Embedding Metadata-Enriched Graphs*

Stefan Bachhofner[1][0000−0001−7785−2090], Peb Ruswono Aryan[2][0000−0002−1698−1064], Bernhard Krabina[1,4][0000−0002−6871−3037], and Robert David[3]

[1] Vienna University of Economics and Business, Institute for Data, Process and Knowledge Management, Welthandelsplatz 1, 1020 Vienna, Austria
{forename.surname}@wu.ac.at
[2] Vienna University of Technology, Vienna, Austria peb.aryan@tuwien.ac.at
[3] Semantic Web Company, Vienna, Austria
[4] KDZ – Centre for Public Administration Research, Vienna, Austria

Abstract. This paper presents an on-going research where we study the problem of embedding meta-data enriched graphs, with a focus on knowledge graphs in a vector space with transformer based deep neural networks. Experimentally, we compare ceteris paribus the performance of a transformer-based model with other non-transformer approaches. Due to their recent success in natural language processing we hypothesize that the former is superior in performance. We test this hypothesizes by comparing the performance of transformer embeddings with non-transformer embeddings on different downstream tasks. Our research might contribute to a better understanding of how random walks influence the learning of features, which might be useful in the design of deep learning architectures for graphs when the input is generated with random walks.

Keywords: Graph Embedding · Knowledge Graph Embedding · Deep Learning · Metadata · Random Walks

1 Introduction

Deep Learning (DL) has drastically improved the state-of-the-art on many tasks in Natural Language Processing (NLP) and Computer Vision (CV) since its breakthrough in 2012 [2]. For the former, [5] claim that DL is able to learn word embeddings which capture material science concepts without any supervision, and that these embeddings can be used to predict materials years before their discovery. This success has been largely attributed to its ability to learn features of a concept in an unsupervised manner, therefore eliminating most, if not all, the

* This research has received funding form the Teaming.AI project, which is part of the European Union's Horizon 2020 research and innovation program under grant agreement No 957402.

need for feature engineering. Unsurprisingly, this success and the prospect of next to none feature engineering lead to an interest in this machine learning technique from both the graph and semantic web research community. Specifically, a stream of research emerged which dedicates itself to learning representations of nodes, edges, sub-graphs, or a whole graph, and any combination of these.

Two approaches for embedding are DeepWalk, which claims to be the first that introduces DL for network analytics, and RDF2Vec, where the former is from the literature on graph embeddings while the latter is from the specialized community on Knowledge Graph (KG) embeddings. Both first use random walks to generate sequences which are then fed into a technique originated in NLP. They hence treat the result of a random walk as being equivalent to a sentence.

In our research, we are interested in enriching these random walks with meta-data present in the graph or KG and the effect this has on different DL models. In particular, we study the ability of transformer based DL models to learn embeddings from random walks enriched with meta-data. We hypothesize that the former is superior ceteris paribus to non-transformer methods in learning representations evaluated by their performance on downstream tasks.

2 Background on Graph Embedding Approaches with Deep Learning and Random Walks

At the core of graph embedding approaches with DL and random walks is the idea to represent the graph as a sequence of random walks, which is the input to the DL model [1]. The random walk is hence a feature engineering preprocessing step to enable the use of existing DL embedding approaches, which are usually from NLP. Two frequently used approaches are Continuous bag-of-words (CBOW) and skip-gram, which are explained in more detail in the next paragraph. DeepWalk and RDF2Vec are examples for approaches that use them, where the former is from the literature on graph embeddings, while the latter is more specialized for knowledge graphs serialized with the Resource Description Framework (RDF). For this embedding family, the random walks are of paramount importance as they are the input to the DL model. The model hence relies on the properties of the paths created by the random walk, which in turn logically implies that the DL model is constrained by (i) the degree to which they preserve the graph properties, and (ii) the expressiveness of these paths. We hypothesize that adding meta-data leads to an increase in performance for a given task ceteris paribus, given the DL model is capable to learn the structure of meta-data enriched paths. Which is the motivation for this research. In the next paragraph, we exemplary describe RDF2Vec.

RDF2Vec is a KG embedding approach specifically designed for RDF serialized KGs, which uses random walks to first generate sequences of a fixed length d. These sequences are then fed into a 3-layer multi-layer perceptron for training [3,4]. The vector representation can then be obtained from the hidden layer. In the original paper, the authors set d to either 4 or 8, and use 500 or 200 walks per entity, depending on the data set. These sequences can be generated with

either random graph walks or the Weisfeiler-Lehman algorithm. As a training strategy one can either use CBOW or Skip-gram, where one is the inverse of the other. In the former, the 3-layer perceptron attempts to predict one missing word in a sequence, while in the latter it attempts to predict surrounding words of a given word.

3 Related Work

In our research we focus on embedding graphs with random walks and DL. Naturally, this implies similarities with these approaches. However, we differentiate ourslefes in two important ways. First, our graph walks can in principle contain a massive amount of meta-data which need to be processed by the DL model. Which leads to the second distinction, which is the use of a transformer based model as they have started to outperform recurrent models in NLP [6]. This might indicate an increased capability to learn structure from sequences, which actually is the research question we contribute to.

