Efficient Estimation of Ontology Entities Distributed Representations

Achref Benarab[1(✉)], Jianguo Sun[1], Allaoua Refoufi[2], and Jian Guan[1]

[1] College of Computer Science and Technology, Harbin Engineering University, Harbin, China
{achref.benarab,sunjianguo,j.guan}@hrbeu.edu.cn

[2] Computer Science Department, University of Sétif-1, Sétif, Algeria
allaoua.refoufi@univ-setif.dz

Abstract. Ontologies have been used as a form of knowledge representation in different fields such as artificial intelligence, semantic web and natural language processing. The success caused by deep learning in recent years as a major upheaval in the field of artificial intelligence depends greatly on the data representation, since these representations can encode different types of hidden syntactic and semantic relationships in data, making their use very common in data science tasks. Ontologies do not escape this trend, applying deep learning techniques in the ontology-engineering field has heightened the need to learn and generate representations of the ontological data, which will allow ontologies to be exploited by such models and algorithms and thus automatizing different ontology-engineering tasks. This paper presents a novel approach for learning low dimensional continuous feature representations for ontology entities based on the semantic embedded in ontologies, using a multi-input feed-forward neural network trained using noise contrastive estimation technique. Semantically similar ontology entities will have relatively close corresponding representations in the projection space. Thus, the relationships between the ontology entities representations mirrors exactly the semantic relations between the corresponding entities in the source ontology.

Keywords: Ontology entities distributed representations · Continuous vector representations · Feature representation · Concept embeddings · Neural networks

1 Introduction

Ontologies are a powerful paradigm for knowledge representation and exchange, they provide a specific vocabulary to a knowledge domain, and according to a variable degree of formalization, they set the sense of the concepts and relationships uniting them. Gruber defines the term ontology as: "An ontology is an explicit specification of a conceptualization" [1]. Ontologies not only allow us to

© Springer Nature Switzerland AG 2019
L. Uden et al. (Eds.): KMO 2019, CCIS 1027, pp. 51–62, 2019.
https://doi.org/10.1007/978-3-030-21451-7_5

represent concepts that completely describe a knowledge domain, but also represent the semantics associated with them. The concepts are linked to each other by taxonomic and semantic relationships forming a semantic network. Ontologies have been used in several fields, such as engineering and knowledge management systems, natural language processing, semantic web, intelligent integration of information etc. The aim is to facilitate knowledge sharing and reuse [2].

In front of the huge success of deep learning techniques, until recently these techniques did not get enough interest in the ontology-engineering field because of the nature of the ontological data. However, representing ontological entities in a low dimensional continuous vector space provides generic representations for most machine learning and deep learning tasks, allowing the ontology-engineering field to benefit from these techniques. Representing ontology entities as continuous vector representations allows also applying several operations to manipulate representations: measures of similarity or distance, addition, subtraction etc. Our contribution in this paper is the proposition of a model capable of generating low dimensional continuous vector representations for ontology entities (concepts, individuals and semantic relationships) using multi-input feed-forward neural networks trained using noise contrastive estimation technique, where the training samples are generated on the basis of the different taxonomic, semantic relationships and restrictions.

The rest of the paper is organized as follows. In Sect. 2 we give an overview of related work. In Sect. 3 we present the technical details and the different methods used to generate the continuous vector representations for ontology entities in our approach. We evaluate and discuss the obtained results using several real world ontologies in Sect. 4. In Sect. 5, we give the conclusion, and highlight some directions and perspectives for future work.

2 Related Work

Conventional natural language processing tasks often use the one-hot or the bag of words representations. However, these simple words representations face several limitations, where they are very expensive i.e. the vectors are of high dimension, another limitation is that they cannot capture relations between words, even if there is a strong semantic or syntactic correlation between some of them. Continuous vector representations have been proposed for the first time for language modeling in [3], the model consists to train a feed-forward neural network to estimate the probability of the next word, based on the continuous representation of the previous words. These representations are called word embeddings, neural embeddings or prediction-based embeddings, they have been introduced through the construction of neural language models [4,5]. Word embeddings are a projection of a vocabulary words into a low dimensional space in order to preserve semantic and syntactic similarities. Thus, if the word vectors are close to one another in terms of distance in the projection space, the words must be semantically or syntactically close. Each dimension represents a latent characteristic of the word, which can capture syntactic and semantic properties.

