. Accuracies of PTMKG-WE-C and PTMKG-WE-C-M for "may_treat" and "may_prevent" in different models.

Relationship	No./Triples	Methods	Accuracies			
			smodel1	smodel2	smodel3	smodel4
may_treat	c1/19	PTMKG-WE-C PTMKG-WE-C-M	45.61% 89.47%	60.8% 89.5%	17.5% 78.9%	23.4% 89.5%
	c2/121	PTMKG-WE-C PTMKG-WE-M	74.02% 89.24%	64.1% 90.7%	20.7% 90.3%	35.7% 88.7%
	c3/599	PTMKG-WE-C PTMKG-WE-M	10.81% 18.67%	16.6% 24%	11.2% 22.3%	14.6% 20.3%
	c4/73	PTMKG-WE-C PTMKG-WE-C-M	70.83% 89.04%	58.4% 97.3%	52.9% 89%	28.8% 91.8%
	c5/29	PTMKG-WE-C PTMKG-WE-C-M	89.16% 86.21%	88.2% 79.3%	77.8% 79.3%	74.3% 82.8%
	c6/210	PTMKG-WE-C PTMKG-WE-C-M	8.2% 26.67%	15.2% 42%	5.3% 17%	7.5% 34.3%
	c7/355	PTMKG-WE-C PTMKG-WE-C-M	10.19% 17.33%	14.5% 21.7%	7% 15%	10.7% 17%
	c8/16	PTMKG-WE-C PTMKG-WE-C-M	62.92% 50%	42.5% 37.5%	14.6% 43.8%	0 43.8
	c9/99	PTMKG-WE-C PTMKG-WE-C-M	17.73% 37.37%	13.1% 43.4%	6.5% 35.4%	9.3% 39%
	c10/64	PTMKG-WE-C PTMKG-WE-C-M	15.25% 25%	21.3% 43.8%	6.2% 21.9%	10.9% 26.6%
	c1-c10/1585	PTMKG-WE-C PTMKG-WE-C-M	15.26% 31.87%	24.02% 38.7%	13.33% 31.47%	19.86% 32.33%
may_prevent	c1/35	PTMKG-WE-C PTMKG-WE-C-M	50.17% 88.57%	57.9% 97.1%	41.5% 81.7%	29.2% 91.4%
	c2/160	PTMKG-WE-C PTMKG-WE-C-M	19.59% 31.00%	18.9% 26.3%	16.4% 26.7%	22.5% 31%
	c1-c2/195	PTMKG-WE-C PTMKG-WE-C-M	25.13% 45.32%	25.9% 39.07%	20.87% 37.3%	23.75% 41.9%

By comparing accuracies of different models on the same method for "may_treat" and "may_prevent" in Figure 5, we know that using smodel2 is more accurate than using smodel1, using smodel4 is more accurate than using smodel3, and using smodel4 is more accurate than using smodel1, except for smodel1 for "may_prevent". As shown in Section 4.2, data2 and data4 are the deep extensions of data1 and data3, respectively, and data4 is the breadth expansion of data1. Therefore, extensions of data in depth or breadth can improve the qualities of embedding models, then affect the accuracy of the final algorithm.

Table 6 shows accuracies of PTMKG-WE-C and PTMKG-WE-C-M on different models for "may_treat" and "may_prevent". "No./triples" denotes the ID of a given cluster and the number of triples included in the cluster. Different clusters contain different numbers of triples and obtain different accuracies. For most clusters, accuracies of PTMKG-WE-C-M are better than those of PTMKG-WE-C on a same model. Similarly, in most clusters, under the same method, the accuracy on smodel2 is higher than that on smodel1, that of smodel4 is higher than that of smodel3, and that of smodel4 is higher than that of smodel1. c4/73, c5/29, and c8/16 are special cases, both of which are violated. The main reason may be that only one or several diseases are involved in these triples in the three clusters. For example, both c4/73 and c8/16 contain only one disease labeled "pain" and "diarrhea". In order to objectively express accuracies of clustering algorithms, the weighted average method can be used to calculate the final comprehensive accuracy, which is represented as c1–c10/1585 in Table 6. For a given cluster, its weight value is the ratio of the number of triples in the cluster and the sum of triples included all clusters. The accuracy of every cluster for the same method varies greatly. For example, the accuracy of c4/73 on PTMKG-WE-C-M is 89.04%, and the accuracy of c3/599 on the same method is only 18.67%. Through detailed analysis of the triples in these clusters, it is found that triples in clusters with higher accuracies have a common feature. The unknown tail entity to be predicted is closely related to the tail entity in an existing reference triple, such as two tail entities are the same entity or have an inclusion relationship with each other in a given Knowledge Graph. For example, only one disease, "pain", is involved in 73 triples in c4. On the contrary, in the clusters with lower accuracy, the unknown tail entity to be predicted is basically independent of the tail entity in the reference triple. For example, a large number of (more than 100) unrelated diseases, such as "trypanosomiasis", "candidiasis", "alcoholism", "dyskinesias", are involved in 599 triples in c3. The same problem exists in c6/210 and c7/355; a large number of unrelated diseases (more than 100) is included these two clusters. This trend shows that the accuracy of PTMKG-WE is closely related to the form of prior knowledge. When there is a similar or inclusive relationship between the predicted tail entity and the prior knowledge tail entity, the prediction accuracy is significantly improved.

