A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization

Abstract

In this paper, we introduce an embedding model, named CapsE, exploring a capsule network to model relationship triples (subject, relation, object). Our CapsE represents each triple as a 3-column matrix where each column vector represents the embedding of an element in the triple. This 3-column matrix is then fed to a convolution layer where multiple filters are operated to generate different feature maps. These feature maps are reconstructed into corresponding capsules which are then routed to another capsule to produce a continuous vector. The length of this vector is used to measure the plausibility score of the triple. Our proposed CapsE obtains better performance than previous state-of-the-art embedding models for knowledge graph completion on two benchmark datasets WN18RR and FB15k-237, and outperforms strong search personalization baselines on SEARCH17.

Introduction

Knowledge graphs (KGs) containing relationship triples (subject, relation, object), denoted as (s, r, o), are the useful resources for many NLP and especially information retrieval applications such as semantic search and question answering BIBREF0 . However, large knowledge graphs, even containing billions of triples, are still incomplete, i.e., missing a lot of valid triples BIBREF1 . Therefore, much research efforts have focused on the knowledge graph completion task which aims to predict missing triples in KGs, i.e., predicting whether a triple not in KGs is likely to be valid or not BIBREF2 , BIBREF3 , BIBREF4 . To this end, many embedding models have been proposed to learn vector representations for entities (i.e., subject/head entity and object/tail entity) and relations in KGs, and obtained state-of-the-art results as summarized by BIBREF5 and BIBREF6 . These embedding models score triples (s, r, o), such that valid triples have higher plausibility scores than invalid ones BIBREF2 , BIBREF3 , BIBREF4 . For example, in the context of KGs, the score for (Melbourne, cityOf, Australia) is higher than the score for (Melbourne, cityOf, United Kingdom).

Triple modeling is applied not only to the KG completion, but also for other tasks which can be formulated as a triple-based prediction problem. An example is in search personalization, one would aim to tailor search results to each specific user based on the user's personal interests and preferences BIBREF7 , BIBREF8 , BIBREF9 , BIBREF10 , BIBREF11 , BIBREF12 . Here the triples can be formulated as (submitted query, user profile, returned document) and used to re-rank documents returned to a user given an input query, by employing an existing KG embedding method such as TransE BIBREF3 , as proposed by BIBREF12 . Previous studies have shown the effectiveness of modeling triple for either KG completion or search personalization. However, there has been no single study investigating the performance on both tasks.

Conventional embedding models, such as TransE BIBREF3 , DISTMULT BIBREF13 and ComplEx BIBREF14 , use addition, subtraction or simple multiplication operators, thus only capture the linear relationships between entities. Recent research has raised interest in applying deep neural networks to triple-based prediction problems. For example, BIBREF15 proposed ConvKB—a convolutional neural network (CNN)-based model for KG completion and achieved state-of-the-art results. Most of KG embedding models are constructed to modeling entries at the same dimension of the given triple, where presumably each dimension captures some relation-specific attribute of entities. To the best of our knowledge, however, none of the existing models has a “deep” architecture for modeling the entries in a triple at the same dimension.

BIBREF16 introduced capsule networks (CapsNet) that employ capsules (i.e., each capsule is a group of neurons) to capture entities in images and then uses a routing process to specify connections from capsules in a layer to those in the next layer. Hence CapsNet could encode the intrinsic spatial relationship between a part and a whole constituting viewpoint invariant knowledge that automatically generalizes to novel viewpoints. Each capsule accounts for capturing variations of an object or object part in the image, which can be efficiently visualized. Our high-level hypothesis is that embedding entries at the same dimension of the triple also have these variations, although it is not straightforward to be visually examined.

To that end, we introduce CapsE to explore a novel application of CapsNet on triple-based data for two problems: KG completion and search personalization. Different from the traditional modeling design of CapsNet where capsules are constructed by splitting feature maps, we use capsules to model the entries at the same dimension in the entity and relation embeddings. In our CapsE, INLINEFORM0 , INLINEFORM1 and INLINEFORM2 are unique INLINEFORM3 -dimensional embeddings of INLINEFORM4 , INLINEFORM5 and INLINEFORM6 , respectively. The embedding triple [INLINEFORM7 , INLINEFORM8 , INLINEFORM9] of (s, r, o) is fed to the convolution layer where multiple filters of the same INLINEFORM10 shape are repeatedly operated over every row of the matrix to produce INLINEFORM11 -dimensional feature maps. Entries at the same dimension from all feature maps are then encapsulated into a capsule. Thus, each capsule can encode many characteristics in the embedding triple to represent the entries at the corresponding dimension. These capsules are then routed to another capsule which outputs a continuous vector whose length is used as a score for the triple. Finally, this score is used to predict whether the triple (s, r, o) is valid or not.