The currently most popular word embeddings in the literature are provided by the Word2vec toolkit [6,7]. The authors proposed two architectures CBOW (Continuous Bag Of Words) and Skip-gram model for learning word embeddings that are less expensive in terms of computing time than previous models. The authors in [8] have shown that word embeddings created by a recurrent neural network capture the similarities between words and word pairs. Another recent approach is called GloVe [9], which combines two approaches: count-based matrix factorization and predictive or neural models. This approach relies on the construction of a global co-occurrence matrix of words, treating the whole corpus using a sliding window. GloVe is a model of unsupervised learning that takes into account all the information carried by the corpus and not only the information carried by a sliding window of words.

Several algorithms have been proposed to solve the problem of dimensionality reduction for graph representations such as [10–13] that were based on Principal Component Analysis and Multi-Dimensional Scaling. Authors in [14] proposed a semi-supervised algorithm to learn continuous feature representations for nodes in networks based on random walks algorithm and motivated by the previous work [6] on natural language processing. Authors in [15,16] proposed a method for representing RDF (Resource Description Framework) nodes in linked open data using language modeling approaches for unsupervised feature extraction from sequences of words based on deep walk and deep graph kernels approaches.

Embeddings evaluation techniques can be classified into two major families, extrinsic and intrinsic evaluations [17]. Extrinsic evaluation aims to evaluate the continuous vector representations on real applications of the use of embeddings for specific task. The intrinsic evaluation aims to evaluate semantic and syntactic relations between words or concepts [18], it is an inexpensive method and it gives a good estimation of a model that works or not. It uses the cosine similarity, Euclidean distance, human judgment, etc. Continuous vector representations models showed a good behavior and achieved good results in language modeling tasks. They are rapid, efficient and easy to train, meanwhile, they need few manipulations to have a model that works well and can be easily integrated into the input of deep learning systems for example.

3 The Proposed Approach

The purpose of our approach is to encode ontology concepts, individuals and semantic relations in a low dimensional space, so that the similarity in the embedding space can be used to approximate the semantic similarity in the ontology. It aims to learn low dimensional embedded vectors for ontology entities using multi-input feed-forward neural network to report semantics contained in ontologies. This method allows us to represent each ontology entity by a corresponding real number vector in $\mathbb{R}^n$. Another objective of our approach lies in putting the information and data contained in ontologies at the disposal of machine learning and deep learning algorithms. The ontology can be seen as a set of triples (subject, predicate and object), it is built from conceptual models that are semantically richer due to the explicit definition of associations and relationships between entities in the conceptual schema. Therefore, in our approach, we take benefit from these characteristics and exploit the semantic relations in the ontologies to generate the embeddings.

3.1 The Ontological Approach

Our contribution aims to use several approaches to generate the ontology entities distributed representations based on the different taxonomic and semantic relationships in the ontology.

Taxonomic Relationships. Taxonomic relations are the main mode of structuring an ontology. We assume two concepts are similar if they have the same super class. Let us consider the concepts $C_1, C_2, C_3 \sqsubseteq \top$ in an ontology O, then:

$$C_1 \sqsubseteq C_2 \land C_3 \sqsubseteq C_2 \Rightarrow C_1 \simeq C_3 \tag{1}$$

Non-taxonomic Relationships and Restrictions. Based on the non-taxonomic semantic relationships (object properties), we assume two concepts are semantically close if they have similar structural roles and they share semantic relations or restrictions with the same concepts:

$$\alpha C_1.r(C_2) \land \beta C_3.r(C_2) \Rightarrow C_1 \simeq C_3 \tag{2}$$

where r is a semantic relation and $\alpha, \beta \in \{\forall, \exists, \leqslant_n, \geqslant_n\}$.

Instances. For the ontology individuals (instances), we have applied three approaches to identify the similar individuals:

- We assume two individuals x and y are similar if they are instantiated from the same class:
$$C_1(x) \land C_1(y) \Rightarrow x \simeq y \tag{3}$$
- The second approach is based on the relations between the individual's concepts:
$$C_1(x) \land C_2(y) \land \alpha C_1.r(C_3) \land \beta C_2.r(C_3) \Rightarrow x \simeq y \tag{4}$$
- The third approach is based on the relations between the instances themselves. x, y and z are ontology individuals and r is an ontology role, then:
$$\alpha x.r(y) \land \beta z.r(y) \Rightarrow x \simeq z \tag{5}$$

