Knowledge graph embeddings

Paolo Rosso, Dingqi Yang, Philippe Cudré-Mauroux

Synonyms

- Knowledge base embeddings
- RDF graph embeddings

Definitions

Knowledge graph embeddings: a vector representation of entities and relations in a knowledge graph that preserves the inherent structure of the knowledge graph as well as the reasoning ability over the graph.

Introduction

With the growing popularity of multi-relational data on the Web, knowledge graphs (KGs) have become a key data source in various application domains, such as Web search, question answering, and natural language understanding. In a typical KG such as Freebase Bollacker et al (2008) or Google's Knowledge Graph Google (2014), entities are connected via relations. For example, *Bern* is capital of *Switzerland*. Formally, a popular approach to represent such relational data is to use the Resource Description Framework. It defines a fact as a triple (subject, predicate and object), which is also known as head, relation, and tail or (h,r,t) for short. Following the above example, the head, relation and tail are *Bern*, *capitalOf* and *Switzerland*, respectively. With a considerable number of entities and relations (e.g., Google's Knowledge Graph has more than 18 billion of triples with 570 million of entities and 35,000 of relations by the end of 2014), KGs now become a valuable information source that can empower many semantic Web applications.

Despite the importance of building large-scale KGs, their symbolic and logical frameworks are not flexible enough to be compatible with modern statistical and machine-learning techniques which require often numerical inputs. In this context, knowledge graph embeddings that project entities and relations in a KG into a low-dimensional continuous vector space have attracted much attention. One of the key benefits of such numeric representation is that they can easily serve as input to classical statistical and machine learning approaches. The learnt entity and relation embeddings can thus be used in different tasks, such as KG completion Bordes et al (2013), relation extraction Weston et al (2013), entity classification and entity resolution Nickel et al (2011).

In the following, we first discuss typical KG embedding models and then their extensions by integrating additional data sources. We then summarize the applications of KG embeddings.

Learning knowledge graph embeddings

Learning KG embeddings consists in two key steps in general:

1. Defining a KG embedding model with a specific scoring function, which computes the probability that a given triple is true;
2. Initializing entity and relation embeddings (e.g. vectors) according to the KG embedding model, and learning those embeddings by maximizing the sum of the scoring function over all triples in the KG. Triples appearing in the KG will have higher scores than the triples that do not exist in the KG.

KG embedding models

Depending on the type of scoring function, there are two categories of embedding models: translational distance models and semantic matching models. Translational distance models, such as *TransE* Bordes et al (2013), use a scoring function that measures the distance between two entities, while semantic matching models, such as RESCAL, use a scoring function that measures the similarity of the facts.

Translational distance model: TransE and its extensions

TransE is a representative translational distance model that projects entities and relations onto a unique vector space. In this model, the head h and the tail t of a triplet are connected by their relation r, holding the fact that the embedding of t should be similar to the embedding of h plus the embedding of r (i.e., $h + r \approx t$). The proposed idea is based on the vector-offset method for identifying linguistic regularities in continuous space word representations Mikolov et al (2013), for example, *USA - dollar* $\approx$ *Japan - yen*. In a KG, this analogy holds since through the *currencyOf* relation we get *dollar + currencyOf* $\approx$ *USA* and *yen + currencyOf* $\approx$ *Japan*. In this way, the scoring function is defined as the negative distance between the sum of the head and the relation, subtracted by the tail:

$$f_r(h,t) = -||\mathbf{h} + \mathbf{r} - \mathbf{t}||_{1/2} \qquad (1)$$

Following this initial idea, several techniques improved TransE by designing sophisticated scoring functions that are able to capture complex KG structures, in particular multi-mapping relations (one-to-many, many-to-one, or many-to-many). TransH Wang et al (2014b), for instance, suggests a new approach by projecting the relations on different hyperplanes in order to capture many-to-many mapping properties of some relations; TransR Lin et al (2015b) defines a mapping matrix and a vector for every relation; TransD Ji et al (2015) introduces dynamic matrices for each entity-relation pair by considering the diversity of entities and relations simultaneously. Sophisticated scoring functions can indeed improve the KG embeddings in some downstream learning tasks, tough they also increase the complexity of the embedding models.

