A Comprehensive Survey of Knowledge Graph Embeddings with Literals: Techniques and Applications

Genet Asefa Gesese[1,2], Russa Biswas[1,2], and Harald Sack[1,2]

[1] FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany
[2] Karlsruhe Institute of Technology, Institute AIFB, Germany
firstname.lastname@kit.edu

Abstract. Knowledge Graphs are organized to describe entities from any discipline and the interrelations between them. Apart from facilitating the inter-connectivity of datasets in the LOD cloud, KGs have been used in a variety of applications such as Web search or entity linking, and recently are part of popular search systems and Q&A applications etc. However, the KG applications suffer from high computational and storage cost. Hence, there arises the necessity of having a representation learning of the high dimensional KGs into low dimensional spaces preserving structural as well as relational information. In this study, we conduct a comprehensive survey based on techniques of KG embedding models which consider the structured information of the graph as well as the unstructured information in form of literals such as text, numerical values etc. Furthermore, we address the challenges in their embedding models followed by a discussion on different application scenarios.

Keywords: Knowledge Graph · Embedding · Literals · Knowledge Graph embedding survey.

1 Introduction

Various Knowledge Graphs (KGs) have been published for the purpose of sharing linked data. Some of the most popular general purpose KGs are DBpedia [14], Freebase [1], Wikidata [23], and YAGO[16]. KGs have become quite invaluable for various applications mainly in the area of artificial intelligence. For instance, in a more general sense, KGs can be used to support decision making process and to improve different machine learning applications. Spam detection, movie recommendation, and market basket analysis are some of the ML applications which can benefit from KGs [25]. General purpose KGs as e.g., Wikidata, often comprise millions of entities, represented as nodes, with hundreds of millions of facts, represented as edges connecting those nodes. However, a significant number of important graph algorithms needed for the efficient manipulation and analysis of graphs have proven to be NP-complete [11]. Although KGs are effective in representing structured data, the underlying symbolic nature of the way data is encoded as triples (i.e. $< subject, predicate, object >$) usually makes KGs hard to manipulate [24]. In order to address these issues and use a KG efficiently, it is recommended to convert it into a low dimensional vector space while preserving the graph properties. To this end, various attempts have been made so far to learn vector representation (embeddings) for KGs. However, most of these approaches, including the state of the art TransE [2], are structure based embeddings which do not include any literal information. This is a major disadvantage because a lot of information encoded in the literals will be left unused when capturing the semantics of a certain entity.

Literals can bring advantages to the process of learning KG embeddings in two major ways. The first is in learning embeddings for novel entities i.e., entities which are not linked to any other entity in the KG but have some literal values associated with them. In most existing structure based embedding models, it is not possible to learn embeddings for such novel entities. However, this can be addressed by utilizing the information held in literals to learn embeddings. The other advantage of literals is improving the representation of entities in structure based embedding models where an entity is required to appear in at least minimum number of relational triples. Some approaches have been proposed to make use of literals for KG embeddings. The focus of this paper is to discuss these different embedding approaches and their advantages and drawbacks in the light of different application scenarios. Our contributions include:

- A detailed analysis of the existing literal enriched KG embedding models and their approaches. In addition, a method is proposed to categorize them into different groups.
- The research gaps in the area of KG embeddings in using literals are indicated as directions for further future works.

The rest of this paper is organized as follows. Sect. 2 presents a brief overview of related work. In Sect. 3, the problem formulation is provided. In Sect. 4, the analysis of the different KG embedding techniques with literals is discussed. In Sect. 5, various tasks used to evaluate the embedding models discussed in Sect. 4 are explained. The survey is concluded in Sect. 6 by providing directions for future work for KG embedding with literals.

2 Related Work

Few attempts have been made to conduct surveys on the techniques and applications of KG embeddings [12, 3, 24]. However, none of these surveys include all the existing KG embedding models which make use of literals. The first survey [12] is conducted with focus on network embedding models. The second [3] and the third [24] surveys discuss only RESCAL [17] and KREAR [15] as methods which use attributes of entities for KG embeddings, and focuses mostly on the structure based embedding methods.

