Challenges of Applying Knowledge Graph and their Embeddings to a Real-world Use-case

Rick Petzold[2,3,4], Genet Asefa Gesese[1,3,5], Viktoria Bogdanova[2,4], Thorsten Zylowski[2,4], Harald Sack[1,3,5] and Mehwish Alam[1,3,5]

[1] *FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany*
[2] *CAS Software AG, Germany*
[3] *Karlsruhe Institute of Technology, Institute AIFB, Germany*
[4] *firstname.lastname@cas.de*
[5] *firstname.lastname@fiz-karlsruhe.de*

Abstract

Different Knowledge Graph Embedding (KGE) models have been proposed so far which are trained on some specific KG completion tasks such as link prediction and evaluated on datasets which are mainly created for such purpose. Mostly, the embeddings learnt on link prediction tasks are not applied for downstream tasks in real-world use-cases such as data available in different companies/organizations. In this paper, the challenges with enriching a KG which is generated from a real-world relational database (RDB) about companies, with information from external sources such as Wikidata and learning representations for the KG are presented. Moreover, a comparative analysis is presented between the KGEs and various text embeddings on some downstream clustering tasks. The results of experiments indicate that in use-cases like the one used in this paper, where the KG is highly skewed, it is beneficial to use text embeddings or language models instead of KGEs.

Keywords

Knowledge Graph Embedding, Language Models, Clustering

1. Introduction

As discussed in [1], according to the 2017 Kaggle Machine Learning & Data Science Survey the majority of data scientists use relational data in their work. In significant number of industries such relational data are modeled and stored in relational databases such as MySQL and Oracle. Data scientists make use of the data stored in these databases to perform different machine learning applications such as clustering and classification. However, in order to apply such algorithms directly to the data significant feature engineering efforts are required. Hence, one way to address this issue is to convert the relational databases into a Knowledge Graph (KG)

Woodstock'21: Symposium on the irreproducible science, June 07–11, 2021, Woodstock, NY

kulyabov-ds@rudn.ru (R. Petzold); Manfred.Jeusfeld@acm.org (V. Bogdanova); Manfred.Jeusfeld@acm.org (T. Zylowski)
http://conceptbase.sourceforge.net/mjf/ (V. Bogdanova); http://conceptbase.sourceforge.net/mjf/ (T. Zylowski)
0000-0001-7116-9338 (G. A. Gesese); 0000-0002-9421-8566 (V. Bogdanova); 0000-0002-9421-8566 (T. Zylowski); 0000-0002-9421-8566 (H. Sack); 0000-0002-9421-8566 (M. Alam)

CEUR Workshop Proceedings (CEUR-WS.org)

and then learn embeddings for the obtained KG which in turn will be used as inputs to the downstream tasks.

The relational database (RDB) used in this paper is hosted by the company CAS Software AG[1] and it contains information about German companies, i.e., their addresses, contact persons, industrial sectors, and so on. The database is converted to a KG using the D2RQ [2] tool. In order to apply machine learning algorithms on the KG, it is necessary to transform the KG into low-dimensional vector space while preserving the semantics present in the KG. There exist various approaches proposed for such purposes like DistMult [3] and ComplEx [4]. However, if the created KG is highly skewed with not enough semantics present which is the exact scenario in our use-case, challenges arise when trying to learn representations for the KG, i.e., KGE models do not perform well on KGs with such characteristics. Experiments with some KGE models are conducted to prove this.

Another alternative to KGEs, is to leverage the textual descriptions of the companies and apply text-based embedding models to get latent representations for the companies. The textual descriptions of the companies are extracted from their respective websites. Some downstream company clustering tasks are performed using the representations learned using both the KGE models. The results of the clustering indicates the effectiveness of the text-embeddings over the KGEs. ExCut [5] performs clustering of entities by combining KG embeddings with rule mining methods. Even though ExCut also uses a real-world KG, the quality of the KG is better and suitable for applying KGEs as compared to the use-case (i.e., CAS-KG which is the KG generated from the RDB provided by CAS) that is being addressed in our paper. The contributions of this work are i) analysing real-world datasets for KG embeddings, ii) applying KG embeddings for a downstream task, and iii) comparing text and KG embeddings on real-world datasets.

The rest of the paper is organized as follows: Section 2 discusses the process of converting the RDB to KG followed by the challenges in mapping the KG to Wikidata and learning latent representations for the KG using KGEs. In Section 3, latent representation learning of companies using text embedding models is discussed. The experimental results on downstream clustering tasks are provided in Section 4 followed by the closing remarks in Section 5.

2. Generating KG from Relational Database

Here, converting the RDB to a KG is discussed along with the challenges that occur while trying to enrich the KG with external information and learn latent representations.

2.1. Applying D2RQ to Convert the Relational Database to a Knowledge Graph

The first step is cleaning the database by normalizing it to BCNF and filtering out unnecessary tables, i.e., tables with data that do not provide any useful information to learn representations for companies. After normalizing the database, it is converted to RDF in N-Triples format using D2RQ. As the result of the conversion, there are 5 entity types, 9,794,528 entities, 3 object relations, 21 datatype properties, 74,220,549 triples among which 12,138,554 contain object

[1] https://www.cas.de/start.html

relations and the rest 62.081.995 are triples with datatype properties. The entity types are Company (8,945,631), City (150,377), State (16), Legal Form (45), and Person (6,98,459).