Semantic matching model: RESCAL and its extensions

RESCAL is a tensor factorization model for KG embeddings, which decomposes a three-way tensor consisting of head, relation and tail dimensions. RESCAL generates for each entity a vector, and for each relation a matrix capturing the interaction between the entities. The proposed model allows for discovering the correlation between multiple interconnected entities. The model represents facts via a tensor product with a corresponding scoring function defined as follows:

$$f_r(h,t) = \mathbf{h}^\top \mathbf{M}_r \mathbf{t} = \sum_{i=1}^{d} \sum_{j=1}^{d} [\mathbf{M}_r]_{ij} \cdot [\mathbf{h}]_i \cdot [\mathbf{t}]_j \tag{2}$$

where h and t are the vectors of the head and tail, respectively, and M_r is the matrix that represents the relations. d refers to the dimension of the embeddings.

Several pieces of work extend RESCAL by designing customized tensor factorization models. DistMult Yang et al (2014) simplifies the RESCAL model by using a bilinear formulation, showing similar performance with less parameters (more efficient in the learning process). However, the model works with symmetric relations only. Trouillon et al (2016) propose a model called Complex Embeddings (ComplEx) that extends DistMult in order to model asymmetric relations. ComplEx introduces a complex space in which head, relation and tail embeddings are represented. In this model, the scoring function generates different scores from facts with asymmetric relations. Neural Tensor Network (NTN) Socher et al (2013) is another model with a neural network architecture. For each fact, the embedding vectors of its head and tail are fed into the input layer of a neural network, and then mapped onto its hidden layer combined with a relation-specific tensor. Finally, the output layer generates a score for each fact.

Learning process

The goal of the learning process is to maximize the sum of the scoring function over all triples in the KG. Typical examples of optimization algorithms used in this context include Stochastic Gradient Descent (SGD) Robbins and Monro (1951), BroydenFletcherGoldfarbShanno (BFGS) Battiti and Masulli (1990) and AdaGrad Duchi et al (2011). In order to accelerate the training process, negative sampling techniques can be applied by replacing the head, the relation or the tail of a given fact. These generated triples are called negative samples. There are two main methods of generating negative samples, based on the *Open World Assumption (OWA)* and on the *Closed World Assumption (CWA)*.

Open World Assumption

The open world assumption assumes that the KG only contains true facts and the facts that do not appear can either be false or just missing Drumond et al (2012). Under this assumption, a negative fact is probabilistically generated given a positive fact by randomly corrupting its head, relation or tail. The entity and relation representations are learned by minimizing a loss function defined based on the scoring function, such as the logistic loss defined as follows:

$$\min_{h,r,t} \sum_{(h,r,t)\in\mathbb{D}^{+}\cup\mathbb{D}^{-}} log(1+exp(-y_{hrt}\cdot f_r(h,t))) \tag{3}$$

where D^{+} and D^{-} are positive and negative training samples, and y_{hrt} is equal to 1 if the label is positive, -1 otherwise. The logistic loss can be optimized using for instance stochastic gradient descent in mini-batch mode. In the training phase, a set of true facts are sampled and a set of negative fact get generated, and the embeddings can be iteratively updated with a fixed or adaptive learning rate.

Closed World Assumption

The closed world assumption assumes that all the facts that are not in the KG are negative samples (i.e., assuming the KG is complete). The entity and relation representations are learned by minimizing, for example, the squared loss:

$$\min_{h,r,t} \sum_{h,t\in\mathbb{E},r\in\mathbb{R}} (y_{hrt}-f_r(h,t))^2 \tag{4}$$

where $\mathbb{E}$ and $\mathbb{R}$ are the set of entities and relations, respectively. y_{hrt} is equal to 1 if the triplet appears in the KG, 0 otherwise. In addition, Nickel and Tresp (2013) proposes the logistic loss and Miettinen (2011) the absolute loss as alternatives to the squared loss.

In summary, the closed world assumption usually has more limitations than the open world assumption, as it penalizes the missing true facts from a KG. In practice, despite their tremendous size, modern KGs all suffer from incompleteness issues West et al (2014). Consequently, the open world assumption is more realistic for most settings and thus performs better on average better than the closed world assumption Guo et al (2017).

KG embedding extension by integrating additional information

When learning KG embeddings, additional information, such as entity types, relation paths, textual descriptions or logical rules, can be added to the embedding model to improve the quality of the embeddings on certain tasks.

Entity types represent semantic categories the entity belongs to. For example, the entity *PresidentObama* can be annotated as a *PERSON* entity type. This piece of information can be incorporated in different ways. Nickel et al (2012) use the entity type as a relation and its corresponding facts *(h,r,t)* as training example. Guo et al (2015) propose a method in which entities of the same type are close to each other in the vector space. The entity type can also be used to set constraints for different relations. For example, Xie et al (2016) use this constraint to generate correct negative samples by filtering out triples with incorrect entity types.

