Neural Computing and Applications (2023) 35:787–797
https://doi.org/10.1007/s00521-022-07796-z

ORIGINAL ARTICLE

Regularized online tensor factorization for sparse knowledge graph embeddings

Unai Zulaika[1] · Aitor Almeida[1] · Diego López-de-Ipiña[1]

Received: 17 January 2022 / Accepted: 6 September 2022 / Published online: 21 September 2022

Abstract

Knowledge Graphs represent real-world facts and are used in several applications; however, they are often incomplete and have many missing facts. Link prediction is the task of completing these missing facts from existing ones. Embedding models based on Tensor Factorization attain state-of-the-art results in link prediction. Nevertheless, the embeddings they produce can not be easily interpreted. Inspired by previous work on word embeddings, we propose inducing sparsity in the bilinear tensor factorization model, RESCAL, to build interpretable Knowledge Graph embeddings. To overcome the difficulties that stochastic gradient descent has when producing sparse solutions, we add l_1 regularization to the learning objective by using the generalized Regularized Dual Averaging online optimization algorithm. The proposed method substantially improves the interpretability of the learned embeddings while maintaining competitive performance in the standard metrics.

Keywords Knowledge graph embedding · Sparse learning · Interpretable embeddings

1 Introduction

Knowledge Graphs are large graph stores that represent real-world facts by connecting entities through relations. These facts follow the triple form (e_s, r, e_o) where e_s and e_o are respectively, the subject and object entities and r is the relation between them. Unfortunately, Knowledge Graphs have many missing facts and therefore are not complete. The amount of facts, entities, and relations is vast and thus, containing all of them in a Knowledge Graph is a complex task, e.g., Wikipedia's task is to contain information on all branches of knowledge. Furthermore, most Knowledge Graphs model information areas that are continually evolving and changing, which causes the graphs to be incomplete. Different Machine Learning approaches have been proposed to tackle this issue.

Since Knowledge Graphs can be represented as a third-order binary tensor where each position of the tensor represents whether a fact (triple) is true or false, different algorithms based on Tensor Factorization have been proposed for inferring the missing knowledge. These factorization algorithms [2, 25, 44] produce distributed representations of the entities and relations within the Knowledge Graph, also known as Knowledge Graph Embeddings. Embeddings are then used to predict new facts from the known ones.

Embeddings, when distributed, are, however, difficult to interpret. They are real-valued vectors whose individual dimensions can not be understood, i.e., their latent dimensions have low semantic meaning. Several cognitive arguments based on the economy of storage maintain that a small set of features is not enough to describe every semantic concept and domain in our vocabularies [23, 24, 31]. From the cognitive point of view, some words require more characteristics to be described and others that require less. Besides, certain properties belong to specific semantic domains and sharing properties is not natural for humans. Furthermore, we do not describe concepts based on what they are not since it would be uneconomical; thus, we do not store that a desk is not a vegetable or a glass can not fly. Following these arguments, as authors in [23] presented, the feature set of a representation should only store positive facts, a wide range of feature types, and only

✉ Unai Zulaika
unai.zulaika@deusto.es

[1] Faculty of Engineering - DeustoTech, University of Deusto, Avda. Universidades, 2, 48007 Bilbao, Bizkaia, Spain

a small quantity of these features should be used to describe a concept. To achieve such properties, the feature set of the concepts representations should be non-negative and sparse. In this work, we present a solution to get the sparsity property in training time for Knowledge Graph Embeddings. Nevertheless, while we do not strictly induce non-negativity in our method, it highly enforces embeddings to have high non-negativity values, as we demonstrate in Sect. 4.4.

Other approaches also have been proposed to increase interpretability for Knowledge Graph embeddings. Authors from [5] regularize embeddings incorporating additional entity co-occurrence statistics from text data, thus inducing interpretability in the embeddings. [12] extract weighted Horn rules from embeddings to interpret them. In the area of word embeddings, different authors proposed various models for modelling sparse word-embeddings [23, 34].

A common approach to introduce sparsity in machine learning models is to apply the l_1 regularization to the training process. However, the online optimization with stochastic gradient descent (SGD) cannot produce sparsity for the l_1 regularizer making it a non-trivial task. To overcome such an issue, we propose to use the generalized Regularized Dual Averaging (gRDA) algorithm [6], which can produce l_1 regularized embeddings in an online learning setting. We are, to our knowledge, the first to propose a sparse Knowledge Graph Embedding (KGE) solution without any external knowledge.

To conduct our study, we propose the RESCAL model and introduce the l_1 regularizer into the embeddings applying gRDA. We perform experiments to evaluate both the performance and interpretability of the embeddings. We test the embeddings on the link prediction task and compare the performance to the other models. Furthermore, we evaluate the interpretability based on the word intrusion task. Results demonstrate that our approach remains competitive on link prediction while improving the interpretability of the learned embeddings.

In summary, our main contributions are:

- Presenting the first sparse linear KGE model, sRESCAL, that increases interpretability and remains competitive with the state-of-the-art;
- Adapting the online optimization algorithm generalized Regularized Dual Averaging to an unknown setting and showing its efficiency;
- an extensive evaluation and comparison between dense KGE models and sRESCAL.

2 Related work

The literature has widely studied the representation of KG's entities and relations as continuous vectors in a low-dimensional space. We first review works that produce embeddings based on tensor decomposition methods which are of primary interest to our approach. Then, we present different techniques for interpreting KG embeddings.

2.1 Knowledge graph embedding models

Various tensor factorisation models for link prediction have been proposed in the literature:

RESCAL [25] is a bilinear model which associates entities with a vector that captures latent semantics. Relations are represented as matrices that model pairwise interactions between the latent factors.

DistMult [44] is a specific case of RESCAL where relations are diagonal matrices. Therefore, relations only capture pairwise interactions between the same dimensions. While this model requires fewer parameters, it cannot model asymmetric relations because of its diagonal property.

ComplEx [37] is an extension of DistMult which outputs complex-valued embeddings. This property makes the model asymmetric, improving the performance in non-symmetric relations.