In summary, our main contributions from this paper are as follows:

INLINEFORM0 We propose an embedding model CapsE using the capsule network BIBREF16 for modeling relationship triples. To our best of knowledge, our work is the first consideration of exploring the capsule network to knowledge graph completion and search personalization.

INLINEFORM0 We evaluate our CapsE for knowledge graph completion on two benchmark datasets WN18RR BIBREF17 and FB15k-237 BIBREF18 . CapsE obtains the best mean rank on WN18RR and the highest mean reciprocal rank and highest Hits@10 on FB15k-237.

INLINEFORM0 We restate the prospective strategy of expanding the triple embedding models to improve the ranking quality of the search personalization systems. We adapt our model to search personalization and evaluate on SEARCH17 BIBREF12 – a dataset of the web search query logs. Experimental results show that our CapsE achieves the new state-of-the-art results with significant improvements over strong baselines.

The proposed CapsE

Let INLINEFORM0 be a collection of valid factual triples in the form of (subject, relation, object) denoted as (s, r, o). Embedding models aim to define a score function giving a score for each triple, such that valid triples receive higher scores than invalid triples.

We denote INLINEFORM0 , INLINEFORM1 and INLINEFORM2 as the INLINEFORM3 -dimensional embeddings of INLINEFORM4 , INLINEFORM5 and INLINEFORM6 , respectively. In our proposed CapsE, we follow BIBREF15 to view each embedding triple [INLINEFORM7 , INLINEFORM8 , INLINEFORM9] as a matrix INLINEFORM10 , and denote INLINEFORM11 as the INLINEFORM12 -th row of INLINEFORM13 . We use a filter INLINEFORM14 operated on the convolution layer. This filter INLINEFORM15 is repeatedly operated over every row of INLINEFORM16 to generate a feature map INLINEFORM17 , in which INLINEFORM18 where INLINEFORM19 denotes a dot product, INLINEFORM20 is a bias term and INLINEFORM21 is a non-linear activation function such as ReLU. Our model uses multiple filters INLINEFORM22 to generate feature maps. We denote INLINEFORM23 as the set of filters and INLINEFORM24 as the number of filters, thus we have INLINEFORM25 INLINEFORM26

-dimensional feature maps, for which each feature map can capture one single characteristic among entries at the same dimension.

We build our CapsE with two single capsule layers for a simplified architecture. In the first layer, we construct INLINEFORM0 capsules, wherein entries at the same dimension from all feature maps are encapsulated into a corresponding capsule. Therefore, each capsule can capture many characteristics among the entries at the corresponding dimension in the embedding triple. These characteristics are generalized into one capsule in the second layer which produces a vector output whose length is used as the score for the triple.

The first capsule layer consists of INLINEFORM0 capsules, for which each capsule INLINEFORM1 has a vector output INLINEFORM2 . Vector outputs INLINEFORM3 are multiplied by weight matrices INLINEFORM4 to produce vectors INLINEFORM5 which are summed to produce a vector input INLINEFORM6 to the capsule in the second layer. The capsule then performs the non-linear squashing function to produce a vector output INLINEFORM7 : DISPLAYFORM0

where INLINEFORM0 , and INLINEFORM1 are coupling coefficients determined by the routing process as presented in Algorithm SECREF2 . Because there is one capsule in the second layer, we make only one difference in the routing process proposed by BIBREF16 , for which we apply the INLINEFORM2 in a direction from all capsules in the previous layer to each of capsules in the next layer.

[ht] 1.25

all capsule i INLINEFORM0 the first layer INLINEFORM1 0 INLINEFORM2 = 1, 2, ..., m INLINEFORM3 INLINEFORM4

INLINEFORM0

all capsule i INLINEFORM0 the first layer INLINEFORM1 The routing process is extended from BIBREF16 .