In addition to accuracy, the ranking results are also important. We selected 10 samples from every cluster of "may_treat" in Table 6 and obtained MRR and MRR+ for PTMKG-WE-C and PTMKG-WE-C-M in smodel1, as shown in Table 7. For all clusters, MRR of PTMKG-WE-C-M has a significant improvement over that of PTMKG-WE-C. For example, comparing MRR of PTMKG-WE-C in c1, that of PTMKG-WE-C-M in c1 is raised from around 0.27 to 0.374. For the same method, MRR+ is generally better than MRR in most clusters. So, in statistical results, ignoring these tail-entity words in the existing samples has a positive impact on the ranking results.

However, in addition to the correct results, there are some noises in the top 20 candidates. Noise words are analyzed in detail. It is found that most of these words belong to the synonym of existing samples, such as, A, B, C in the formula for predicting the unknown tail entity X, that is $v(X) = v(C) - (v(A_j) - v(B_j))$. For example, for $v(X) = v(dalteparin) - (v(globulin) - v(rubella))$, the top 20 candidates from embeddings are as follows: "dalteparin—0.88262767, fondaparinux—0.6558497, enoxaparin—0.65285647, thromboprophylaxis—0.6289057, ... ". The top three are synonyms of "dalteparin". These values can be filtered in advance. Filtering rules will be the main focus of our work in the future.

Table 7. MRR and MRR+ for PTMKG-WE-C and PTMKG-WE-C-M on smodel1.

No./Triples	Methods	MRR	MRR+
c1/19	PTMKG-WE-C	0.27	0.28
	PTMKG-WE-C-M	0.374	0.5
c2/121	PTMKG-WE-C	0.355	0.383
	PTMKG-WE-C-M	0.734	0.742
c3/599	PTMKG-WE-C	0.16	0.169
	PTMKG-WE-C-M	0.195	0.213
c4/73	PTMKG-WE-C	0.327	0.327
	PTMKG-WE-C-M	1	1
c5/29	PTMKG-WE-C	0.596	0.616
	PTMKG-WE-C-M	0.816	1
c6/210	PTMKG-WE-C	0.168	0.177
	PTMKG-WE-C-M	0.364	0.37
c7/355	PTMKG-WE-C	0.357	0.357
	PTMKG-WE-C-M	0.361	0.361
c8/16	PTMKG-WE-C	0.183	0.188
	PTMKG-WE-C-M	1	1
c9/99	PTMKG-WE-C	0.31	0.313
	PTMKG-WE-C-M	0.487	0.563
c10/64	PTMKG-WE-C	0.116	0.116
	PTMKG-WE-C-M	0.279	0.281