TuckER [2] is a linear model based on the Tucker decomposition [38] and previous linear models can be interpreted as special cases of this. Its power relies on a core tensor which allows multi-task learning by sharing knowledge between entities and relations.

Other models or variations on tensor factorisation such as DUality-induced RegulArizer (DURA) [45], CP decomposition [17], ComplEx [37], KGFP [26], knowledge-driven regularizers for embedding learning [22] or SeAttE [20] are also worth mentioning. Additionally, the survey [21] provides a complete overview of the different models and techniques, and [29] offers a comparative analysis of the approaches.

All of these are based on tensor factorisation techniques and produce distributed representations. However, unlike our model, they do not introduce sparsity in the embeddings, and the semantic meaning of the representations' dimensions is less understandable. Furthermore, our method could be adapted to different models since it relies on the optimisation process rather than the factorisation itself.

2.2 Interpreting knowledge graph embeddings

Authors in the literature have proposed different methods for interpreting KGEs. In [5] authors induce interpretability in the embeddings by adding a measure of coherence as a regularisation term of the overall loss function using additional entity co-occurrence statistics from the text. The coherence measure allows automated evaluation of the quality of topics learned by topic modelling methods by using additional Point-wise Mutual Information (PMI) for word pairs. In [12], authors adopt “pedagogical approaches” to interpret KGEs and extract weighted Horn rules to increase their interpretability. In the work [3], authors present a model that does predict links and decides whether it is a “topical” or a “social” link. A different work, [43], presents a translational model (ITransF) which learns associations between relations and concepts via sparse interpretable attention vectors. Authors in [9] use the topological properties of a network to explain the contribution of particular categories of features in link prediction. In the work [11] authors improve KGEs using background taxonomic information. Finally, in [1] authors analyse the latent structure and semantic meaning of KGEs based on theoretical interpretations of word embeddings. While all of these works try to increase the interpretability of link prediction in Knowledge Graphs, several properties differentiate our approach: it directly increases the interpretability in the embeddings, the process is done at training time, and it does not use external knowledge in the process.

Not only methods for interpreting KGEs, but different approaches for improving KG-related tasks and techniques were proposed in the literature. In [40], the authors propose an ensemble framework to enhance the robustness and trust of knowledge graph completion. Authors in [39] consider a Bayesian reinforcement learning paradigm to harness uncertainty into multi-hop reasoning; in this manner, the method improves interpretability and performance in the task. Authors in [46] incorporate external knowledge with explicitly syntactic and contextual information for the task of aspect-based sentiment analysis.

Techniques for interpreting word embeddings by sparsity mechanism have also been proposed. In [23], authors apply a sparse non-negative matrix factorisation model to produce word embeddings. In the work [10], they use sparse coding techniques to derive sparse word embeddings from dense word representations. Similar to our work, authors from [35] modify the Continuous Bag of Words model and add the l_1 regulariser in online training by employing the Regularized Dual Averaging method. Authors in [34] present a k-sparse denoising autoencoder to produce a sparse non-negative high dimensional projection of word embeddings. Finally, [27] produce word embeddings by an underlying LDA-based generative model, which helps to generate sparse vectors. These approaches are closer to the approach presented in this work since they produce sparse embeddings. However, they all focus on word embeddings which are developed differently. Our approach works on tensor factorisation techniques that can not be used to build word embeddings.

3 Sparse tensor factorization

This section presents our approach for applying the l_1 regularization to the RESCAL model via the gRDA optimization algorithm. First, we introduce the background and the RESCAL model itself, then describe the details of the approach. The same technique can be applied to other KGE models such as TuckER or DistMult.

3.1 Background

Let $\mathcal{E}$ denote the set of all entities and $\mathcal{R}$ the set of all relations present in a knowledge graph. We denote the collection of triples in a KG as $\mathcal{D}$ and each triple is represented as $(e_s, r, e_o) \in \mathcal{D}$, with $e_s, e_o \in \mathcal{E}$ denoting subject and object entities respectively and $r \in \mathcal{R}$ the relation between them.

In the link prediction task the objective is to learn a scoring function ϕ which scores, $s = \phi(e_s, r, e_o) \in \mathbb{R}$, whether a triple is true or false. To complete the task, we observe a subset of all true triples, aiming to score all the missing ones correctly. In this work, we only consider scoring functions given by a tensor factorization technique, e.g., RESCAL.

3.2 RESCAL model

The RESCAL model [25] is a relaxed version of the DEDICOM [13] matrix factorization method, which decomposes a matrix into two matrices that provide an asymmetric relation between entities.

In the case of KGs the tensor is three-mode, therefore the original tensor $\mathcal{X} \in \mathbb{R}^{n_e \times n_e \times n_r}$, is decomposed by RESCAL into an entity matrix $\mathbf{A} \in \mathbb{R}^{n_e \times d}$ and k relation matrices $\mathbf{R}_k \in \mathbb{R}^{d \times d}$ where n_e represents the number of entities in the KG and d is a hyperpameter for the embedding dimensionality. Thus, the k-th slice of the tensor is factorized as

$$\mathcal{X}_k \approx A\mathbf{R}_k\mathbf{A}^T, \text{for } k = 1, \ldots, n_r \qquad (1)$$

where n_r is the number of slices (relations) in the tensor.

RESCAL is a latent feature model which scores triples by the interaction of the latent features of the subject and object entities. The scoring is given by

$$\phi(e_s, r, e_o) = \mathbf{e}_s^T \mathbf{R}_k \mathbf{e}_o, \tag{2}$$

where $\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^d$ are respectively the subject and object embeddings from the entity embedding matrix $\mathbf{E}$ ($\mathbf{A}$ in Eq. 1), and $\mathbf{R}_k$ is the asymmetric relation matrix corresponding to the k-th relation in the KG.

3.3 Sparse RESCAL

We aim to introduce sparsity into the embeddings via l_1 regularization. As we know, the l_1 regularization applies a penalty in the learned weights, which makes them push towards 0, leading to finding sparse solutions for the embeddings.