We illustrate our proposed model in Figure FIGREF1 where embedding size: INLINEFORM0 , the number of filters: INLINEFORM1 , the number of neurons within the capsules in the first layer is equal to INLINEFORM2 , and the number of neurons within the capsule in the second layer: INLINEFORM3 . The length of the vector output INLINEFORM4 is used as the score for the input triple.

Formally, we define the score function INLINEFORM0 for the triple INLINEFORM1 as follows: DISPLAYFORM0

where the set of filters INLINEFORM0 is shared parameters in the convolution layer; INLINEFORM1 denotes a convolution operator; and INLINEFORM2 denotes a capsule network operator. We use the Adam optimizer BIBREF19 to train CapsE by minimizing the loss function BIBREF14 , BIBREF15 as follows: DISPLAYFORM0

INLINEFORM0

here INLINEFORM0 and INLINEFORM1 are collections of valid and invalid triples, respectively. INLINEFORM2 is generated by corrupting valid triples in INLINEFORM3 .

Knowledge graph completion evaluation

In the knowledge graph completion task BIBREF3 , the goal is to predict a missing entity given a relation and another entity, i.e, inferring a head entity INLINEFORM0 given INLINEFORM1 or inferring a tail entity INLINEFORM2 given INLINEFORM3 . The results are calculated based on ranking the scores produced by the score function INLINEFORM4 on test triples.

Experimental setup

Datasets: We use two recent benchmark datasets WN18RR BIBREF17 and FB15k-237 BIBREF18 . These two datasets are created to avoid reversible relation problems, thus the prediction task becomes more realistic and hence more challenging BIBREF18 . Table TABREF7 presents the statistics of WN18RR and FB15k-237.

Evaluation protocol: Following BIBREF3 , for each valid test triple INLINEFORM0 , we replace either INLINEFORM1 or INLINEFORM2 by each of all other entities to create a set of corrupted triples. We use the “Filtered” setting protocol BIBREF3 , i.e., not taking any corrupted triples that appear in the KG into accounts. We rank the valid test triple and corrupted triples in descending order of their scores. We employ evaluation metrics: mean rank (MR), mean reciprocal rank (MRR) and Hits@10 (i.e., the proportion of the valid test triples ranking in top 10 predictions). Lower MR, higher MRR or higher Hits@10 indicate better performance. Final scores on the test set are reported for the model obtaining the highest Hits@10 on the validation set.

Training protocol: We use the common Bernoulli strategy BIBREF20 , BIBREF21 when sampling invalid triples. For WN18RR, BIBREF22 found a strong evidence to support the necessity of a WordNet-related semantic setup, in which they averaged pre-trained word embeddings for word surface forms within the WordNet to create synset embeddings, and then used these synset embeddings to initialize entity embeddings for training their TransE association model. We follow this evidence in using the pre-trained 100-dimensional Glove word embeddings BIBREF23 to train a TransE model on WN18RR.

We employ the TransE and ConvKB implementations provided by BIBREF24 and BIBREF15 . For ConvKB, we use a new process of training up to 100 epochs and monitor the Hits@10 score after every 10 training epochs to choose optimal hyper-parameters with the Adam initial learning rate in INLINEFORM0 and the number of filters INLINEFORM1 in INLINEFORM2 . We obtain the highest Hits@10 scores on the validation set when using N= 400 and the initial learning rate INLINEFORM3 on WN18RR; and N= 100 and the initial learning rate INLINEFORM4 on FB15k-237.

Like in ConvKB, we use the same pre-trained entity and relation embeddings produced by TransE to initialize entity and relation embeddings in our CapsE for both WN18RR and FB15k-237 (INLINEFORM0). We set the batch size to 128, the number of neurons within the capsule in the second capsule layer to 10 (INLINEFORM1), and the number of iterations in the routing algorithm INLINEFORM2 in INLINEFORM3 . We run CapsE up to 50 epochs and monitor the Hits@10 score after each 10 training epochs to choose optimal hyper-parameters. The highest Hits@10 scores for our CapsE on the validation set are obtained when using INLINEFORM4 , INLINEFORM5 and the initial learning rate at INLINEFORM6 on WN18RR; and INLINEFORM7 , INLINEFORM8 and the initial learning rate at INLINEFORM9 on FB15k-237.