3.2 The Neural Network Model

We have proposed a feed-forward multi-input neural network model, it aims to predict an object o, given a subject s and a relation r in an ontology O i.e. $G(s,r) \longrightarrow o$. The training set is a sequence of triples $(subject, predecate, object)$ defined as $\langle (s_1, r_1, o_1), \cdots, (s_k, r_l, o_k) \rangle$ where $k = |C|$, $l \in \{1, 2, \cdots, |R|\}$ and C is the set of the concepts and instances in the ontology O. The objective is to learn a model $g(s,r,o) = P(o|r,s)$ where g can be decomposed into three parts:

– A mapping M to a vector for ontology entity s in O.
– A mapping M' to a vector for each semantic relation r.
– The probability function P over the ontology entities in O.

The neural network model consists of two separated input layers, two separated projection layers and an output layer (see Fig. 1).

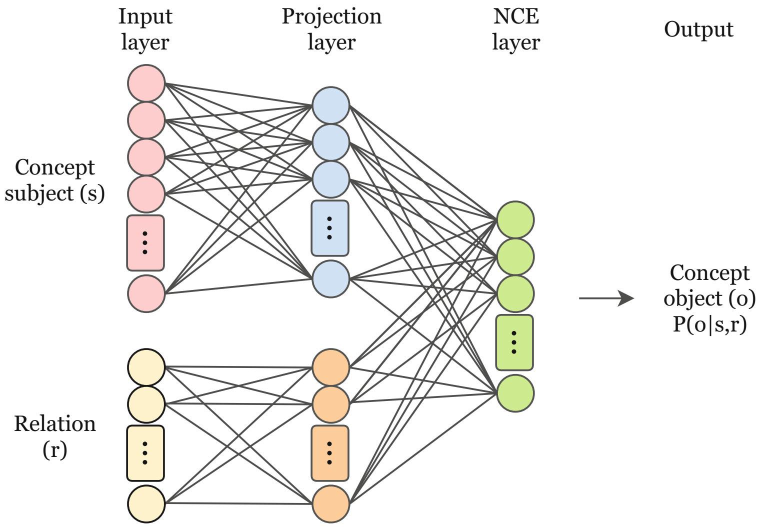

Fig. 1. The proposed neural network model architecture.

The neural probabilistic model specifies the distribution for the target concept o given a subject s and a relation r using the scoring function f:

$$P_\theta(o|s,r) = \frac{exp(f_\theta(v_o, v_s, v_r))}{\sum_{i=1}^{|C|} exp(f_\theta(v_{c_i}, v_s, v_r))} \tag{6}$$

where θ represents the model parameters, c_i is an entity (concept or instance) from the ontology O and v_o, v_s, v_r, v_{c_i} represent the vectors of o, s, r, c_i respectively. The scoring function f requires normalizing over the entire ontology entities, which is impractical and computationally expensive [19,20] when dealing with large ontologies. The noise contrastive estimation (NCE) [21] method has been used to train the model to reduce the density estimation to a probabilistic binary classification. The condition likelihood becomes:

$$P(D=0|s,r,o) = \frac{k \times q(v_o)}{exp(f_\theta(v_o, v_s, v_r)) + k \times q(v_o)} \tag{7}$$

$$P(D=1|s,r,o) = \frac{exp(f_\theta(v_o, v_s, v_r))}{exp(f_\theta(v_o, v_s, v_r)) + k \times q(v_o)} \tag{8}$$

where k represents noise samples from q. $D = 0$ means that it is a noise sample, whereas $D = 1$ defines true distribution sample. We can summarize the learning process as follows:

- Each ontology entity c is associated with a vector $v \in \mathbb{R}^n$ initialized randomly.
- For each couple s, r in a triple (s, r, o) from O as an input, the element o is taken into account as an output.
- $P(D = 1|s, r, o)$ represents the probability that an ontology entity s is the subject of a relation (predicate) r, which o is its object.
- $P(D = 0|s, r, o)$ the probability that s is not the subject of a relation r, which o is its object.
- The optimization objective function is defined as follows:

$$\begin{aligned} L(V) = arg\ max \sum_{s,o \in C, r \in R} \log P(D=1|s,r,o) + \\ \sum_{s',o \in C, r \in R} \log P(D=0|s',r,o) \end{aligned} \tag{9}$$

  It defines the sum of the logarithms of the probabilities $P(D = 1|s, r, o)$ for all the ontology elements o as objects, s as subjects, r as a relation in the set of triples, and the sum of the logarithms of the probabilities $P(D = 0|s', r, o)$ for all the elements o and r in the set of triples and a random sample of elements s' out of their triples. The symbol V indicates the set of all the vectors v of the elements which represents our model, and of which we try here to look for the optimal values in order to maximize the objective function $L(V)$. The gradient descent approach is used to find the optimal values of all vectors v corresponding to the ontology elements.