The RESCAL model was originally [25] optimized using the Alternating Least Squares (ALS) method. Nevertheless, different works have pointed out that newer training techniques improve model performance [15, 18]. Furthermore, in a recent work [30] that provides an extensive analysis on KG models and their training processes, authors present the Adam [16] method to be the best for training RESCAL. However, as Adam is a stochastic gradient descent (SGD) extension, it can not produce sparse solutions directly applying l_1 regularization to the loss function. The minor updates which SGD makes in the vectors' values difficult the output of many zero entries [8], i.e., it is quite challenging to get two float numbers to add up and equal zero. To overcome this issue, we propose using generalized Regularized Dual Averaging (gRDA) for optimizing the RESCAL model with sparse constraints. gRDA is itself a generalization of the RDA algorithm for sparse neural networks [42], which can be very useful for sparse online learning with l_1 regularization as it can explicitly exploit the regularization structure. In each iteration of the RDA algorithm, the weights of the model are updated, taking into account not only the loss function but also the whole regularization term introduced to achieve the sparsity.

To apply l_1 regularization to our model, we first update the RESCAL model's loss function:

$$\mathcal{L}_{\text{sRESCAL}} = \mathcal{L}_{\text{RESCAL}} + \lambda \sum_{w \in W} \{||w\}||_1 \tag{3}$$

where λ is a hyperparameter that controls the importance of the regularization term.

We follow the same training procedure as [2]. We apply the data augmentation method used by [7] adding reciprocal relations for every triple in the dataset. We also use *1-N scoring*, i.e., we score all entity-relation pairs $\{(e_s, r\})$ and the corresponding inverse $\{(e_o, r^{-1}\})$ with every entity $e \in \mathcal{E}$. We use the Binary Cross Entropy (BCE) loss function:

$$\mathcal{L}_{\text{RESCAL}} = -\frac{1}{n_e} \sum_{i=1}^{n_e} \{(\mathbf{y}^i \log\{(\mathbf{p}^i\}) + \{(1 - \mathbf{y}^i\}) \log\{(1 - \mathbf{p}^i\})\}), \tag{4}$$

where $\mathbf{p} \in \mathbb{R}^{n_e}$ is the vector of predicted probabilities and $\mathbf{y} \in \mathbb{R}^{n_e}$ is the binary label vector.

3.4 Optimization

In gRDA, at each iteration, the learning weights are adjusted by solving a simple optimization problem that involves the running average of all past subgradients of the loss function. Its update rule is the following:

$$\mathbf{w}_{n+1} = \underset{\mathbf{w} \in \mathbb{R}^d}{\arg\min} \left\{ \mathrm{w}^T \left\{ \left(-\mathrm{w}_0 + \gamma \sum_{1=0}^{\mathrm{n}} \Delta \mathrm{f}\{(\mathrm{w}_\mathrm{i}; \mathrm{Z}_{\mathrm{i}+1}\}) \right\} \right) + g\{(n, \gamma\})\mathcal{P}(\mathrm{w}) + F(\mathrm{w}) \right\} \tag{5}$$

where γ is the step size, $\mathcal{P}(\mathbf{w})$ is the penalty function, $F(\mathbf{w})$ is a deterministic and convex regularizer which stabilizes the optimization process in the same manner as proposed in [32] and $g\{(n, \gamma\})$ is a deterministic non-negative function of n, γ. Notice that RDA is a special case of gRDA where $g\{(n, \gamma\}) = n\gamma$ and $\mathbf{w}_0 = 0$.

Since we want to apply the l_1 regularization to the RESCAL model, we follow the same criteria given by [6] for gRDA-l_1. We set the strongly convex function $F(\mathbf{w}) = \frac{1}{2}\{||\mathbf{w}\}||_2^2$ and the penalty function to be $\mathcal{P}(\mathbf{w}) = \{||w\}||_1$. We also follow the function

$$g(n, \gamma) = c\gamma^{\frac{1}{2}}(n\gamma - t_0)_+^{\mu}, \tag{6}$$

which is conjectured by authors to be a universal formula for applying gRDA-l_1 in difficult tasks. In Eq. 6, $c = \gamma$ and μ are hyperparameters and $t_0 \geq 0$ is the time mean dynamics. Furthermore, μ is the trade-off hyperparameter between accuracy and sparsity.

Following the stochastic mirror descent representation of gRDA (see [6] Section 2) and given $F(\mathbf{w}) = \frac{1}{2}\{||\mathbf{w}\}||_2^2$, we can use the closed-form proximal operator of $g(n, \lambda)\mathcal{P}(\mathbf{w})$ [28], for $j = 1, 2, \ldots, d$,

$$\Delta\Psi_{n,\gamma,j}^*(\mathbf{v}) = \mathrm{sgn}(v_j) \cdot \{(\{|v_j\}| - g\{(n, \gamma\})\})_+, \tag{7}$$

which will serve as our penalty function $\mathcal{P}$ for gRDA-l_1. Thanks to the closed-form proximal operator, the computational cost per iteration in the stochastic mirror descent gRDA is as cheap as SGD.

We present the algorithm for optimizing the sparse RESCAL model (sRESCAL) in Algorithm 1. We first initialize the weights of the entity and relation matrices (for each slice) $\mathbf{E}, \mathbf{R}_k$. Then, for each triplet n in the training set $\mathcal{D}$, we update the accumulator of gradients v by applying Eq. 6. Finally, we set the weights by following the update rule in Eq. 7.

Algorithm 1 gRDA-l_1 algorithm for sparse RESCAL

```
Input: $\mathcal{D}$
Initialize: $\mathbf{e}, \forall \mathbf{e} \in \mathbf{A}, \mathbf{w}, \forall \mathbf{w} \in \mathbf{R}$
for $n$ in $\mathcal{D}$ do
    $t \leftarrow$ update time of triple $n$
    accumulate gradients
    $v := c\gamma^{\frac{1}{2}}(n\gamma - t)^{\mu}_{+}$
    for $j = 1$ to $d$ do
        $w_{j+1} := sgn(v_j) \cdot (\{ \mathsf{v} \} - \gamma \Delta f\{(\mathbf{w}_j;\ Z_{j+1}\}))$
    end for
end for
```

4 Experiments

In this section, we analyze the performance in the link prediction task and the interpretability of the proposed algorithm. We compare our sparse model with several dense model baselines. First of all, we describe the datasets and the setup used for the experiments. Then, we study the link prediction task and compare our model with the state-of-the-art dense models. Finally, we analyze the interpretability of the model using the word intrusion task.