Relation paths refer to a sequence of relations between two entities. The multi-hop relationships contain useful information that can be used for KG completion. For example, Lao and Cohen (2010) predict the relation between entities using a path ranking algorithm that connects two entities. More precisely, the relations can be represented as vectors or matrices, and their addition or multiplication can be used to compose a path as vector or matrix composition. Lin et al (2015a) show a method to approximate the relation path via sampling and pruning. Along similar lines, Toutanova et al (2016) propose an algorithm that incorporates paths with specific lengths and intermediate entities in the model.

Most of the KGs contain entity descriptions that can be used to enrich the semantic information of the entities. External information sources, such as Wikipedia articles Wang et al (2014a) or news releases Socher et al (2013), can be used to extend the entity description by providing richer textual information. A representative work by Wang et al (2014a) introduces a model that combines text corpus and KG to align them in the same vector space and creates KG embeddings and text embeddings. The model includes three main parts: a KG model, a text model and an alignment model. Specifically, the KG model is used to generate embeddings of entities and relations in the KG while the text model is used to generate embeddings from the text corpus. Finally, the alignment model is used to align the KG embedding and text embedding in the same vector space using different alignment mechanisms, such as entity name and Wikipedia anchors. In this way, the model is able to predict out-of-KG entities (phrases not stored in the KG but that appear in the text).

Logical rules are another type of information that could be integrated into embedding models. For instance, if two entities are connected by the relation *HasWife*, then they should also be connected by the relation *HasSpouse*. There exist systems, such as *WARMR* Dehaspe and Toivonen (1999), *ALEPH* Muggleton (1995), and *AMIE* Galárraga et al (2013), that can automatically extract such kinds of relations. Richardson and Domingos (2006) prove that the logical rules contain rich information and that they can be used to acquire and infer further knowledge. Following this idea, Wang et al (2015) propose an approach seamlessly incorporating logic rules into KG embedding models by reducing the solution space and thus improving the inference accuracy for knowledge base completion tasks. Guo et al (2016) propose a model that embeds KG and logical rules in a unified framework. Specifically, logical rules are first instantiated into ground rules, for example, *HasWife(x,y)* implies *HasSpouse(x,y)* and vector embeddings are introduced for entity pairs. For each fact of the logical rule, a score is computed in order to indicate whether the ground rule is satisfied or not. The embedding model is then learned based on the unified facts and rules. In this way, the model is more effective for knowledge acquisition and inference as the embeddings are compatible with both facts and rules.

In addition to the information described above, further types of information that can be added to the embedding models include entity attributes, temporal information and graph structures. Nickel et al (2012) highlight the fact that entity attributes and relations must be separated. The authors propose a new algorithm to handle attributes efficiently. Jiang et al (2016) show that the KG facts are often time-sensitive and that they develop a time-aware knowledge base embedding approach by taking advantage of the time at which facts have occurred. The proposed solution forces the embeddings to be temporally consistent by using temporal constraints to model the relations. Feng et al (2016) show a graph-aware approach that learns entitiy and relation embeddings by leveraging the relation paths and edge context (i.e., all the relations that connect an entity). The intuition behind this approach is that all the relations linking to and from an entity are representative of that entity.

KG embedding applications

Typical applications of KG embedding include link prediction, triple classification, entity classification and entity resolution.

- Link prediction (also called KG completion) attempts to discover missing facts based on the contents of the KG. Specifically, it predicts an entity given a relation and a second entity, i.e., given *(r, t)* it predicts *h*, denoted also as *(?, r, t)*, or given *(h, r)* it predicts *t*, denoted also as *(h, r, ?)*. Lin et al (2015a) define this task as entity prediction while Bordes et al (2014b) define it as entity ranking. A similar approach predicts a relation given its head and tail entities, denoted also as *(h, ?, t)*, which is similar also to relation prediction Xie et al (2016). In order to evaluate the results generated by this task, a common practice is to store in a list all the answers and see the rank of the correct answer. Several evaluation metrics can be used in this context, such as *Hits@n* that considers only the ranks smaller than *n*, or the *mean rank*, that is, the average of the predicted ranks.
- Triple classification aims at determining whether a triple appears in a KG. More precisely, triple classification can be performed based on the score of a candidate triple *(h, r, t)* that can be easily computed using the scoring function. In this way, an unseen fact can be either true if its score is higher than a threshold and false otherwise. Traditional evaluation metrics can be used in this task, for example, mean average precision Guo et al (2016) or

micro and macro averaged accuracy Guo et al (2015).