This approach allows us to find the vectors that bring together the semantically close ontology entities and can keep the semantically distant entities away.

4 Evaluation

In order to evaluate the performance of the proposed model, we have used several ontologies from different domains obtained from The Open Biological and Biomedical Ontologies (OBO[1]) Foundry [22] and BioPortal[2] repository [23].

[1] http://obofoundry.org.
[2] https://bioportal.bioontology.org.

4.1 Evaluation Metrics

To evaluate the quality of the ontology entities representations generated for each ontology, three different approaches have been used: the projection onto a two or three-dimensional map, cosine similarity and the Euclidean distance. Then we compare these representations with their corresponding ontology entities based on the semantic relations, restrictions and axioms using the Jaccard similarity. This task aims to measure the degree of similarity of the close concepts and instances in the ontology. The Jaccard index between two sets A and B is defined as follows:

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

For two vectors x and y, the cosine similarity and the Euclidean distance are defined as follows:

$$Cosine\_similarity(x,y) = \frac{x \cdot y}{\parallel x \parallel \cdot \parallel y \parallel}$$

$$Euclidean\_distance(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

The similarity values obtained from the generated representations and those obtained from the corresponding concepts and instances in the ontologies are evaluated using the standard metrics: Precision, Recall and F-measure. Given a set of similar concepts B, the precision (P) of the generated similar vector representations A is the ratio of the correct matches found and the total number of matches:

$$P(A,B) = \frac{|A \cap B|}{|A|}$$

The recall (R) computes the ratio of the correct matches found and the total number of expected connections in the ontology:

$$R(A,B) = \frac{|A \cap B|}{|B|}$$

The metric F-measure (F_1) is a harmonic measure, it combines both measures of Precision and Recall:

$$F_1(A,B) = 2 \times \frac{P(A,B) \times R(A,B)}{P(A,B) + R(A,B)}$$

4.2 Results and Discussion

For the visualization of the generated vectors in a two or three-dimensional map, we have employed the t-Distributed Stochastic Neighbor Embedding (t-SNE) technique [24], which is a non-linear dimension reduction technique particularly suitable for projecting high dimensional data onto a two or three-dimensional space. These representations can be viewed as a scatter plot, Fig. 2 illustrates a 2D representation of some concept embeddings obtained using our approach on The Human Ancestry Ontology [25]. The countries that belong to the same geographic region have close vector representations: Northern Africa countries, South-Central Asia, Western Europe, Central America, Northern Europe and the concepts representing the geographical regions as well.

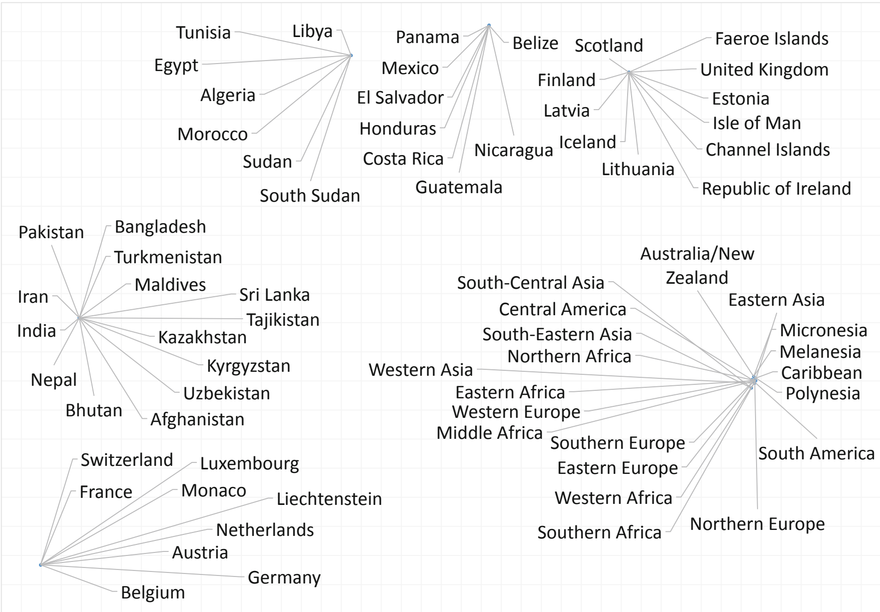

Fig. 2. The projection of some concept vectors obtained from The Human Ancestry Ontology.