However, RESCAL is a matrix-factorization method for relational learning which encodes each object/data property as a slice of the tensor ending up increasing the dimensionality of the tensor automatically. Thus, this method is not efficient to utilize literals in KG embedding. Similarly, KREAR is not a proper KG embedding model with literals since it takes only those data properties which have categorical values and ignores those which take any random literals as values. This shows that there is a gap in the KG embedding surveys. Taking this into consideration, in this paper, a survey on KG embedding models, which make use of literals is provided.

3 Problem Formulation

In this section, a brief introduction is provided on fundamental KG and its embeddings followed by a formal definition of KG embedding with literals.

3.1 Definitions and preliminaries

Relations (or Properties). Based on the nature of the objects, relations are classified into two main categories:

- **Object Relations** – relations that link entities to entities
- **Data Type Relations** – relations that link entities to data values (literals).The triples consisting of literals as objects are often referred to as attributive triples.

3.2 Types of literals

Literals in a KG encode information that is not captured by the relations or links between the entities. There are different types of literals present in the KGs:

- **Text** – Wide variety of different information can be stored in KG in the form of free text such as names, labels, titles, descriptions, comments, etc. In most of the KG embedding models with literals, text information has been further categorized into ***Short text*** and ***Long text*** for better capture of the semantics in the model. The literals which are fairly short such as for relations like names, titles, etc. are considered as *Short text*. On the other hand, for strings that are much longer such as descriptions of entities, comments, etc. are considered as *Long text*.
- **Numeric** – Information encoded in the form of real numbers, decimal numbers such as height, year or date, population, etc. also provide useful insight. It is worth considering the numbers as distinct entities in the embeddings models, as it has its own semantics to be covered which cannot be covered by string distance metrics. For e. g. 777 is more similar to 788 than 77.
- **Units of Measurement** – (Numeric) literals often denote units of measurements to a definite magnitude. For e. g. Wikidata property wdt:P2048 takes values in mm, cm, m, km, inch, foot, and pixel. Hence, discarding the units and considering only the numeric values without normalization results in loss of semantics, especially in the case if units are not comparable, as e.g. units of length and units of weight.

- **Images** – Images also provide latent useful information for modelling of the entities. For example, a person's details such as age, gender etc. can be deduced via visual analysis of an image depicting the person.
- **Others** – Useful information encoded in the form of other literals such as external URIs which could lead to an image, text, audio or video files.

Since the information present in the KGs is diverse, modelling of the entities is a challenging task.

- **RQ1** – *How to combine the structured (triples with object relations) and unstructured information (attributive triples) in the KGs into the representation learning?*
- **RQ2** – *How to capture and combine the heterogeneity of the types of literals present in the KGs into representation learning?*

4 Knowledge Graph Embeddings with Literals

In this study, the investigated KG embedding models with literals are divided into the following different categories based on the literals utilized: (i) Text, (ii) Numeric, (iii) Image , and (iv) Multi-modal. A KG embedding model which utilizes at least two types of literals is considered as multi-modal. This section consists of an analysis of the models in each category, with their similarities and differences, followed by a discussion of potential drawbacks.

4.1 Text Literals

Subsequently, four KG models considering text literals are discussed, namely, Extended RESCAL [18], DKRL [28], KDCoE [4], and KGloVe with literals [5].

Extended RESCAL improves the original RESCAL approach by processing literal values more efficiently and deal with the sparsity nature of the tensors. In this method, attributive triples are handled in a separate matrix factorization, which is performed jointly with the tensor factorization of the non-attributive triples. Attributive triples containing only text literals are encoded in an entity-attributes matrix in such a way that given a triple, one or more $< \textit{data type relation}, \textit{value} >$ pairs are created by tokenizing and stemming the object literal. Despite the advantage that this approach handles multi-valued literals, it does not consider the sequence of words in the literal values.