Due to their success in NLP, the semantic web community has also started to investigate DL for graph embeddings. For example, [7] use them to generate embeddings for context-aware and temporal KGs. Their initial empirical results may provide evidence for their increased capability to learn structure from sequences. In particular, they report improvement by a factor of up to 15 on Hit@3 compared to their baseline models TransE, SimplE, and Hol3. However, they also report a decrease in performance and no increase at all for Hit@1. In addition, they acknowledge that the baseline models have performance issues which, as they argue, may be due to a skewed distribution in the data set. Our research is similar to theirs as we also study the problem of designing DL architectures that are best suited for learning structure from sequences with additional information. We however have a focus on meta-data, while they have a focus on time and context. This is not a sharp distinction as context may be added to the KG via meta-data. Finally, our research might shine light on why the transformer models were not able to improve the performance on the above mentioned Hit@1.

In our research we do not intend to introduce a new DL architecture or propose a pre-processing method as the authors above. Instead, we contribute to a better understanding of how DL learns the graph structure when it has only access to a set of paths generated by random walks. This means that we will not alter the architecture and its hyperparameters (for example the number of filters or the kernel size), except for the case where we want to keep the number of parameters approximately equal among all architectures. Please find more experimental details in the next section.

4 Experiment Details

We have a strong focus on reproducability and comparability in our research (Fig. 1). To ensure that, we take the following steps. First, we will take care to control all involved random number generators with seeds and will mention

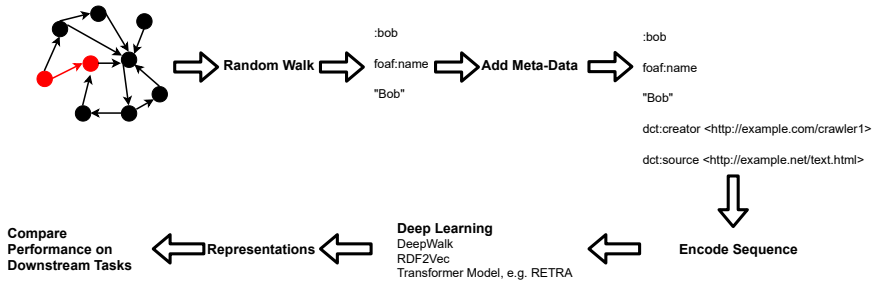

Fig. 1: Embedding Metadata-Enriched Graphs with Random Walks and DL.

them in the paper. Second, we will separate the random walk generation from the training loop as we first generate the sequences and save them. These will then be the input to the approaches. Third, will record and save the order in which these sequences are presented to the DL model. The sequences and order will be made publicly available. Fourth, we will perform preliminary short experiments that are specifically designed to test reproducability. Fifth, we will make sure that the number of model parameters are approximately equal given the respective approach and architecture restrictions, e.g. some approaches may have different parameter scaling factors. Evaluation data sets we are considering are, inter alia, American Association of University Professors (AAUP), Angewandte Informatik und Formale Beschreibungsverfahren (AIFB), and British Geological Survey (BGS).

5 Discussion and Outlook

In this paper, we report on an on-going research were we study the problem of embedding graphs represented as a sequence of meta-data enriched random walks with DL. We are in particular interested in the embedding capabilities of transformer based models. In our experiment design we put sizeable effort in ensuring reproducability and comparability. Since the quality of embeddings have a huge influence on downstream tasks (e.g. node prediction and link prediction), our research might have broad implications for many streams of reserach. Among others, results of our research might have an influence on the quality of knowledge completion and fact checking technologies, e.g. detecting fake news and tracing back a news story to its origins. Further, our research might aid the design of DL architectures for graphs if the input is the result of random walks.

References

1. Cai, H., Zheng, V.W., Chang, K.C.C.: A comprehensive survey of graph embedding: Problems, techniques, and applications. IEEE Transactions on Knowledge and Data Engineering **30**(9), 1616–1637 (2018)

2. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
3. Ristoski, P., Paulheim, H.: Rdf2vec: Rdf graph embeddings for data mining. In: The Semantic Web – ISWC 2016. pp. 498–514. Springer International Publishing, Kobe, Japan (October 2016)
4. Ristoski, P., Rosati, J., Di Noia, T., De Leone, R., Paulheim, H.: Rdf2vec: Rdf graph embeddings and their applications. Semantic Web **10**(4), 721–752 (2019)
5. Tshitoyan, V., Dagdelen, J., Weston, L., Dunn, A., Rong, Z., Kononova, O., Persson, K.A., Ceder, G., Jain, A.: Unsupervised word embeddings capture latent knowledge from materials science literature. Nature **571**(7763), 95–98 (2019)
6. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, u., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 6000–6010. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
7. Werner, S., Rettinger, A., Halilaj, L., Lüttin, J.: RETRA: Recurrent transformers for learning temporally contextualized knowledge graph embeddings. In: The Semantic Web. pp. 425–440. Springer International Publishing, Cham (2021)