Note that there is no any direct connection between two entities of the same type. Due to this fact, the generated KG (i.e., CAS-KG) is highly skewed and is not rich in semantics. In order to increase the quality of CAS-KG, it is beneficial to enrich the graph with external information.

2.2. Challenges in Mapping CAS-KG to Wikidata

As discussed above CAS-KG is required to be enriched with information from external sources. One of such sources is Wikidata which is a publicly available Linked Oped Data. An attempt has been made to map the companies that are in CAS-KG to items in Wikidata. However, the following two challenges arise when dealing with the mapping **i)** Most of the companies in the CAS-KG are small local businesses which do not have corresponding items in Wikidata. This is observed while trying to perform simple string-based comparison of the names of the companies in CAS with the labels of items that are of type Organization/Business/Company in Wikidata. **ii)** It was possible to map the entities of type LegalForm, and City to Wikidata items. For instance, the Legal Form GmbH in Cas could be mapped to GmbH (Q460178) in Wikidata. However, mapping entities of such types do not actually bring much of usable semantic enrichment without being able to map Companies.

2.3. Applying KGEs on CAS-KG

Here, the challenge is proven by applying some KGE-based link prediction task on CAS-KG. Since CAS-KG is huge in terms of triples and the total number of entities, it is necessary to select a sub-graph for the experiments, which is referred to as CAS286K. CAS286K contains 285,808 entities, 3 relations, 382,964 structured triples, 306,371 training triples, 38,296 test triples, and 38,297 validation triples. The dataset is available at https://github.com/rickpetzold/CAS-Knowledge-Graph.

DistMult [3] and ComplEx [4] KGE models are used to learn representations for the dataset CAS286K. These two models are selected to show the differences that they have in handling asymmetric relations, i.e., unlike ComplEx, DistMult does not perform well with asymmetric relation and all the 3 relations that exist in the CAS286K are asymmetric. Note that the choice of the KGE model does not affect the purpose of these experiments which is to prove that the KG lacks the required quality to apply a KGE model on it. Note that two different ways of initialization are used with the ComplEx model, i.e., random initialization (ComplEx) and initialization with fastText [6] embeddings (ComplEx$_{init}$). The fastText Embeddings are generated by averaging embeddings of the labels and keywords associated with the corresponding entities.

The Stochastic Local Closed World Assumption (sLCWA) [7] training approach is used with model optimization hyperparameter ranges - Embedding dimension: {64,128,256}, Optimizers: {Adam, AdaGrad}, Regularizers: {None, L1, L2}, Weight for L1 and L2: [0.01,1.0), lr:[0.001,0.1), batch size: {128,256,512,1024}, Loss: {BCEL, MRL}, Number of negatives: {1,2, …,30}, and Margin for MRL: {0.5,1.5, …, 9.5}. Number of trials: 10, epochs:100, early stopping with patience of 50 epochs evaluating every 10 epochs. For ComplEx, the optimizer, the loss, and the regularizer are fixed to Adam, BCEL, and L2 respectively so as to reduce computational cost. The optimal hyperparameter values for DistMult and ComplEx are embedding dimension: 128 & 100, Regularizer: L2 & L2, weight: 0.025 & 0.0228, Loss: MRL & BCEL, negative sampler: 6 & 61, optimizer: Adam & Adam, and Batch Size: 256 & 512. Detailed information about sLCWA and the aforementioned loss functions is available in [7].

The results obtained are MRR 0.000034, 0.2, and 0.0074 for DistMult, ComplEx, and ComplEx$_{init}$ respectively. The values of each of these evaluation metrics are too low mainly with DistMult due to some characteristics of the CAS286K dataset which already makes it hard to learn embeddings using KGE approaches. Firstly, the entities of type Company have no incoming relations, i.e., they never occur as tails in the KG which makes the graph highly skewed. This indicates that there exist no single direct connection between any two entities of type Company. Since ComplEx is better than DistMult in dealing with asymmetric relations and most of the relations are asymmetric in CAS286K, the MRR with ComplEx (0.2) is better than with Distmult (0.000034). Moreover, even though initializing ComplEx with FastText embeddings is better than DistMult, it is not better than the randomly initialized ComplEx model due to the fact that only less than 1% of the entities of type company have keywords. Pykeen[2] is used to undertake the experiments.

3. Text Embeddings

As it has already been discussed in Section 2.2 and 2.3, applying the KGE approaches in such highly skewed KG with very limited links between entities is not beneficial. Hence, it is better to apply text embedding models instead as it could be more feasible to find textual descriptions for the companies. Therefore, web crawling is performed to get the textual descriptions of companies and while doing so, those websites containing either very short or non-german text are removed.