4.1 Datasets

The datasets used for the evaluation are the following (see Table 1):

- FB15k [4] is a subset of the Freebase database which contains facts about the world.
- FB15k-237 [36] is a better-suited version of FB15k where the inverse of many relations was removed to increase the difficulty.
- WN18 [4] is a subset of WordNet, a hierarchical database containing lexical relations between words.
- WN18RR [7] is a better-suited version of WN18 where the inverse of many relations was removed to increase the difficulty.

Both datasets were created for the link prediction task.

Table 1 Datasets for link prediction and their number of entities and relations

Dataset	$\mid \mathcal{E} \mid$	$\mid \mathcal{R} \mid$
FB15k	14,951	1345
FB15k-237	14,541	237
WN18	40,943	18
WN18RR	40,943	11

4.2 Implementation details

We open source the PyTorch implementation of the sRESCAL model on GitHub[1].

We select the hyper-parameters using random search by validation set performance. For FB15k and FB15k-237, we select the entity and relation embedding size from 128, 248, 512. Embedding size for WN18 and WN18RR is chosen from 50, 100, 200 due to the small number of relations in the dataset. Additionally, we apply batch normalization [14] and dropout [33] to ease the training procedure. We choose the dropout value in the range (0.1, 0.5) for every dataset. The learning rate is selected within the range (0.1, 0.9) also for both datasets as we observed high learning rate values necessary to overcome local minima. We set c to 0.00005 since we found it to be a suitable value for the starting sparsity in the embeddings. We choose μ in the range (0.6, 0.8) where high values correspond to higher sparsity in embeddings. We analyze the impact of μ in the following subsections. Finally, the batch size is selected from 248, 512, 1028. The best hyperparameters for each dataset are presented in Table 2.

4.3 Link prediction

For the evaluation of the link prediction task, we create all possible candidate triples by adding every entity in $\mathcal{E}$ to the test entity-pair, then we use our model to score and rank each of the candidates. We apply the filtered setting where all known true triples in the Knowledge Graph $\mathcal{D}$ are removed from the candidate set. We use the standard metrics for the task: mean reciprocal rank and hits@k, k $\in$ 1, 3, 10. Mean reciprocal rank is the average of the inverse of the mean rank assigned to the true triple overall candidate triples. Hits@k measures the percentage of times a true triple is ranked within the top k candidate triples.

We present the link prediction results in Table 3. Although sRESCAL does not outperform the state-of-the-art models, it does remain competitive while finding more interpretable and sparse solutions (see Sect. 4.4).

Regarding the simplest versions of the datasets, in WN18, sRESCAL achieves high results on every metric and gets close to the other models. For FB15k, sRESCAL suffers the same phenomena as the standard RESCAL model. The performance on the dataset is considerably lower when compared to other models. We argue that this happens due to not finding the optimal hyperparameters for the dataset. Results from [30, 41] also provide low values for RESCAL in such dataset. However, as we will see next, in the more challenging version of the dataset, FB15k-237,

[1] https://github.com/unai-zulaika/sRESCAL. Code will be available when the paper is published

Table 2 Best hyperparameters for sRESCAL for datasets where: *LR* denotes learning rate, μ is the trade-off hyperparameter between accuracy and sparsity, c is the hyperparameter controlling the starting sparsity value, d_e and d_r correspond to the entity and relation embedding size, $d\#k, k \in \{1, 2, 3\}$ are the dropout values applied on the subject entity embedding, relation matrix, and subject entity embedding after it has been transformed by the relation matrix respectively, and LS is the label smoothing

Dataset	*LR*	μ	c	d_e	d_r	$d\#1$	$d\#2$	$d\#3$	LS
FB15k	0.2	0.65	0.00005	248	128	0.36	0.4	0.4	0.17
FB15k-237	0.2	0.65	0.00005	248	128	0.36	0.4	0.4	0.17
WN18	0.9	0.4	0.00005	200	50	0.36	0.4	0.4	0.17
WN18RR	0.9	0.4	0.00005	200	50	0.36	0.4	0.4	0.17

Table 3 Results in link prediction

		RESCAL	DistMult	ComplEx	TuckER	sRESCAL
FB15k	MRR	.64	**.84**	.83	.79	.38
	Hits@10	.82	**.90**	.89	.89	.55
	Hits@3	.70	**.86**	.85	.83	.42
	Hits@1	.54	**.80**	**.80**	.74	.29
FB15k-237	MRR	**.35**	.24	.24	**.35**	**.34**
	Hits@10	**.54**	.41	.42	**.54**	.52
	Hits@3	**.39**	.26	.27	**.39**	.37
	Hits@1	**.26**	.15	.15	**.26**	**.25**
WN18	MRR	.94	.94	**.95**	**.95**	.89
	Hits@10	.95	.95	.95	.95	.94
	Hits@3	.95	.94	.95	.95	.92
	Hits@1	.94	.93	.94	**.95**	.87
WN18RR	MRR	.46	.43	.44	**.47**	.43
	Hits@10	.51	.49	.51	**.52**	.49
	Hits@3	**.48**	.44	.46	**.48**	.45
	Hits@1	.43	.39	.41	**.44**	.39

Bold numbers indicate the best performance across models

RESCAL, and sRESCAL provide significantly better results, being two of the strongest models overall.

Regarding the most challenging datasets, sRESCAL achieves fantastic results, being competitive in every metric. Our model equates DistMult and is close to ComplEx performance-wise in the WN18RR dataset and stays close to the state-of-the-art model, TuckER, by less than 4 points in the metrics. For FB15k-237, sRESCAL outperforms both DistMult and ComplEx by a significant margin in every metric. Furthermore, the performance almost equals RESCAL and TuckER providing fantastic results on every metric.