Dataset: We use the SEARCH17 dataset BIBREF12 of query logs of 106 users collected by a large-scale web search engine. A log entity consists of a user identifier, a query, top-10 ranked documents returned by the search engine and clicked documents along with the user's dwell time. BIBREF12 constructed short-term (session-based) user profiles and used the profiles to personalize the returned results. They then employed the SAT criteria BIBREF26 to identify whether a returned document is relevant from the query logs as either a clicked document with a dwell time of at least 30 seconds or the last clicked document in a search session (i.e., a SAT click). After that, they assigned a INLINEFORM0 label to a returned document if it is a SAT click and also assigned INLINEFORM1 labels to the remaining top-10 documents. The rank position of the INLINEFORM2 labeled documents is used as the ground truth to evaluate the search performance before and after re-ranking.

The dataset was uniformly split into the training, validation and test sets. This split is for the purpose of using historical data in the training set to predict new data in the test set BIBREF12 . The training, validation and test sets consist of 5,658, 1,184 and 1,210 relevant (i.e., valid) triples; and 40,239, 7,882 and 8,540 irrelevant (i.e., invalid) triples, respectively.

Evaluation protocol: Our CapsE is used to re-rank the original list of documents returned by a search engine as follows: (i) We train our model and employ the trained model to calculate the score for each INLINEFORM0 triple. (ii) We then sort the scores in the descending order to obtain a new ranked list. To evaluate the performance of our proposed model, we use two standard evaluation metrics: mean reciprocal rank (MRR) and Hits@1. For each metric, the higher value indicates better ranking performance.

We compare CapsE with the following baselines using the same experimental setup: (1) SE: The original rank is returned by the search engine. (2) CI BIBREF27 : This baseline uses a personalized navigation method based on previously clicking returned documents. (3) SP BIBREF9 , BIBREF11 : A search personalization method makes use of the session-based user profiles. (4) Following BIBREF12 , we use TransE as a strong baseline model for the search personalization task. Previous work shows that the well-known embedding model TransE, despite its simplicity, obtains very competitive results for the knowledge graph completion BIBREF28 , BIBREF29 , BIBREF14 , BIBREF30 , BIBREF15 . (5) The CNN-based model ConvKB is the most closely related model to our CapsE.

Embedding initialization: We follow BIBREF12 to initialize user profile, query and document embeddings for the baselines TransE and ConvKB, and our CapsE.

We train a LDA topic model BIBREF31 with 200 topics only on the relevant documents (i.e., SAT clicks) extracted from the query logs. We then use the trained LDA model to infer the probability distribution over topics for every returned document. We use the topic proportion vector of each document as its document embedding (i.e. INLINEFORM0). In particular, the INLINEFORM1 element (INLINEFORM2) of the vector embedding for document INLINEFORM3 is: INLINEFORM4 where INLINEFORM5 is the probability of the topic INLINEFORM6 given the document INLINEFORM7 .

We also represent each query by a probability distribution vector over topics. Let INLINEFORM0 be the set of top INLINEFORM1 ranked documents returned for a query INLINEFORM2 (here, INLINEFORM3). The INLINEFORM4 element of the vector embedding for query INLINEFORM5 is defined as in BIBREF12 : INLINEFORM6 , where INLINEFORM7 is the exponential decay function of INLINEFORM8 which is the rank of INLINEFORM9 in INLINEFORM10 . And INLINEFORM11 is the decay hyper-parameter (INLINEFORM12). Following BIBREF12 , we use INLINEFORM13 . Note that if we learn query and document embeddings during training, the models will overfit to the data and will not work for new queries and documents. Thus, after the initialization process, we fix (i.e., not updating) query and document embeddings during training for TransE, ConvKB and CapsE.

In addition, as mentioned by BIBREF9 , the more recently clicked document expresses more about the user current search interest. Hence, we make use of the user clicked documents in the training set with the temporal weighting scheme proposed by BIBREF11 to initialize user profile embeddings for the three embedding models.

