Toward Predicting Impact of Changes in Evolving Knowledge Graphs

Romana Pernischová[1], Daniele Dell'Aglio[1], Matthew Horridge[2], Matthias Baumgartner[1], and Abraham Bernstein[1]

[1] University of Zurich, Zurich, Switzerland
{pernischova,dellaglio,baumgartner, bernstein}@ifi.uzh.ch
[2] Stanford University, Stanford, USA
matthew.horridge@stanford.edu

Abstract The updates on knowledge graphs (KGs) affect the services built on top of them. However, changes are not all the same: some updates drastically change the result of operations based on knowledge graph content; others do not lead to any variation. Estimating the impact of a change ex-ante is highly important, as it might make KG engineers aware of the consequences of their action during KG editing or may be used to highlight the importance of a new fragment of knowledge to be added to the KG for some application. The main goal of this contribution is to offer a formalization of the problem. Additionally, it presents some preliminary experiments on three different datasets considering embeddings as operation. Results show that the estimation can reach AUCs of 0.85, suggesting the feasibility of this research.

Keywords: KG Evolution · Impact · Embeddings.

1 Introduction

Knowledge graphs (KG) and ontologies[1] change over time. Such graphs are updated by inserting new knowledge and removing or changing outdated and wrong information. As an example, consider the computation of a (logical) materialization: a small change in the schema can have a big impact on the KG, significantly vary the number of materialized axioms or even its consistency. However, not every change in the KG leads to significant changes in the materialized graph. The computation of the materialization consumes considerable amounts of resources. Consequentially, the indication of a potentially big difference from the old materialization to the new one would signal the necessity for recomputation.

This topic is not at all new, but so far studies come from different directions. It relates to stream processing since the process executed over a stream

[1] We use the terms knowledge graph and ontology interchangably because of the different datasets used in the case study.

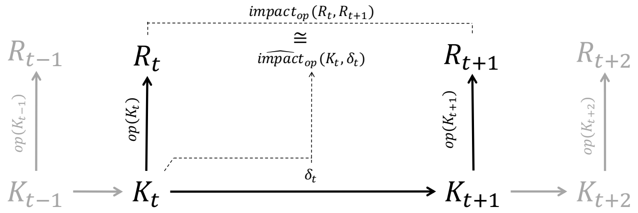

Figure 1: General model of the problem setting with K_t being the KG at time t, δ_t the changes leading to K_{t+1}, $op(\cdot)$ the operation executed on the KG, R_t the result of $op(\cdot)$, $impact_{op}(\cdot)$ the impact, and $\widehat{impact}_{op}(\cdot)$ the estimation.

of data changes results as the data comes in [2]. Moreover, Gross et al. [5] examine how the changes in an biomedical ontology impact previously conducted functional analysis. Gottron and Gottron [4] evaluate how indexes are affected by the evolution of the LOD dataset.

In this poster, we present a formalization of the problem of estimating the impact on a operation result over an evolving KG. Furthermore, we show first experiments with three different datasets on embeddings operations [8].

2 The Problem of Predicting the Impact

Fig. 1 shows the conceptualization of the impact prediction problem. A knowledge graph K is a set of triples (s, p, o), where s and o are two resources connected by a predicate/relation p. We define an *evolving knowledge graph* $\mathcal{K}$ as a sequence $(K_1, K_2, \ldots, K_t, K_{t+1}, \ldots)$, where K_t denotes the KG at the time instant t. This definition of evolving KG is similar to the one of *ontology stream* proposed by Ren and Pan in [7]. Let K_t and K_{t+1} be two consecutive versions of $\mathcal{K}$. The update of $\mathcal{K}$ between t and $t+1$ is described by a set of changes δ_t. δ indicates a set of edits that are authored by one or more agents, such as ontology engineers or maintenance bots.

We define $op(\cdot)$ as a function which applies to a KG and produces a result R. $op(\cdot)$ takes as arguments a KG and zero or more additional parameters if necessary. When the operation $op(\cdot)$ is applied to $\mathcal{K}$, it will create a sequence of results $\mathcal{R} = (R_1, R_2, \ldots, R_t, \ldots)$, where R_t is the result of $op(\cdot)$ on K_t.

Given K_t and K_{t+1}, the respective results R_t and R_{t+1} can be the same (if the changes δ_t do not affect the result of $op(\cdot)$), or they can differ. We model this comparison through the function $impact_{op}(R_t, R_{t+1})$, which represents the impact that the evolution had on the results of $op(\cdot)$.

Therefore, the goal of this research is to *build an estimator of* $impact_{op}(\cdot)$, which does not require the computation of all $op(\cdot)$ results. We indicate the estimator with $\widehat{impact}_{op}(K_t, \delta_t)$ since it takes as arguments a knowledge graph and the set of changes leading to a new version. Moreover, $op(\cdot)$ should be considered as well, since different operations may lead to different impact functions.

3 Case study

We discuss our model in one case study considering embeddings as $op(\cdot)$.