DKRL generates embeddings of entities and relations of a KG by combining structure-based and description-based representations. The structure based representation of entities and relations are obtained via TransE [2], in which the relation in each triple *(head, relation, tail)*, is regarded as the translation from head entity to tail entity. On the other hand, continuous bag of words (CBOW) and a deep convolutional neural network model (CNN) have been used to generate the description based representations of the head and tail entities. In case of CBOW, short text is generated from the description based on keywords and their corresponding word embeddings are summed up to generate the entity embedding. In the CNN model, after preprocessing of the description, pre-trained word vectors from Wikipedia are provided as input. The CNN has five layers and after every convolutional layer pooling is applied to decrease the parameter space of CNN and filter noises. Max-pooling is applied for the first pooling and mean pooling for the last one. CNN model works better than CBOW because it preserves the sequence of words.

KDCoE focuses on the creation of an alignment between entities of multilingual KGs by creating new inter-lingual links (ILLs). The model leverages a weakly aligned multilingual KG for semi-supervised cross-lingual learning using entity descriptions. It performs co-training of a multilingual KG embedding model (KGEM) and a multilingual literal description embedding model (DEM) iteratively in order for each model to propose a new ILL alternately. KGEM adopts TransE whereas DEM uses an attentive gated recurrent unit encoder (AGRU) to encode the multilingual entity descriptions.

KGloVe with literals works by first creating a cooccurrence matrix from the underlying graph by performing Personalized PageRank (PPR) on the (weighted) graph followed by the same optimisation used in the GloVe [19] approach. Two cooccurrence matrices are generated independently and merged in the end. The first matrix is generated using KGloVe [6] technique and Named Entity Recognition is performed prior to the creation of the second matrix.

The basic differences between these models lie in the methods used to exploit the information given in the text literals and combine them with structure-based representation. One major advantage of KDCoE over text literal based embedding models is that it considers descriptions present in multilingual KGs. Also, both DKRL and KDCoE embedding models are designed to perform well for the novel entities which have only attributive triples in the KGs. Other types of text literals are not widely considered.

4.2 Numeric literals

In this section, four models which make use of numeric literals, namely, MT-KGNN [21], KBLRN [10], LiteralE [13], and TransEA [26] are discussed.

MT-KGNN trains a relational network (RelNet) for triple classification and an attribute network (AttrNet) for attribute value regression in order to learn embeddings for entities, object properties, and data properties. Only data properties with non-discrete literal values are considered in this approach. RelNet is a simple binary (pointwise) classification whereas the AttrNet is a regression task. In RelNet, a concatenated triple is passed through a nonlinear transform and then a sigmoid function is applied to get a linear transform. In the case of AttrNet, two regression tasks are performed for head and tail data properties respectively. Finally, the two networks are trained in a multi-task fashion using a shared embedding space.

KBLRN combines the relational, latent (learned by adapting TransE), and numerical features together. It uses a probabilistic PoE (Product of Experts) method to combine these feature types and train them jointly end to end. Each relational feature is formulated by adopting the rule mining approach AMIE +[9], to be evaluated in the KG to compute the value of the features. Numerical features are used with the assumption that, for some relation types, the differences between the head and tail is seen as characteristics for the relation itself. In PoE, one expert is trained for each (relation type, feature type) pair. The parameters of the entity embedding model are shared by all the experts in order to create dependencies between them. For numerical features, a radial basis function is applied as activation function if the difference of values is in a specific range.

LiteralE is designed in order to incorporate literals into existing latent feature models, which are designed for link prediction. Given a base model, for instance *Distmult*, LiteralE modifies the scoring function f used in *Distmult* by replacing the vector representations of the entities e_i in f with literal enriched representations e_i^{lit}. In order to generate e_i^{lit}, LiteralE uses a learnable transformation function g which takes e_i and its corresponding literal vectors l_i as inputs and maps them to a new vector. For g, linear transformations, non-linear transformations, simple multi-layer NNs, and non-linear transformations with gating mechanisms are proposed. The modified scoring function f is trained following the same procedure as in the base model.