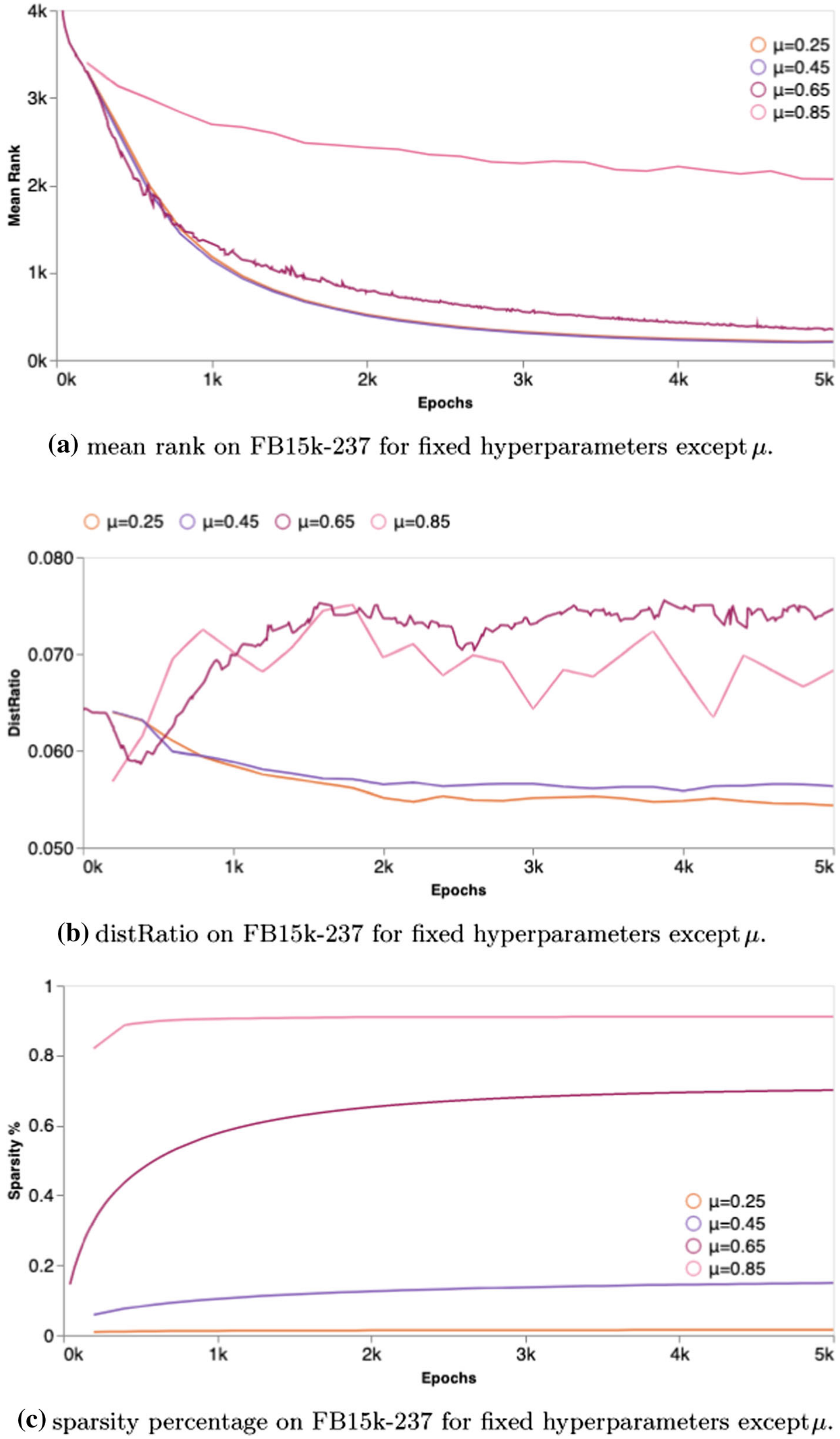
While the performance of sRESCAL remains competitive to other models, it does also achieve high sparsity and interpretability levels. Thus, we present sRESCAL as a valid and robust model for the link prediction task. Furthermore, as stated before, the induction of sparsity in the resulting embeddings will increase their interpretability by raising semanticity in their latent dimension. The increase of interpretability is further analyzed in Sect. 4.4 and the outcome is observable in Table 6. However, the introduction of sparsity to the embeddings does also have a penalization on performance. Thus, we analyze the tradeoff between sparsity, thus interpretability, and performance.

4.3.1 Sparsity and performance trade-off

The trade-off between the sparsity level and performance in the embeddings is a crucial aspect to study. sRESCAL is a flexible model that can be tuned to increase or decrease its sparsity levels and, thus, its performance.

Since sRESCAL uses the gRDA-l_1 optimization algorithm, sparsity is defined by hyperparameters c (initial sparsity level) and μ, which defines the penalization applied to the learning weights. By tuning those hyperparameters, we can achieve the sparsity level we want at the cost of lowering the final performance of the model. To demonstrate the variation on those hyperparameters, we present different metrics: (a) mean rank value, (b) DistRatio (an interpretability metric based on the word

Fig. 1 Results on FB15k-237 for fixed hyperparameters except μ



(a) mean rank on FB15k-237 for fixed hyperparameters except μ.

(b) distRatio on FB15k-237 for fixed hyperparameters except μ.

(c) sparsity percentage on FB15k-237 for fixed hyperparameters except μ.

intrusion task) value and (c) sparsity percentage for dataset FB15k-237 in Fig. 1. We provide results on different values of μ while fixing every other hyperparameter (we also fix c since we found the value 0.0005 to be the most optimal for sRESCAL in every case).