In order to learn representations for companies using the crawled texts, different embedding models are used separately, i.e., pretrained fastText and GloVe [8] embeddings, Multilingual Bert [9] & Sentence BERT [10] with/without fine tuning, and Multilingual Universal Sentence Encoder (MUSE) [11]. BERT is fine-tuned on a multiclass-classification task with and without removing stopwords, i.e., BERT$^{nostoprem}_{finetuned}$ using 4049 companies for training & 1200 for validation on 24 classes/sectors and (BERT$_{finetuned}$) using 2552 companies for training & 800 for validation on 16 classes/sectors.

4. Downstream task: Clustering

The clustering task is to group together companies based on their industrial sectors. A gold standard dataset is created with 503 companies where the maximum, minimum, average number of tokens in the textual descriptions of these companies are 695, 209, and 547.67. This gold standard contains 12 industry sectors (classes) in total, the sectors and their corresponding number of companies are: 'Photographers (86)', 'Onlineshop (51)', 'Webdesigner (51)', 'Coaching, Training, and Workshop (50)', 'Real Estate Agent (50)', 'Dentist (50)', 'Advertising Agencies (46)',

[2] https://pykeen.readthedocs.io/en/stable/

Table 1
Clustering results using text, KG, and combined embeddings

		HDBSCAN		BIRCH		K-Means	
	Model	AMI	silh.	AMI	silh.	AMI	silh.
Text embeddings	fastText	0.187	-0.082	0.205	0.056	0.198	0.08
	GloVe	0.234	0.093	0.582	0.132	0.543	0.13
	BERT	0.245	0.017	0.463	0.125	0.432	0.1
	BERT$_{finetuned}$	0.324	-0.022	0.467	0.229	0.46	0.256
	BERT$^{nostoprem}_{finetuned}$	0.488	**0.16**	0.547	0.263	0.548	0.314
	SentenceBERT	0.456	0.068	0.614	0.147	0.627	0.15
	SentenceBERT$_{finetuned}$	0.195	0.143	0.237	**0.451**	0.216	**0.523**
	MUSE	**0.579**	0.158	**0.692**	0.221	**0.676**	0.206
KG embeddings	DistMult	0.013	-0.087	0.01	0.057	0.003	0.106
	ComplEx	0.006	-0.022	0.003	-0.094	0.0003	-0.128
	ComplEx$_{init}$	0.014	-0.053	0.012	0.088	0.007	0.004
Combined	MUSE + ComplEx$_{init}$	0.015	-0.03	0.01	0.074	0.006	0.151

‘Consulting (37)’, ‘IT Services (25)’, ‘Online Agencies (23)’, ‘Attorney (21)’, and ‘Travel Agencies (13)’.

BIRCH [12], HDBSCAN [13], and K-means are selected for the clustering task. For BIRCH the hyperparameters are the number of clusters 1-20, branching factor 10-200, and threshold 0.1-0.9. For HDBSCAN the minimal samples are 1-50 and the minimal cluster size is 2-100 whereas for K-means the number of clusters is 2-30. As the result in Table 1 indicates, the text embeddings give better results as compared to the KG embeddings in the clustering task. This is due to the highly skewed nature of the CAS286K dataset. Information from external resources is required in order to improve the KGE results. Note that, the MRR results with ComplEx$_{init}$ is not better than ComplEx on the link prediction task. However, the opposite holds on the clustering task because 82% of the companies in the gold standard have keywords. Note that the combined embeddings are generated by simply concatenating representations from MUSE and ComplEx$_{init}$.

5. Conclusion

In this paper, the challenges in applying KGE models on real world use-cases are discussed. Experiments on clustering tasks are conducted using as inputs latent representations learned by applying both KGEs and text embeddings separately. The results of the experiments prove the initial analysis that are made about KGEs not working well on datasets with very low quality such as CAS286K.

References

[1] M. Cvitkovic, Supervised learning on relational databases with graph neural networks, arXiv preprint arXiv:2002.02046 (2020).

[2] C. Bizer, A. Seaborne, D2rq-treating non-rdf databases as virtual rdf graphs, in: Proceedings of the 3rd International Semantic Web Conference, 2004.

[3] B. Yang, W.-t. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: International Conference on Learning Representations (ICLR), 2015.

[4] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, ICML'16, JMLR.org, 2016, p. 2071–2080.

[5] M. H. Gad-Elrab, D. Stepanova, T. Tran, H. Adel, G. Weikum, Excut: Explainable embedding-based clustering over knowledge graphs, in: Proceedings of 19th International Semantic Web Conference, 2020, pp. 218–237.

[6] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2016).

[7] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, M. Galkin, S. Sharifzadeh, A. Fischer, V. Tresp, J. Lehmann, Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework, arXiv preprint arXiv:2006.13365 (2020).

[8] J. Pennington, R. Socher, C. Manning, GloVe: Global vectors for word representation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: NAACL, 2019.

[10] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019.

[11] D. M. Cer, Y. Yang, S. yi Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, R. Kurzweil, Universal sentence encoder, ArXiv abs/1803.11175 (2018).

[12] T. Zhang, R. Ramakrishnan, M. Livny, Birch: An efficient data clustering method for very large databases, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, 1996, pp. 103–114.

[13] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, AAAI Press, 1996, p. 226–231.