Hyper-parameter tuning: For our CapsE model, we set batch size to 128, and also the number of neurons within the capsule in the second capsule layer to 10 (INLINEFORM0). The number of iterations in the routing algorithm is set to 1 (INLINEFORM1). For the training model, we use the Adam optimizer with the initial learning rate INLINEFORM2 INLINEFORM3 INLINEFORM4 INLINEFORM5 INLINEFORM6

INLINEFORM7 . We also use ReLU as the activation function INLINEFORM8 . We select the number of filters INLINEFORM9 . We run the model up to 200 epochs and perform a grid search to choose optimal hyper-parameters on the validation set. We monitor the MRR score after each training epoch and obtain the highest MRR score on the validation set when using INLINEFORM10 and the initial learning rate at INLINEFORM11 .

We employ the TransE and ConvKB implementations provided by BIBREF24 and BIBREF15 and then follow their training protocols to tune hyper-parameters for TransE and ConvKB, respectively. We also monitor the MRR score after each training epoch and attain the highest MRR score on the validation set when using margin = 5, INLINEFORM0 -norm and SGD learning rate at INLINEFORM1 for TransE; and INLINEFORM2 and the Adam initial learning rate at INLINEFORM3 for ConvKB.

Main experimental results

Table TABREF10 compares the experimental results of our CapsE with previous state-of-the-art published results, using the same evaluation protocol. Our CapsE performs better than its closely related CNN-based model ConvKB on both experimental datasets (except Hits@10 on WN18RR and MR on FB15k-237), especially on FB15k-237 where our CapsE gains significant improvements of INLINEFORM0 in MRR (which is about 25.1% relative improvement), and INLINEFORM1 % absolute improvement in Hits@10. Table TABREF10 also shows that our CapsE obtains the best MR score on WN18RR and the highest MRR and Hits@10 scores on FB15k-237.

Following BIBREF3 , for each relation INLINEFORM0 in FB15k-237, we calculate the averaged number INLINEFORM1 of head entities per tail entity and the averaged number INLINEFORM2 of tail entities per head entity. If INLINEFORM3 1.5 and INLINEFORM4 1.5, INLINEFORM5 is categorized one-to-one (1-1). If INLINEFORM6 1.5 and INLINEFORM7 1.5, INLINEFORM8 is categorized one-to-many (1-M). If

INLINEFORM9 1.5 and INLINEFORM10 1.5, INLINEFORM11 is categorized many-to-one (M-1). If INLINEFORM12 1.5 and INLINEFORM13 1.5, INLINEFORM14 is categorized many-to-many (M-M). As a result, 17, 26, 81 and 113 relations are labelled 1-1, 1-M, M-1 and M-M, respectively. And 0.9%, 6.3%, 20.5% and 72.3% of the test triples in FB15k-237 contain 1-1, 1-M, M-1 and M-M relations, respectively.

Figure FIGREF11 shows the Hits@10 and MRR results for predicting head and tail entities w.r.t each relation category on FB15k-237. CapsE works better than ConvKB in predicting entities on the “side M” of triples (e.g., predicting head entities in M-1 and M-M; and predicting tail entities in 1-M and M-M), while ConvKB performs better than CapsE in predicting entities on the “side 1” of triples (i.e., predicting head entities in 1-1 and 1-M; and predicting tail entities in 1-1 and M-1).

Figure FIGREF12 shows the Hits@10 and MRR scores w.r.t each relation on WN18RR. INLINEFORM0 , INLINEFORM1 , INLINEFORM2 and INLINEFORM3 are symmetric relations which can be considered as M-M relations. Our CapsE also performs better than ConvKB on these 4 M-M relations. Thus, results shown in Figures FIGREF11 and FIGREF12 are consistent. These also imply that our CapsE would be a potential candidate for applications which contain many M-M relations such as search personalization.

We see that the length and orientation of each capsule in the first layer can also help to model the important entries in the corresponding dimension, thus CapsE can work well on the “side M” of triples where entities often appear less frequently than others appearing in the “side 1” of triples. Additionally, existing models such as DISTMULT, ComplEx and ConvE can perform well for entities with high frequency, but may not for rare entities with low frequency. These are reasons why our CapsE can be considered as the best one on FB15k-237 and it outperforms most existing models on WN18RR.