Results show a clear trade-off between μ, thus the sparsity level of the solution, and the final results on both

Table 4 Sparsity, DistRatio, and negativity results for FB15k-237

	Entity			Relation		
Model	Sparsity	DistRatio	Negativity	Sparsity	DistRatio	Negativity
Rescal	0	.056	.464	0	.052	.499
TuckER	0	.055	.509	0	**.059**	.503
sRESCAL	**.684**	**.067**	.094	**.954**	.047	.022

Bold numbers indicate the best performance across models

Table 5 Sparsity, DistRatio, and negativity results for WN18RR

	Entity			Relation		
Model	Sparsity	DistRatio	Negativity	Sparsity	DistRatio	Negativity
Rescal	0	.051	.499	0	.061	.498
TuckER	0	.050	.507	0	.058	.508
sRESCAL	**.275**	**.053**	.335	**.436**	**.066**	.280

Bold numbers indicate the best performance across models

performance and interpretability. Figure 1a presents a difference between μ values where high values, 0.85 and 0.65, achieve worst results on mean rank when compared to the lower, 0.25, 0.45 values. However, we find that $\mu = 0.65$ remains competitive to lower values as the difference in the mean rank is quite small. Furthermore, when revising the interpretability metric DistRatio (further explained in next subsection) in (b), the difference between the most interpretable models with high μ values is evidenced. As expected, solutions with a high μ value achieve good results on DistRatio while those with low μ perform worse. This fact is directly related to the sparsity level of the solution, which is shown in (c), where high μ valued solutions have higher sparsity levels. As we stated, gRDA allows sRESCAL to be a model that obtains high sparsity values and improves its interpretability[2]. Nevertheless, this comes with a trade-off on the standard link prediction performance metrics such as mean rank, lowering the model's performance on those but raising the interpretability of the outcome embeddings.

4.4 Interpretability

In this section, we analyze the interpretability of our sparse model compared to the dense state-of-the-art models. In the same manner as [10, 23], we perform experiments on the word intrusion task to evaluate the interpretability of our model. For this section, we only consider the datasets FB15k-237 and WN18RR due to their difficulty and the fact that they are the standard datasets in the current literature.

The word intrusion task evaluates coherence regarding the semantic meaning in each dimension belonging to the representations. Since sparse models create embeddings with only a few dimensions activated, i.e., the value is not 0, each dimension should correspond to a meaningful semantic concept. The task works in the following way: for each dimension i of the embeddings, it finds the k ($k \in \mathbb{N}$) most relevant Knowledge Graph entities, which are those that have the highest values in the corresponding dimension. Then, a non-relevant Knowledge Graph entity (one that has a low value in the corresponding dimension) is chosen, and it is combined with the top-k entities to create a set of $k + 1$ entities for dimension i. We refer to the non-relevant Knowledge Graph entity as the intruder, and we seek to identify it. An example set of entities for $k = 5$: {Screenwriter, Film director, Film producer, Television producer, Actor, Erie} where *Erie* is the intruder entity since it is a city and does not belong to the TV and film industry. Human annotators usually perform the word intrusion task; however, we adopt the automatic version of the task presented in [35]. In this version, the DistRatio evaluation metric is applied. The idea is that to find the intruder entity automatically, its distance to the top-k entities should be high. We measure the distance ratio between the intruder entity and the top-k entities to the distance between the top-k entities themselves. High ratio values mean better interpretability levels because the intruder entity is far (in the embedding space) from the top words (which are close due to their semantic meaning). The metric can formally be presented as:

$$\text{DistRatio} = \frac{1}{d}\sum_{i=1}^{d}\frac{\text{InterDist}_i}{\text{IntraDist}_i} \tag{8}$$

[2] gRDA can be reduced to RDA by setting $g(n,\gamma) = n\gamma$ and $\mathbf{w}_0 = 0$, which at the same time is an un-penalized version of SGD as setting $\mathcal{P}(\mathbf{w})$ and $F(\mathbf{w}) = \frac{1}{2}\{||w\}||_w^2$. In such a case, we could convert sRESCAL into RESCAL itself.

Table 6 Top 5 words of some dimensions in RESCAL and sRESCAL

Model	Top-5 words
RESCAL	Condition, device, fastener, quality, computer
	Mammal family, unpleassant person, masculinization, photograph, process
	Break, city, asterid dicot genus, right, meaning
	Decoration, tract, mining, magic, improvement
	Rescue, touch, ethnic group, stay, stand
sRESCAL	Bulldog, Golden Retriever, Labrador Retriever, Yorkshire Terrier, German Shepherd
	Finding Neverland, Showgirls, Remember the Titans, Cry Freedom, Shadowlands
	Woody Allen, Christopher Hitchens, George Carlin, Harold Pinter, Oliver Stone
	Electronic keyboard, Percussion, Electric guitar, Acoustic guitar, Guitar
	Miami Heat, Orlando Magic, San Antonio Spurs, Chicago Bulls, Boston Celtics

$$\mathrm{IntraDist}_i = \sum_{w_j \in \mathrm{top}_k(i)} \sum_{\substack{w_j \in \mathrm{top}_k(i) \\ w_k \neq w_j}} \frac{\mathrm{dist(w_j, w_k)}}{k(k-1)} \tag{9}$$

$$\mathrm{InterDist}_i = \sum_{w_j \in \mathrm{top}_k(i)} \frac{\mathrm{dist}(w_j, w_{bi})}{k} \tag{10}$$

where $\mathrm{top}_k(i)$ denotes the top-k entities corresponding to dimension i, w_{bi} denotes the intruder entity for dimension i, $\mathrm{dist}(w_j, w_k)$ is the distance between entities w_j and w_k, $\mathrm{IntraDist}_i$ is the average distance between the top-k entities on dimension i and $\mathrm{InterDist}_i$ denotes the average distance between the intruder entity and top-k words on dimension i. We set $k = 5$ and the distance function to be the Euclidean distance.