TransEA has two component models; a newly proposed attribute embedding model and a directly adopted translation-based structure embedding model, TransE. For the attribute embedding, it uses all attributive triples containing numeric values as input and applies a linear regression model to learn embeddings of entities and attributes. The loss function for TransE is defined by taking the sum of the respective loss functions of the component models with a hyperparameter to assign a weight for each of the models. Finally, the two models are jointly optimized in the training process by sharing the embeddings of entities.

Despite their support for numerical literals, all the embedding methods discussed fail to interpret the semantics behind data types of literals and units. For e. g., '1999€ ' and 'the year 1999' could be considered same because type semantics are discarded. Moreover, none of the models apply normalization for literal values, hence the semantic similarity between two literal values such as, 200 mm and 2 cm is not captured. Also, most of the models do not have proper mechanism to handle multi-valued literals.

4.3 Image

IKRL [27] learns embeddings by jointly training a structure-based (by adapting TransE) with an image-based representation. For the image-based representation, an image encoder is applied to generate embedding for each instance of a multi-valued image relation. Attention-based multi-instance learning is used to integrate the representations learned for each image instance by automatically calculating the attention that should be given to each instance. Given a triple, the overall energy function is defined by combining four energy functions which are based on two kinds of entity representations. The first energy function is same as TransE and the second uses their corresponding image-based representations for both head and tail entities. The third function is based on the structure-based representation of the head entity and the image-based representation of the tail entity whereas the fourth function is the exact opposite.

4.4 Multi-modal

Numeric literals and text: LiteralE with blocking [8] proposes to improve the effectiveness of the data linking task by combining LiteralE with a CER blocking[7] strategy. Unlike LiteralE, it also considers literals from URI infixes of the head entities and data relations of attributive triples. The CER blocking is based on a two-pass indexing scheme. In the first pass, Levenshtein distance metric is used to process literal objects and URI infixes whereas in the second pass semantic similarity computation with Wordnet is applied to process object/data relations. All the extracted literals are tokenized into word lists so as to create the indices.

EAKGAE [22] jointly learns entity embeddings of two KGs using structure embedding (by adapting TransE) and attribute character embedding. Given a triple (h, r, a), the data property r is interpreted as a translation from the head entity h to the literal value a i.e. $h + r = f_a(a)$ where $f_a(a)$ is a compositional function. Three different compositional functions SUM, LSTM, and N-gram-based functions have been proposed. SUM is defined as a summation of all character embeddings of the attribute value. In LSTM, the final hidden state is taken as a vector representation of the attribute value. The N-gram-based function, which shows better performance than the others, uses the summation of n-gram combination of the attribute value.

The common drawback with both methods is that text and numeric literals are treated in the same way. They also do not consider literal data type semantics or multi-valued literals in their approach. Furthermore, since EAKGAE is using character-based attribute embedding, it fails to capture the semantics behind the cooccurrence of syllables.

Numeric literals, Text, and Images: MKBE [20] is a a multi-modal knowledge graph embedding, in which the text, numeric and image literals are modelled together. It extends DistMult, which creates embedding for entities and relations, by adding neural encoders for different data types. For image triples, a fixed-length vector is encoded using CNN. On the other hand, textual attributes are encoded using sequential embedding approaches like LSTMs. Given the vectors representations of the entities, relations and attributes, the same scoring function from DistMult is used to determine the correctness probability of triples.

5 Applications

In this section, the different KG application scenarios used by the techniques discussed in Sect. 4 are presented.