Definition of Impact. To evaluate embeddings of two snapshots, we make the comparison using neighborhoods. Taking the 100 closest neighbors of a particular node, we compare how many of these are also in the neighborhood of the same node in the embedding of the next version. The aggregation for the whole graph is then done via the *sum* and the *mean*, producing two impact measures.

Definition of Features.[2] Different measures have been established in the network analysis community such as degree centrality or clustering indicators. Further, we calculate sparseness and graph entropy measures. To use these measures as features, we calculated a simple difference between the graph measure value in two versions of the graph. Other features are based on change actions. A change action is an item in δ. We used the classification introduced by Hartung, Gross, and Rahm [6] to identify *high-level change actions* such as `move`, `merge`, `split`, `add`, and `delete` node. Subsequently, we calculate degree, closeness, betweenness, and degree centrality for the affected nodes.

Data. We test the impacts and features defined above through experiments on three datasets: Bear-B-instant (BB), the Gene Ontology (GO) and an anonymized and de-identified WebProtege Ontology (WP). The BB dataset was produced using the 100 most volatile subjects of the DBpedia [3]. In our experiments, we use 3'923 datapoints. GO is a well known ontology in the biomedical domain and has been maintained by the Gene Ontology Consortium since 2000 [1]. It has (only) 99 versions producing 90 datapoints that we were able to use. Lastly, we randomly selected a WebProtege Ontology (WP) based on three criteria: it (1) has at least 2'000 snapshots, (2) was edited by more than two users, and (3) has less than 30'000 axioms.

The $\widehat{impact}$ estimator. We split the data into a training and testing datasets, where the training dataset contains 70% of the data points. For learning, we use 10-fold cross validation. The area under the receiver operating characteristics curve (AUC; also called c-statistic) is calculated to compare the models after testing. To calculate the AUC for regression, we determine a threshold for the observed impact to turn the regression into a binary classification. Features were selected using Pearson correlation (*corr*) and 10-fold Ridge regression (*ridge*).

Results. We test the performance of $\widehat{impact}$ by running three experiments on the considered datasets and impact measures. Table 1 reports the AUC results of the experiments. For the BB dataset, performance does not exceed an AUC of 0.63. On the other hand, for WP performance reaches 0.851. With GO, the only dataset including change action features, the performance is 0.8 and higher on multiple occasions.

[2] The complete list of the features can be found in the source code at `http://tinyurl.com/y29t77v4`

Table 1: Results of the experiments.

	BB				GO				WP			
	sum		mean		sum		mean		sum		mean	
	corr	ridge	corr	ridge	corr	ridge	corr	ridge	corr	ridge	corr	ridge
GLM	0.534	0.553	0.569	0.612	0.500	0.811	0.843	0.830	0.796	0.796	0.622	0.617
SVMLin	0.536	0.553	0.580	0.622	0.500	0.811	0.744	0.764	0.787	0.782	0.620	0.611
SVMRad	0.564	0.535	0.600	0.640	0.650	0.661	0.753	0.694	0.681	0.767	0.851	0.764
Features	4	15	4	12	21	9	17	7	13	10	2	10

4 Conclusions

Following the butterfly effect, a small change in a KG can lead to large differences in operation results. This poster presents the problem setting and definition along with a case study using embeddings over three datasets and two impact measures. The successful experiments show that it is possible to predict the impact. We can therefore decide when an update to the operation is necessary, potentially saving valuable resources.

This study triggers new questions. For instance, is it possible to construct a general model spanning over different datasets and operations? We need to gain a deeper understanding of the common factors among datasets and operations. Which is why we plan to run further experiments in this direction.

This study clearly shows that the investigation of KG-evolution and the impact of changes on KGs is fruitful and needs to be intensified.

References

1. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene Ontology: Tool for the unification of biology. Nat. Genet. **25**, 25 EP – (05 2000/05/01/online)
2. Chen, J., Lecue, F., Pan, J.Z., Chen, H.: Learning from Ontology Streams with Semantic Concept Drift. In: IJCAI. pp. 957–963. ijcai.org (2017)
3. Fernandez, J.D., Umbrich, J., Polleres, A., Knuth, M.: Evaluating Query and Storage Strategies for RDF Archives. In: SEMANTICS. pp. 41–48. ACM (2016)
4. Gottron, T., Gottron, C.: Perplexity of Index Models over Evolving Linked Data. In: ESWC. LNCS, vol. 8465, pp. 161–175. Springer (2014)
5. Gross, A., Hartung, M., Prüfer, K., Kelso, J., Rahm, E.: Impact of ontology evolution on functional analyses. Bioinformatics **28**(20), 2671–2677 (2012)
6. Hartung, M., Gross, A., Rahm, E.: COnto-Diff: Generation of complex evolution mappings for life science ontologies. Journal of Biomedical Informatics **46**(1), 15–32 (Feb 2013)
7. Ren, Y., Pan, J.Z.: Optimising ontology stream reasoning with truth maintenance system. In: CIKM. pp. 831–836. ACM (2011)
8. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge Graph Embedding: A Survey of Approaches and Applications. IEEE Trans Knowl Data Eng **29**(12), 2724–2743 (2017)