We present the interpretability results on FB15k-237 and WN18RR in Tables 4 and 5 respectively. We provide sparsity percentages, DistRatio and negativity percentages for entities and relations for sRESCAL, RESCAl and TuckER models. sRESCAL achieves the best results on DistRatio for both datasets, except in relations for FB15k-237, while maintaining high sparsity values. Both RESCAL and TuckER do not have any sparsity level since they are optimized using the Adam method. Furthermore, our experiments demonstrate the effectiveness of sparsity for developing more interpretable embeddings. Moreover, we present results on negativity since gRDA-l_1, while not constraining to positive values, enforces non-negativity on embeddings. Positivity is an important characteristic for the interpretability of the embeddings since human people do not describe a concept by what is not. Besides, positivity allows to perform additive combinations and create representations from differents parts [19, 23]. Results present sRESCAL as a highly non-negative method when compared to RESCAL and TuckER, improving interpretability in the embeddings.

We also provide a qualitative evaluation of the interpretability results. We select the top 5 words of a few dimensions in RESCAL and sRESCAL and present them in Table 6. While the top 5 words provided by the standard RESCAL model do not have any coherence or semantic meaning, the sRESCAL model gets clear topics. From the first row to the fifth: dog breed, films, actors and directors, music topics (instruments), and basketball teams.

5 Conclusion

In this work, we present a novel technique for learning sparse Knowledge Graph Representations. The approach uses the generalized Regularized Dual Averaging online optimization algorithm and applies the l_1 regularization into the RESCAL model. The experiments demonstrate that the sparse RESCAL remains competitive with the state-of-the-art while achieving high sparsity and non-negativity in the embeddings. We prove that gRDA-l_1 can effectively be applied to RESCAL and other tensor factorization models. This fact provides the opportunity to increase the interpretability in the link prediction task from the core elements used to perform it, the embeddings. Furthermore, the results in the word intrusion task show that the model's interpretability is effectively improved. The developed embeddings contain a higher semantic coherence and are more understandable for people. In the future, we consider to extend gRDA-l_1 to other tensor factorisation models.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. We gratefully acknowledge the support of the Basque Government's Department of Education for the predoctoral funding; the Ministry of Economy, Industry and Competitiveness of Spain under Grant No.: RTI2018-101045-A-C22 (FuturAAL), PEACEOFMIND project (ref. PID2019-105470RB-C31) and the INCEPTION(PID2021-128969OB-I00) project. Finally, we would like to thank NVIDIA for providing us with their hardware via the Nvidia GPU Grant and the Basque Government for their Grant on scientific equipment acquisition with No. EC-19.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. Allen C, Balazevic I, Hospedales T (2021) Interpreting knowledge graph relation representation from word embeddings. In: International conference on learning representations, https://openreview.net/forum?id=gLWj29369lW
2. Balazevic I, Allen C, Hospedales T (2019) TuckER: tensor factorization for knowledge graph completion. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, China, pp. 5185–5194, https://doi.org/10.18653/v1/D19-1522 https://aclanthology.org/D19-1522
3. Barbieri N, Bonchi F, Manco G (2014) Who to follow and why: link prediction with explanations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1266–1275
4. Bordes A, Usunier N, Garcia-Duran A, et al (2013) Translating embeddings for modeling multi-relational data. In: Burges C, Bottou L, Welling M, et al (eds) Advances in neural information processing systems, vol 26. Curran Associates, Inc., https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf
5. Chandrahas , Sengupta T, Pragadeesh C, et al (2020) Inducing interpretability in knowledge graph embeddings. In: Proceedings of the 17th international conference on natural language processing (ICON). NLP Association of India (NLPAI), Indian Institute of Technology Patna, Patna, India, pp. 70–75, https://aclanthology.org/2020.icon-main.9
6. Chao SK, Wang Z, Xing Y, et al (2020) Directional pruning of deep neural networks. In: Larochelle H, Ranzato M, Hadsell R, et al (eds) Advances in neural information processing systems, vol 33. Curran Associates, Inc., pp. 13986–13998, https://proceedings.neurips.cc/paper/2020/file/a09e75c5c86a7bf6582d2b4d75aad615-Paper.pdf
7. Dettmers T, Minervini P, Stenetorp P, et al (2018) Convolutional 2d knowledge graph embeddings. In: Proceedings of the thirty-second AAAI conference on artificial intelligence and thirtieth innovative applications of artificial intelligence conference and eighth AAAI symposium on educational advances in artificial intelligence. AAAI Press, AAAI'18/IAAI'18/EAAI'18
8. Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res 12:2121–2159
9. Engelen van JE, Boekhout HD, Takes FW (2016) Explainable and efficient link prediction in real-world network data. In: International symposium on intelligent data analysis, Springer, pp. 295–307
10. Faruqui M, Tsvetkov Y, Yogatama D, et al (2015) Sparse overcomplete word vector representations. In: Proceedings of the 53rd annual meeting of the association for computational Linguistics and the 7th international joint conference on natural language processing (Volume 1: Long Papers). Association for computational Linguistics, Beijing, China, pp. 1491–1500, https://doi.org/10.3115/v1/P15-1144, https://www.aclweb.org/anthology/P15-1144
11. Fatemi B, Ravanbakhsh S, Poole D (2019) Improved knowledge graph embedding using background taxonomic information. In: Proceedings of the AAAI conference on artificial intelligence, pp. 3526–3533
12. Gusmao AC, Correia AHC, De Bona G, et al (2018) Interpreting embedding models of knowledge bases: a pedagogical approach. arXiv preprint arXiv:1806.09504
13. Harshman RA, Green PE, Wind Y et al (1982) A model for the analysis of asymmetric data in marketing research. Mark Sci 1(2):205–242
14. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd international conference on international conference on machine learning - Volume 37. JMLR.org, ICML'15, pp. 448–456
15. Kadlec R, Bajgar O, Kleindienst J (2017) Knowledge base completion: baselines strike back. In: Proceedings of the 2nd workshop on representation learning for NLP. Association for computational Linguistics, Vancouver, Canada, pp. 69–74, https://doi.org/10.18653/v1/W17-2609, https://aclanthology.org/W17-2609
16. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980
17. Lacroix T, Usunier N, Obozinski G (2018) Canonical tensor decomposition for knowledge base completion. In: International conference on machine learning, PMLR, pp. 2863–2872
18. Lacroix T, Obozinski G, Usunier N (2020) Tensor decompositions for temporal knowledge base completion. In: International conference on learning representations, https://openreview.net/forum?id=rke2P1BFwS
19. Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401(6755):788–791
20. Liang Z, Yang J, Liu H et al (2022) Seatte: An embedding model based on separating attribute space for knowledge graph completion. Electronics 11(7):1058
21. Makarov I, Kiselev D, Nikitinsky N et al (2021) Survey on graph embeddings and their applications to machine learning problems on graphs. PeerJ Comput Sci 7:e357
22. Minervini P, Costabello L, Muñoz E, et al (2017) Regularizing knowledge graph embeddings via equivalence and inversion axioms. In: Joint European conference on machine learning and knowledge discovery in databases, Springer, pp. 668–683
23. Murphy B, Talukdar P, Mitchell T (2012) Learning effective and interpretable semantic models using non-negative sparse embedding. Proc COLING 2012:1933–1950
24. Murphy G (2004) The big book of concepts. MIT Press, Cambridge
25. Nickel M, Tresp V, Kriegel HP (2011) A three-way model for collective learning on multi-relational data. In: Icml, pp. 809–816
26. Padia A, Kalpakis K, Ferraro F et al (2019) Knowledge graph fact prediction via knowledge-enriched tensor factorization. Web Semant. https://doi.org/10.1016/j.websem.2019.01.004
27. Panigrahi A, Simhadri HV, Bhattacharyya C (2019) Word2Sense: Sparse interpretable word embeddings. In: Proceedings of the