- Entity classification classifies entities into different semantic categories. Concretely, the type of an entity is usually denoted using a *IsA* relation, and entity classification can thus be seen as a particular case of the link prediction task, in which only *(h, IsA, ?)* triples are predicted.
- Entity resolution verifies whether two entities are actually referring to the same object or not. Bordes et al (2014b) tackle this problem by considering a scenario in which the relation of two equivalent entities is explicit denoted as *EqualTo*. By learning the embedding for this type of relation, the problem of entity resolution becomes a triple classification problem. Fundamentally, the triple classification problem judges whether the fact *(h, EqualTo, t)* is true or not. Alternatively, Nickel et al (2011), propose a different approach that computes the similarity between two entities and use the score to calculate the likelihood that two entities refer to the same object. This method works even if the relation *EqualTo* is not encoded in the KG.

KG embeddings can also be applied to other application domains beyond KGs. Three most popular out-of-KG applications are relation extraction, question answering and recommendation systems.

- Relation extraction tries to discover relations from text where entities have already been identified. Weston et al (2013) propose a method to extract relations by combining TransE and text, showing that the integration of TransE and a traditional text-based extractor can actually improve the performance of relation extraction.
- Question answering refers to the task of answering questions over KGs. Given a question in plain text, a fact or a set of facts containing the correct answer is extracted as an answer. This task is challenging because of the extended variability of natural language text used to formulate the question and of the extensive size of the KGs. A successful solution that involves KG embedding is proposed by Bordes et al (2014a), which learns embeddings in order to put questions and corresponding answers closer in the vector space. Given a question and an answer, the model generates a high score if the answer is correct, low score otherwise. The results show that, by involving the KG, the task is successfully performed without using any rules or additional tagging step as most traditional question answering applications do.
- Recommendation systems suggest users a list of items according to the users' preferences. Collaborative filtering techniques are often used to perform recommendations based on the historical interaction between users and items. However, user-item interactions are often sparse, leading to unsatisfactory performance. To alleviate this issue, hybrid recommendation systems were developed by adding auxiliary information Yu et al (2014). Zhang et al (2016) propose a hybrid recommendation system that integrates a KG. More precisely, the hybrid recommendation system models structural knowledge by applying a KG embedding technique such as *TransR* in order to learn the representation

of each item. Similarly, the users are represented by vectors, and each item is represented by its KG vector representation plus an offset. Finally, the preference of a user for a specific item is computed as a product of the user and item vectors. In this way, the hybrid recommendation system automatically extracts semantic representations from facts in the KG to improve the quality of the recommendation system.

Conclusion

With the booming of multi-relational data on the Web, knowledge graphs have become an important data source empowering many applications. However, the symbolic and logical representation of KGs make it difficult to take them as input to machine-learning or processing pipelines. To tackle this issue, knowledge graph embedding techniques were proposed to project entities and relations from a KG into a low-dimensional continuous vector space, while still preserving the inherent structure of the KG and reasoning capabilities over the KG. The learnt embeddings have been successfully used in both KG-reasoning applications, such as link prediction, triple classification, entity classification and entity resolution, and out-of-KG applications, such as relation extraction, question answering and recommendation systems.

Acknowledgement

This work was supported by the Swiss National Science Foundation under grant number 407540_167320 (Tighten-it-All).

References

Battiti R, Masulli F (1990) Bfgs optimization for faster and automated supervised learning. In: International neural network conference, Springer, pp 757–760

Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J (2008) Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, ACM, pp 1247–1250

Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O (2013) Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems, pp 2787–2795

Bordes A, Chopra S, Weston J (2014a) Question answering with subgraph embeddings. arXiv preprint arXiv:14063676

Bordes A, Glorot X, Weston J, Bengio Y (2014b) A semantic matching energy function for learning with multi-relational data. Machine Learning 94(2):233–259

Dehaspe L, Toivonen H (1999) Discovery of frequent datalog patterns. Data Mining and knowledge discovery 3(1):7–36

Drumond L, Rendle S, Schmidt-Thieme L (2012) Predicting rdf triples in incomplete knowledge bases with tensor factorization. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing, ACM, pp 326–331

Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research 12(Jul):2121–2159

Feng J, Huang M, Yang Y, et al (2016) Gake: Graph aware knowledge embedding. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp 641–651

Galárraga LA, Teflioudi C, Hose K, Suchanek F (2013) Amie: association rule mining under incomplete evidence in ontological knowledge bases. In: Proceedings of the 22nd international conference on World Wide Web, ACM, pp 413–422