Link prediction. Link prediction aims to predict new links for a KG given the existing links among the KG entities. The models Extended RESCAL, LiteralE, TransEA, KBLRN, DKRL, KDCoE, EAKGAE, IKRL, and MKBE have

	FB15K	**FB15K-237**	**YAGO-10**
KBLN	0.739	0.301	0.487
MTKGNN(DistMult+MultiTask)	0.669	0.285	0.481
DistMult+LiteralE	0.583	**0.314**	0.504
DistMult+LiteralE-Gate	0.723	0.300	-
ComplEx+LiteralE	**0.765**	0.299	**0.509**
ConvE+LiteralE	0.66	**0.314**	0.506

(a)

Model	**MRR**
Numeric	
KBLN	0.503
S+N	**0.549**
Image	
IKRL	0.509
S+I	**0.566**

(b)

Table 1: (a) MRR results on link prediction taken from LiteralE [13], and (b) MRR results on link prediction task on YAGO-10 taken from MKBE [20].

been evaluated on the link prediction task. However, it is not possible to compare the obtained evaluation results because the experiments have been carried out on different datasets. The authors of the LiteralE and MKBE models conducted some experiments to compare their proposed models/submodels with already existing ones. LiteralE has been compared with KBLN, which is a submodel of KBLRN designed without taking into consideration the relational information of graph feature methods. Besides KBLN, LiteralE has been compared with a new modified version of MTKGNN, where its ER-MLP part is replaced with DistMult to make it compatible with their specific implementation environment. The results taken from LiteralE are shown in Table 1a. From the result, it can be seen that DistMult+LiteralE delivers better MRR values when it is compared with both KBLN and MTKGNN on the datasets FB15k-237 and YAGO-10. The authors of LiteralE argue that the performance of DistMult+LiteralE is lower than the others on the FB15K dataset because this dataset includes a lot of inverse relations and hence claim that it is not an appropriate dataset for link prediction. The other experiments conducted are in MKBE where the submodels, structures along with numeric and image literals respectively arecompared with KBLN and IKRL respectively as shown in Table 1b. Thereby, it can be inferred that the two submodels of MKBE perform better than their counterparts.

Triple Classification. A potential triple is classified as 0 (false) or 1 (true). MTKGNN, KGlove with literals, and IKRL have been evaluated on this task. Since they do not use a common evaluation dataset, it is not possible to compare the reported results directly.

Entity Classification. Given a KG and an entity, the entity type is predicted using a multilabel classification algorithm with KG entity types as given classes. DKRL has been evaluated on this task.

Entity Alignment. Semantically similar entities are determined from multiple KGs using specific similarity metrics. EAKGAE has been evaluated on an entity alignment task. In addition, KDCoE has also been evaluated on a cross-lingual entity alignment task which determines similar entities in different languages. Despite the fact that both these models use the same task for evaluation, their experimental results cannot be compared since they are based on different datasets.

Other Machine Learning problems. Attribute value prediction, nearest neighbor analysis, data linking, and document classification are other applications scenarios used for the evaluation of the models discussed in Sect. 4. In MTKGNN, attribute value prediction is applied using an attribute-specific Linear Regression classifier for evaluation. Nearest neighbor analysis has been performed in LiteralE to compare DistMult+LiteralE with the base model DistMult. Data linking and document classification tasks have been used in LiteralE with blocking and KGlove with literals respectively.

6 Conclusion and Future Directions

To sum up, in this paper, a comprehensive survey of KG embedding models with literals is presented. The survey provides a detailed analysis and categorization of the embedding techniques of these models along with their application scenarios and limitations. As mentioned in Section 4, these embedding models have different drawbacks. None of them consider the effect that data types and units have on the semantics of literals. Most of them also do not have a proper mechanism to handle multi-valued literals. Thus, filling these gaps will be taken as a direction for future work.

Moreover, only few approaches have been proposed for multi-modal KG embeddings and none of them take into consideration literals with URIs linking to items such as audio, video, or pdf files. This clearly indicates that more work has to be invested to address different types of literals. Regarding the comparison of the quality of the models, as discussed in Section 5, it was only possible to use the experimental results conducted for some of the models as most use different datasets and application scenarios. However, as a future work, experiments for all of the models on different applications will be performed to enable better comparability.