We can see that the concepts that are semantically close tend to have close vector representations in the embedding space. The neural network learned also to layout the corresponding concept vector representations hierarchically in the embedding space based on the ontology taxonomic and semantic relationships between concepts from general to the most specific relationships. Initially, the entirety of the concept-vector-representations are seen as some initial concept islets. By drilling down, we can visualize the next hierarchical level that displays the respective concepts which increases the amount of information displayed locally for that particular islet.

We have applied two qualitative methods for the similarity measures, the cosine similarity and the Euclidean distance to identify the closest concepts (similar) of a given concept. The concepts considered similar based on the generated vector representations (using the cosine similarity and the Euclidean distance) are then compared to the set of similar concepts obtained from the ontology using the Jaccard index based on the taxonomic and semantic relationships. The three standard criteria: precision, recall and f-measure values are calculated for each ontology based on the cosine similarity (see Table 1) and the Euclidean distance (see Table 2).

Table 1. Values of precision, recall and f-measure for each ontology obtained using the cosine similarity.

Ontology	Precision	Recall	F-measure
HANCESTRO	0.99	0.99	0.99
SPD	0.91	0.99	0.95
UO	1.00	1.00	1.00
MF	0.99	1.00	0.99
BNO	1.00	1.00	1.00
NPI	1.00	1.00	1.00
BP	0.96	0.97	0.97
AO	1.00	1.00	1.00
BCTT	1.00	1.00	1.00
ADMIN	1.00	1.00	1.00
BFO	1.00	1.00	1.00
FAO	1.00	1.00	1.00
FHHO	0.99	1.00	0.99
SYMP	0.99	1.00	0.99
Average	0.99	0.99	0.99

From what preceded, and based on the values of the precision and recall, the generated vector representations using our approach mimic to a large degree the semantic properties of the corresponding ontology entities, where the semantically close concepts have close vector representations in the projection space (an average of 99% for cosine similarity and Euclidean distance).

Another behavior have been observed concerning the generated representations, where we have found that the semantic relationships don not have the same influence on the generated vector representations (the way that the generated vector representations are grouped in the projection space). We have found that the relationships that are frequently used in the ontology have more influence on the generated vector representations than those that are less used. That gives our method more expressiveness and an ability to better represent the concepts. For the concepts: $C_1, C_2, C_3, C_4, C_5 \sqsubseteq \top$, and the ontology roles r_1 and r_2, the relation r_1 is widely used in the ontology O than r_2, then:

$$\alpha C_1.r_1(C_2) \wedge \beta C_3.r_1(C_2) \wedge \gamma C_1.r_2(C_4) \wedge \delta C_5.r_2(C_4) \Rightarrow C_1 \simeq C_3 \tag{10}$$

where $\alpha, \beta, \gamma, \delta \in \{\forall, \exists, \leqslant_n, \geqslant_n\}$. Thus, the concept C_1 is more similar to C_3 compared to C_5.

Table 2. Values of precision, recall and f-measure for each ontology obtained using the Euclidean distance.

Ontology	Precision	Recall	F-measure
HANCESTRO	0.99	0.99	0.99
SPD	0.91	0.98	0.94
UO	1.00	1.00	1.00
MF	0.99	1.00	0.99
BNO	1.00	1.00	1.00
NPI	1.00	1.00	1.00
BP	0.95	1.00	0.97
AO	1.00	1.00	1.00
BCTT	1.00	1.00	1.00
ADMIN	1.00	1.00	1.00
BFO	1.00	1.00	1.00
FAO	1.00	1.00	1.00
FHHO	1.00	1.00	1.00
SYMP	0.99	1.00	0.99
Average	0.99	0.99	0.99

The proposed model generates distributed vector representations in $\mathbb{R}^n$ for each entity in the ontology, these vectors express the probability function of the semantic relations in the ontology. The probability function is expressed as the product of conditional probabilities of the object given the subject and the predicate in the ontology triples. A low dimensional space makes it possible to group semantically similar elements together where the position (the distance and the direction) in the vector space makes it possible to encode the semantics embedded in ontologies in a suitable continuous vector representation.