57th annual meeting of the association for computational Linguistics. Association for computational Linguistics, Florence, Italy, pp 5692–5705, https://doi.org/10.18653/v1/P19-1570, https://www.aclweb.org/anthology/P19-1570

28. Parikh N, Boyd S et al (2014) Proximal algorithms. Found Trends® Optim 1(3):127–239
29. Rossi A, Barbosa D, Firmani D et al (2021) Knowledge graph embedding for link prediction: a comparative analysis. ACM Trans Knowl Discov Data (TKDD) 15(2):1–49
30. Ruffinelli D, Broscheit S, Gemulla R (2020) You can teach an old dog new tricks! on training knowledge graph embeddings. In: International conference on learning representations, https://openreview.net/forum?id=BkxSmlBFvr
31. Schunn CD (1999) The presence and absence of category knowledge in lsa. In: 21st annual conference of the cognitive science society, Citeseer
32. Shalev-Shwartz S et al (2012) Online learning and online convex optimization. Found Trends® Mach Learn 4(2):107–194
33. Srivastava N, Hinton G, Krizhevsky A et al (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
34. Subramanian A, Pruthi D, Jhamtani H, et al (2018) Spine: Sparse interpretable neural embeddings. In: Proceedings of the AAAI conference on artificial intelligence, vol. 32(1). https://doi.org/10.1609/aaai.v32i1.11935, https://ojs.aaai.org/index.php/AAAI/article/view/11935
35. Sun F, Guo J, Lan Y, et al (2016) Sparse word embeddings using l1 regularized online learning. In: Proceedings of the twenty-fifth international joint conference on artificial intelligence, AAAI Press, pp. 2915–2921
36. Toutanova K, Chen D, Pantel P, et al (2015) Representing text for joint embedding of text and knowledge bases. In: Proceedings of the 2015 conference on empirical methods in natural language processing. Association for computational Linguistics, Lisbon, Portugal, pp. 1499–1509, https://doi.org/10.18653/v1/D15-1174, https://www.aclweb.org/anthology/D15-1174
37. Trouillon T, Dance CR, Gaussier É et al (2017) Knowledge graph completion via complex tensor factorization. J Mach Learn Res 18(1):4735–4772
38. Tucker LR (1964) The extension of factor analysis to three-dimensional matrices. In: Gulliksen H, Frederiksen N (eds) Contributions to mathematical psychology. Holt Rinehart and Winston, New York, pp 110–127
39. Wan G, Du B (2021) Gaussianpath:a bayesian multi-hop reasoning framework for knowledge graph reasoning. In: Proceedings of the AAAI conference on artificial intelligence 35(5):4393–4401. https://ojs.aaai.org/index.php/AAAI/article/view/16565
40. Wan G, Du B, Pan S et al (2020) Adaptive knowledge subgraph ensemble for robust and trustworthy knowledge graph completion. World Wide Web 23(1):471–490
41. Wang Y, Ruffinelli D, Gemulla R, et al (2019) On evaluating embedding models for knowledge base completion. In: Proceedings of the 4th workshop on representation learning for NLP (RepL4NLP-2019). Association for computational Linguistics, Florence, Italy, pp. 104–112, https://doi.org/10.18653/v1/W19-4313, https://aclanthology.org/W19-4313
42. Xiao L (2010) Dual averaging methods for regularized stochastic learning and online optimization. J Mach Learn Res 11(Oct):2543–2596
43. Xie Q, Ma X, Dai Z, et al (2017) An interpretable knowledge transfer model for knowledge base completion. In: Proceedings of the 55th annual meeting of the association for computational Linguistics (Volume 1: long papers). Association for computational Linguistics, Vancouver, Canada, pp. 950–962, https://doi.org/10.18653/v1/P17-1088, https://aclanthology.org/P17-1088
44. Yang B, Yih W, He X, et al (2015) Embedding entities and relations for learning and inference in knowledge bases. In: Bengio Y, LeCun Y (eds) 3rd international conference on learning representations, ICLR 2015, San Diego, CA, US. http://arxiv.org/abs/1412.6575
45. Zhang Z, Cai J, Wang J (2020) Duality-induced regularizer for tensor factorization based knowledge graph completion. In: Proceedings of the 34th international conference on neural information processing systems. Curran Associates Inc., Red Hook, NY, USA
46. Zhong Q, Ding L, Liu J, et al (2022) Knowledge graph augmented network towards multiview representation learning for aspect-based sentiment analysis. arXiv preprint arXiv:2201.04831