4.4.3. Experimental Results for Double Words

Figure 6 shows accuracies of "may_treat" and "may_prevent" on different methods and different models. The accuracies of two clustering methods denote the final comprehensive accuracy in Table 8. By comparing accuracies of PTMKG-WE and PTMKG-WE-M, it can be found that the accuracy of PTMKG-WE-M is nearly 1 time higher than that of PTMKG-WE. The accuracy of PTMKG-WE-C-M is twice as high as that of PTMKG-WE-C. This means that the embedding feature of a given relationship represented by the mean of samples, as performed in PTMKG-WE-M and PTMKG-WE-C-M, is very effective. Although the accuracy increase of PTMKG-WE-C is different from that of PTMKG-WE on different models, the trend of the accuracy increase is still obvious. This explains the extraordinary effect of subdividing the relationship in sub-relations through clustering in PTMKG-WE-C. When clustering and mean effect are superimposed on one method, the accuracy of this method is greatly improved. As shown in Figure 6, PTMKG-WE-C-M is the superposition of four methods, and obtains the best accuracy. By comparing accuracies on different models for the same method in Figure 6, we know that using c34dmodel2/c38dmodel2 is more accurate than using c34dmodel1/c38dmodel1, using c34dmodel4/c38dmodel4 is more accurate than using c34dmodel3/c38dmodel3, and using c34dmodel4/c38dmodel4 is more accurate than using c34dmodel1/c38dmodel1, with a few exceptions. As shown in Section 4.2, dmodel2 and dmodel4 are the deep extensions of dmodel1 and dmodel3, respectively, and dmodel4 is the breadth expansion of dmodel1. Therefore, extensions of data in depth or breadth can improve qualities of embedding models, then affect the accuracy of the final algorithm for double words.

Table 8 shows accuracies of PTMKG-WE-C and PTMKG-WE-C-M for "may_prevent" and "may_treat" on different models when min−filter = five and TopN = 20. "No./triples" denotes a cluster formed after clustering and the number of triples included in the cluster. Different clusters contain different number of triples, and obtain different accuracy. In order to objectively express the accuracy of the clustering algorithm, the weighted average method can be used to calculate the final comprehensive accuracy, such as the "c1-c10/3078" in Table 8. For a given cluster, its weight value is the ratio of the number of triples in the cluster and the total number of all triples processed. The accuracy of every cluster for the same method varies greatly. For example, the accuracy of c1/762 on PTMKG-WE-C-M is

only 7.67%, and the accuracy of c3/466 on the same method is 33.33%. Similar to the change form of single words, when the models are the same, the accuracy of PTMKG-WE-C-M is much better than that of PTMKG-WE-C for most clusters. Through detailed analysis of the triples in these clusters, it was found that triples in clusters have the same trend as single word triples. That is, the clusters with higher accuracy include a small number of related tail entities, while the clusters with lower accuracy include a large number of unrelated tail entities. However, since more tail entities are included in each double-word cluster, the overall accuracy of double words is smaller than that of single words. This shows that when predicting new knowledge, the number of existing samples is not more beneficial, and closely related samples are more valuable.

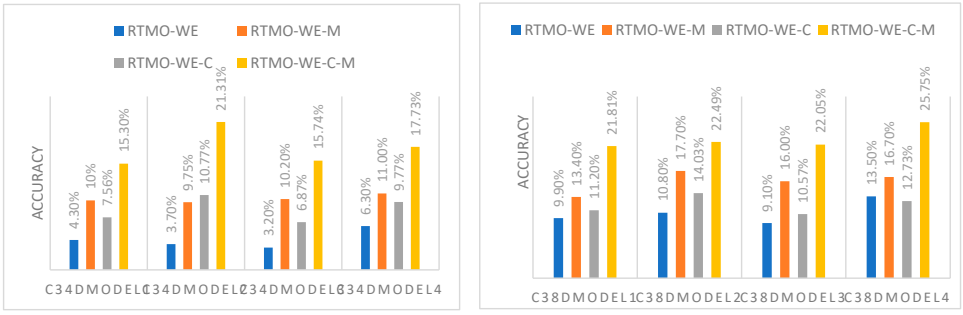

Figure 6. Accuracy comparison of four algorithms on a same model. Among, (**a**,**b**) show accuracies of four algorithms on a same model for "may_treat" and "may_prevent"; (**a**) may_treat; (**b**) may_prevent.

Table 8. The accuracies of PTMKG-WE-C and PTMKG-WE-C-M on different models.