5 Conclusion

In this paper, we have presented a novel approach for learning low dimensional continuous vector representations for ontology entities, based on taxonomic, semantic relations and restrictions. A multi input feed-forward neural network model have been proposed and used to generate ontology entities vector representations, trained using noise contrastive estimation technique. This research gives a glimpse of the potential of the neural networks and embedding approaches to identify relationships between concepts and instances in ontologies, where the geometric relationships between the generated vector representations in the vector space fully reflect the semantic relations between the corresponding entities in the source ontology. Summing up the results, it can be concluded that the continuous vector representations model is relatively simple to grasp (linear algebra) and easy to implement. It makes it possible to find semantically equivalent entities in an ontology. Experiments also showed that its effectiveness depends for a large part on the quality of the representations in ontologies (concepts and semantic relations between them). In this paper, we have only examined different semantic relationships between concepts and instances. More broadly, we intend to concentrate on exploring different characteristics of the data types properties as well. Further studies are needed to apply the results of our approach in the ontology-engineering field tasks.

Acknowledgments. This paper is funded by the International Exchange Program of Harbin Engineering University for Innovation-oriented Talented Cultivation.

References

1. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquisit. **5**(2), 199–220 (1993)
2. Gómez-Pérez, A., Corcho, O.: Ontology languages for the semantic web. IEEE Intell. Syst. **17**(1), 54–60 (2002)
3. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**(Feb), 1137–1155 (2003)
4. Schwenk, H., Dchelotte, D., Gauvain, J.-L.: Continuous space language models for statistical machine translation. In: Proceedings of the COLING/ACL on Main Conference Poster Sessions, pp. 723–730. Association for Computational Linguistics (2006)
5. Schwenk, H.: CSLM-a modular open-source continuous space language modeling toolkit. In: INTERSPEECH, pp. 1198–1202 (2013)
6. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
8. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 746–751 (2013)
9. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
10. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
11. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000)
12. Yan, S., Xu, D., Zhang, B., Zhang, H.-J.: Graph embedding: a general framework for dimensionality reduction. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, pp. 830–837. IEEE (2005)
13. Yan, S., Xu, D., Zhang, B., Zhang, H.-J., Yang, Q., Lin, S.: Graph embedding and extensions: a general framework for dimensionality reduction. IEEE Trans. Pattern Anal. Mach. Intell. **29**(1), 40–51 (2007)

14. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864. ACM (2016)
15. Ristoski, P., Rosati, J., Di Noia, T., De Leone, R., Paulheim, H.: Rdf2Vec: RDF graph embeddings and their applications. Semant. Web (Preprint), 1–32 (2018)
16. Cochez, M., Ristoski, P., Ponzetto, S.P., Paulheim, H.: Global RDF vector space embeddings. In: d'Amato, C., et al. (eds.) ISWC 2017. LNCS, vol. 10587, pp. 190–207. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68288-4_12
17. Bakarov, A.: A survey of word embeddings evaluation methods. CoRR, abs/1801.09536 (2018)
18. Schnabel, T., Labutov, I., Mimno, D., Joachims, T.: Evaluation methods for unsupervised word embeddings. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 298–307 (2015)
19. Mnih, A., Kavukcuoglu, K.: Learning word embeddings efficiently with noise-contrastive estimation. In: Advances in Neural Information Processing Systems, pp. 2265–2273 (2013)
20. Dyer, C.: Notes on noise contrastive estimation and negative sampling. arXiv preprint arXiv:1410.8251 (2014)
21. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: a new estimation principle for unnormalized statistical models. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 297–304 (2010)
22. Smith, B., et al.: The OBO foundry: coordinated evolution of ontologies to support biomedical data integration. Nat. Biotechnol. **25**(11), 1251 (2007)
23. Whetzel, P.L., et al.: BioPortal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. Nucleic Acids Res. **39**(suppl_2), W541–W545 (2011)
24. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(Nov), 2579–2605 (2008)
25. Morales, J., et al.: A standardized framework for representation of ancestry data in genomics studies, with application to the NHGRI-EBI GWAS catalog. Genome Biology. **19**(1), 21 (2018)