Google (2014) https://www.google.com/intl/bn/insidesearch/features/search/knowledge.html

Guo S, Wang Q, Wang B, Wang L, Guo L (2015) Semantically smooth knowledge graph embedding. In: ACL (1), pp 84–94

Guo S, Wang Q, Wang L, Wang B, Guo L (2016) Jointly embedding knowledge graphs and logical rules. In: EMNLP, pp 192–202

Guo S, Wang Q, Wang B, Wang L, Guo L (2017) Sse: Semantically smooth embedding for knowledge graphs. IEEE Transactions on Knowledge and Data Engineering 29(4):884–897

Ji G, He S, Xu L, Liu K, Zhao J (2015) Knowledge graph embedding via dynamic mapping matrix. In: ACL (1), pp 687–696

Jiang T, Liu T, Ge T, Sha L, Li S, Chang B, Sui Z (2016) Encoding temporal information for time-aware link prediction. In: EMNLP, pp 2350–2354

Lao N, Cohen WW (2010) Relational retrieval using a combination of path-constrained random walks. Machine learning 81(1):53–67

Lin Y, Liu Z, Luan H, Sun M, Rao S, Liu S (2015a) Modeling relation paths for representation learning of knowledge bases. arXiv preprint arXiv:150600379

Lin Y, Liu Z, Sun M, Liu Y, Zhu X (2015b) Learning entity and relation embeddings for knowledge graph completion. In: AAAI, pp 2181–2187

Miettinen P (2011) Boolean tensor factorizations. In: Data Mining (ICDM), 2011 IEEE 11th International Conference on, IEEE, pp 447–456

Mikolov T, Yih Wt, Zweig G (2013) Linguistic regularities in continuous space word representations. In: hlt-Naacl, vol 13, pp 746–751

Muggleton S (1995) Inverse entailment and progol. New generation computing 13(3):245–286

Nickel M, Tresp V (2013) Logistic tensor factorization for multi-relational data. arXiv preprint arXiv:13062084

Nickel M, Tresp V, Kriegel HP (2011) A three-way model for collective learning on multi-relational data. In: ICML, vol 11, pp 809–816

Nickel M, Tresp V, Kriegel HP (2012) Factorizing yago: scalable machine learning for linked data. In: Proceedings of the 21st international conference on World Wide Web, ACM, pp 271–280

Richardson M, Domingos P (2006) Markov logic networks. Machine learning 62(1):107–136

Robbins H, Monro S (1951) A stochastic approximation method. The annals of mathematical statistics pp 400–407

Socher R, Chen D, Manning CD, Ng A (2013) Reasoning with neural tensor networks for knowledge base completion. In: Advances in neural information processing systems, pp 926–934

Toutanova K, Lin V, Yih Wt, Poon H, Quirk C (2016) Compositional learning of embeddings for relation paths in knowledge base and text. In: ACL (1)

Trouillon T, Welbl J, Riedel S, Gaussier É, Bouchard G (2016) Complex embeddings for simple link prediction. In: International Conference on Machine Learning, pp 2071–2080

Wang Q, Wang B, Guo L (2015) Knowledge base completion using embeddings and rules. In: IJCAI, pp 1859–1866

Wang Z, Zhang J, Feng J, Chen Z (2014a) Knowledge graph and text jointly embedding. In: EMNLP, vol 14, pp 1591–1601

Wang Z, Zhang J, Feng J, Chen Z (2014b) Knowledge graph embedding by translating on hyperplanes. In: AAAI, pp 1112–1119

West R, Gabrilovich E, Murphy K, Sun S, Gupta R, Lin D (2014) Knowledge base completion via search-based question answering. In: Proceedings of the 23rd international conference on World wide web, ACM, pp 515–526

Weston J, Bordes A, Yakhnenko O, Usunier N (2013) Connecting language and knowledge bases with embedding models for relation extraction. arXiv preprint arXiv:13077973

Xie R, Liu Z, Sun M (2016) Representation learning of knowledge graphs with hierarchical types. In: IJCAI, pp 2965–2971

Yang B, Yih Wt, He X, Gao J, Deng L (2014) Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:14126575

Yu X, Ren X, Sun Y, Gu Q, Sturt B, Khandelwal U, Norick B, Han J (2014) Personalized entity recommendation: A heterogeneous information network approach. In: Proceedings of the 7th ACM international conference on Web search and data mining, ACM, pp 283–292

Zhang F, Yuan NJ, Lian D, Xie X, Ma WY (2016) Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, ACM, pp 353–362