Effects of routing iterations: We study how the number of routing iterations affect the performance. Table TABREF13 shows the Hits@10 scores on the WN18RR validation set for a comparison w.r.t each number value of the routing iterations and epochs with the number of filters INLINEFORM0 and the Adam initial learning rate at INLINEFORM1 . We see that the best performance for each setup over each 10 epochs is obtained by setting the number INLINEFORM2 of routing iterations to 1. This indicates the opposite side for knowledge graphs compared to images. In the image classification task, setting the number INLINEFORM3 of iterations in the routing process higher than 1 helps to capture the relative positions of entities in an image (e.g., eyes, nose and mouth) properly. In contrast, this property from images may be only right for the 1-1 relations, but not for the 1-M, M-1 and M-M relations in the KGs because of the high variant of each relation type (e.g., symmetric relations) among different entities.

Search personalization application

Given a user, a submitted query and the documents returned by a search system for that query, our approach is to re-rank the returned documents so that the more relevant documents should be ranked higher. Following BIBREF12 , we represent the relationship between the submitted query, the user and the returned document as a (s, r, o)-like triple (query, user, document). The triple captures how much interest a user puts on a document given a query. Thus, we can evaluate the effectiveness of our CapsE for the search personalization task.

Main results

Table TABREF17 presents the experimental results of the baselines and our model. Embedding models TranE, ConvKB and CapsE produce better ranking performances than traditional learning-to-rank search personalization models CI and SP. This indicates a prospective strategy of expanding the triple embedding models to improve the ranking quality of the search personalization systems. In particular, our MRR and Hits@1 scores are higher than those of TransE (with relative improvements of 14.5% and 22% over TransE, respectively). Specifically, our CapsE achieves the highest performances in both MRR and

Hits@1 (our improvements over all five baselines are statistically significant with INLINEFORM0 using the paired t-test).

To illustrate our training progress, we plot performances of CapsE on the validation set over epochs in Figure FIGREF18 . We observe that the performance is improved with the increase in the number of filters since capsules can encode more useful properties for a large embedding size.

Related work

Other transition-based models extend TransE to additionally use projection vectors or matrices to translate embeddings of INLINEFORM0 and INLINEFORM1 into the vector space of INLINEFORM2 , such as: TransH BIBREF20 , TransR BIBREF21 , TransD BIBREF32 and STransE BIBREF24 . Furthermore, DISTMULT BIBREF13 and ComplEx BIBREF14 use a tri-linear dot product to compute the score for each triple. Moreover, ConvKB BIBREF15 applies convolutional neural network, in which feature maps are concatenated into a single feature vector which is then computed with a weight vector via a dot product to produce the score for the input triple. ConvKB is the most closely related model to our CapsE. See an overview of embedding models for KG completion in BIBREF6 .

For search tasks, unlike classical methods, personalized search systems utilize the historical interactions between the user and the search system, such as submitted queries and clicked documents to tailor returned results to the need of that user BIBREF7 , BIBREF8 . That historical information can be used to build the user profile, which is crucial to an effective search personalization system. Widely used approaches consist of two separated steps: (1) building the user profile from the interactions between the user and the search system; and then (2) learning a ranking function to re-rank the search results using the user profile BIBREF9 , BIBREF33 , BIBREF10 , BIBREF11 . The general goal is to re-rank the documents returned by the search system in such a way that the more relevant documents are ranked higher. In this case, apart from the user profile, dozens of other features have been proposed as the input of a learning-to-rank algorithm BIBREF9 , BIBREF33 . Alternatively, BIBREF12 modeled the potential user-oriented relationship between the submitted query and the returned document by applying TransE to reward higher scores for more relevant documents (e.g., clicked documents). They achieved better performances than the standard ranker as well as competitive search personalization baselines BIBREF27 , BIBREF9 , BIBREF11 .

Conclusion

We propose CapsE—a novel embedding model using the capsule network to model relationship triples for knowledge graph completion and search personalization. Experimental results show that our CapsE outperforms other state-of-the-art models on two benchmark datasets WN18RR and FB15k-237 for the knowledge graph completion. We then show the effectiveness of our CapsE for the search personalization, in which CapsE outperforms the competitive baselines on the dataset SEARCH17 of the web search query logs. In addition, our CapsE is capable to effectively model many-to-many relationships. Our code is available at: https://github.com/daiquocnguyen/CapsE.

Acknowledgement

This research was partially supported by the ARC Discovery Projects DP150100031 and DP160103934. The authors thank Yuval Pinter for assisting us in running his code.