Relationship	No./Triples	Methods	Accuracies			
			dd1	dd2	dd3	dd4
may_treat	c1/762	PTMKG-WE-C PTMKG-WE-C-M	4.64% 7.67%8	6.00% 11.00%	2.22% 5.0%	6.44% 11.67%
	c2/278	PTMKG-WE-C PTMKG-WE-C-M	10.61% 17%	12.6% 19.00%	10.60% 19.00%	16.67% 23.00%
	c3/466	PTMKG-WE-C PTMKG-WE-C-M	16.24% 33.33%	17.5% 37.7%	17.37% 43.00%	17.29% 27.67%
	c4/342	PTMKG-WE-C PTMKG-WE-C-M	2.83% 11.67%	5.30% 23.00%	2.0% 5.7%	3.31% 19.00%
	c5/429	PTMKG-WE-C PTMKG-WE-C-M	2.77% 8%	8.30% 18.3%	2.00% 5.7%	4.61% 12.33%
	c6/324	PTMKG-WE-C PTMKG-WE-C-M	2.01% 4.33%	3.70% 12.7%	1.88% 7.3%	4.64% 9.67%
	c7/794	PTMKG-WE-C PTMKG-WE-C-M	8.01% 15.33%	11.8% 19.3%	8.45% 15.7%	12.79% 19.67%
	c8/301	PTMKG-WE-C PTMKG-WE-C-M	13.65% 31.67%	19.26% 37.00%	11.44% 29.7%	7.31% 18.33%
	c9/152	PTMKG-WE-C PTMKG-WE-C-M	8.33% 12.67%	16.93% 24.3%	7.47% 17.3%	13.96% 21.67%
	c10/30	PTMKG-WE-C PTMKG-WE-C-M	22.99% 23.33%	40.00% 50.00%	13.9% 40.00%	38.74% 40.00%
	c1-c10/3878	PTMKG-WE-C PTMKG-WE-C-M	7.56% 15.3%	10.77% 21.31%	6.87% 15.74%	9.77% 17.74%
may_prevent	c1/466	PTMKG-WE-C PTMKG-WE-C-M	7.7% 14%	10.94% 14.33%	8.5% 16.00%	13.4% 18.7%
	c2/51	PTMKG-WE-C PTMKG-WE-C-M	17.5% 68.6%	27.45% 68.62%	9.3% 51.00%	11.5% 62.7%
	c3/19	PTMKG-WE-C PTMKG-WE-C-M	81.3% 89.5%	54.39% 100.00%	65.8% 94.7%	0% 100.00%
	c1-c3/536	PTMKG-WE-C PTMKG-WE-C-M	11.2% 21.81%	14.03% 22.49%	10.57% 22.05%	12.73% 25.75%

5. Conclusions

In order to solve the problem of entity completion in Medical Knowledge Graphs, this paper proposes a framework for predicting RDF triples for Medical Knowledge Graphs based on word embeddings (named PTMKG-WE). The framework first formalizes existing samples for a given relationship from a Medical Knowledge Graph as prior knowledge. Then, it trains word embeddings from big medical data through Word2vec. Third, the framework can acquire candidate triples from word embeddings based on analogies with existing samples. From the point of view of a given relationship feature, the analogy equation can express the embedding feature about the specific relationship. In order to refine a semantic relationship, this paper proposes two ways to improve the embedding feature of a given relationship. One way is used to refine a relational semantic by clustering existing triples samples. The other way is used to improve the embedding feature of a relationship through the mean of existing samples. Finally, the paper collects PubMed data and NDF-RT and finishes a series of experiments to assess the framework. In experiments, single-word and double-word triples in NDF-RT are considered, respectively. The experimental results show that the framework in the paper and two improvement methods can be used to predict new triples for Medical Knowledge Graphs, when medical data are sufficiently abundant and Knowledge Graphs have appropriate prior knowledge. It is more effective to predict new triples closely related to prior knowledge, such as reference triples. Under same parameter settings, the semantic precision of word embeddings can be improved by extending the breadth and depth of data, and accuracies of the framework in this paper can be further improved in most cases. Thus, collecting and training big medical data is a viable way to learn more useful knowledge.

In this paper, we only predict an unknown tail-entity for a given head-entity in a triple. In the future, we need to comprehensively learn an unknown head-entity for a given tail-entity in a triple. The current framework only considers the analogy of embedding features, and prediction accuracies are very limited. Next, we should consider combining more prior knowledge and more expressive models, such as BERT or GTP. Another limitation is that this paper only considers the accuracies of new Knowledge Graphs. The consistency and conciseness of Knowledge Graphs are not really considered in this paper, and will be mainly considered in next studies.