References

1. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In: ACM SIGMOD international conference on Management of data (2008)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating Embeddings for Modeling Multi-Relational Data. In: NIPS (2013)
3. Cai, H., Zheng, V.W., Chang, K.C.C.: A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. TKDE (2018)
4. Chen, M., Tian, Y., Chang, K.W., Skiena, S., Zaniolo, C.: Co-training Embeddings of Knowledge Graphs and Entity Descriptions for Cross-Lingual Entity Alignment. arXiv preprint arXiv:1806.06478 (2018)
5. Cochez, M., Garofalo, M., Lenßen, J., Pellegrino, M.A.: A First Experiment on Including Text Literals in KGloVe. arXiv preprint arXiv:1807.11761 (2018)
6. Cochez, M., Ristoski, P., Ponzetto, S.P., Paulheim, H.: Global rdf Vector Space Embeddings. In: International Semantic Web Conference. Springer (2017)

7. de Assis Costa, G., de Oliveira, J.M.P.: A Blocking Scheme for Entity Resolution in the Semantic Web. In: AINA (2016)
8. de Assis Costa, G., de Oliveira, J.M.P.: Towards Exploring Literals to Enrich Data Linking in Knowledge Graphs. In: AIKE (2018)
9. Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast Rule Mining in Ontological Knowledge Bases with AMIE+. VLDB (2015)
10. García-Durán, A., Niepert, M.: Kblrn: End-to-End Learning of Knowledge Base Representations with Latent, Relational, and Numerical Features. In: UAI (2018)
11. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA (1990)
12. Goyal, P., Ferrara, E.: Graph Embedding Techniques, Applications, and Performance: A Survey. Knowl.-Based Syst. (2018)
13. Kristiadi, A., Khan, M.A., Lukovnikov, D., Lehmann, J., Fischer, A.: Incorporating Literals into Knowledge Graph Embeddings. CoRR (2018)
14. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al.: Dbpedia–A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. Semantic Web (2015)
15. Lin, Y., Liu, Z., Sun, M.: Knowledge Representation Learning with Entities, Attributes and Relations. ethnicity (2016)
16. Mahdisoltani, F., Biega, J., Suchanek, F.M.: Yago3: A Knowledge Base from Multilingual Wikipedias. In: CIDR (2013)
17. Nickel, M., Tresp, V., Kriegel, H.P.: A Three-Way Model for Collective Learning on Multi-Relational Data. In: ICML (2011)
18. Nickel, M., Tresp, V., Kriegel, H.P.: Factorizing Yago: Scalable Machine Learning for Linked Data. In: Proceedings of the 21st international conference on World Wide Web. ACM (2012)
19. Pennington, J., Socher, R., Manning, C.: Glove: Gobal Vectors for Word Representation. In: EMNLP (2014)
20. Pezeshkpour, P., Chen, L., Singh, S.: Embedding multimodal relational data for knowledge base completion. arXiv preprint arXiv:1809.01341 (2018)
21. Tay, Y., Luu, A.T., Phan, M.C., Hui, S.C.: Multi-task Neural Network for Nondiscrete Attribute Prediction in Knowledge Graphs. CoRR (2017)
22. Trsedya, B.D., Qi, J., Zhang, R.: Entity Alignment between Knowledge Graphs Using Attribute Embeddings. In: AAAI (2019)
23. Vrandečić, D., Krötzsch, M.: Wikidata: A Free Collaborative Knowledge Base (2014)
24. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge Graph Embedding: A Survey of Approaches and Applications. TKDE (2017)
25. Wilcke, X., Bloem, P., De Boer, V.: The Knowledge Graph as the Default Data Model for Learning on Heterogeneous Knowledge. Data Science (2017)
26. Wu, Y., Wang, Z.: Knowledge Graph Embedding with Numeric Attributes of Entities. In: Rep4NLP@ACL (2018)
27. Xie, R., Liu, Z., Chua, T.S., Luan, H.B., Sun, M.: Image-embodied knowledge representation learning. In: IJCAI (2017)
28. Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation Learning of Knowledge Graphs with Entity Descriptions. In: AAAI (2016)