Author Contributions: Conceptualization, J.L. and M.G.; methodology, M.G.; software, F.C.; validation, M.G. and F.C.; formal analysis, M.G.; investigation, M.G.; resources, M.G.; data curation, M.G.; writing—original draft preparation, M.G.; writing—review and editing, M.G.; visualization, M.G.; supervision, M.G.; project administration, M.G.; funding acquisition, M.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Beijing Natural Science Foundation, grant number 4192007.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: "PubMed accessed on 1 October 2016" at https://pubmed.ncbi.nlm.nih.gov/. "PubMed Central Open Access Subset (PMCOAS) accessed on 1 October 2016" at https://www.ncbi.nlm.nih.gov/pmc/. "National Drug File-Reference Terminology (NDF-RT) accessed on 1 October 2016 " at https://bioportal.bioontology.org/ontologies/NDF-RT.

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

References

1. Gu, P.; Chen, H.; Yu, T. Ontology-oriented diagnostic system for traditional Chinese medicine based on relation refinement. *Comput. Math. Methods Med.* **2013**, *57*, 317803. [CrossRef] [PubMed]
2. Jalali, V.; Matash Borujerdi, M.R. Mohammad. A unified architecture for biomedical search engines based on semantic web technologies. *J. Med. Syst.* **2011**, *35*, 237–249. [CrossRef] [PubMed]
3. Midhunlal, M.; Gopika, M. Xmqas—An ontology based medical question answering system. *Int. J. Adv. Res. Comput. Commun. Eng.* **2016**, *5*, 829–832.

4. Coulet, A.; Shah, N.H.; Garten, Y.; Musen, M.; Altman, R.B. Using text to build semantic networks for pharmacogenomics. *J. Biomed. Inform.* **2010**, *43*, 1009–1019. [CrossRef] [PubMed]
5. Ben Abacha, A.; Zweigenbaum, P. Automatic extraction of semantic relations between medical entities: A rule based approach. *J. Biomed. Semant.* **2011**, *2* (Suppl. 5), 1–11. [CrossRef] [PubMed]
6. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781v3.
7. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.
8. Jevlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
9. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
10. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv* **2019**, arXiv:1906.08237.
11. Su, M.; Su, H. Deep learning for knowledge graph completion with XLNET. In Proceedings of the ICDLT 2021, Qingdao, China, 23–25 July 2021.
12. Jaradeh, M.Y.; Singh, K.; Stocker, M.; Auer, S. Triple classification for scholarly knowledge graph completion. In Proceedings of the K-CAP'21, Virtual Event, 2–3 December 2021.
13. Minarro-Gimenez, J.A.; Marın-Alonso, O.; Samwald, M. Applying deep learning techniques on medical corpora from the world wide web: A prototypical system and evaluation. *arXiv* **2015**, arXiv:1502.03682.
14. Casteleiro, M.A.; Demetriou, G.; Read, W.J.; Prieto, M.J.F.; MasedaFernandez, D.; Nenadic, G.; Klein, J.; Keane, J.A.; Stevens, R. Deep learning meets semantic web: A feasibility study with the cardiovascular disease ontology and pubmed citations. In Proceedings of the 7th Workshop on Ontologies and Data in Life Sciences 2016, Halle (Saale), Germany, 29–30 September 2016.
15. Lan, Y.; He, S.; Liu, K.; Zeng, X.; Liu, S.; Zhao, J. Path-based knowledge reasoning with textual semantic information for medical knowledge graph completion. *BMC Med. Inform. Decis. Mak.* **2021**, *21*, 1–10. [CrossRef] [PubMed]
16. He, B.; Zhou, D.; Xiao, J.; Jiang, X.; Liu, Q.; Yuan, N.J.; Xu, T. BERT-MK: Integrating graph contextualized knowledge into pre-trained language models. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Hong Kong, China, 16–20 November 2020.
17. Arnaud, É.; Elbattah, M.; Gignon, M.; Dequen, G. Learning embeddings from free-text triage notes using pretrained transformer models. In Proceedings of the 15th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2022), Vienna, Austria, 9–11 February 2022.
18. Chang, D.; Hong, W.S.; Taylor, R.A. Generating contextual embeddings for emergency department chief complaints. *JAMIA Open* **2020**, *3*, 160–166. [CrossRef] [PubMed]