# The Butterfly Effect in Knowledge Graphs: Predicting the Impact of Changes in the Evolving Web of Data

Pernischova, Romana

Abstract: Knowledge graphs (KGs) are at the core of numerous applications and their importance is increasing. Yet, knowledge evolves and so do KGs. PubMed, a search engine that primarily provides access to medical publications, adds an estimated 500'000 new records per year - each having the potential to require updates to a medical KG, like the National Cancer Institute Thesaurus. Depending on the applications that use such a medical KG, some of these updates have possibly wide-ranging impact, while others have only local effects. Estimating the impact of a change ex-ante is highly important, as it might make KG-engineers aware of the consequences of their actions during editing or may be used to highlight the importance of a new fragment of knowledge to be added to the KG for some application. This research description proposes a unified methodology for predicting the impact of changes in evolving KGs and introduces an evaluation framework to assess the quality of these predictions.

# The Butterfly Effect in Knowledge Graphs: Predicting the Impact of Changes in the Evolving Web of Data

Romana Pernischová

University of Zurich, Zurich, Switzerland
pernischova@ifi.uzh.ch

**Abstract.** Knowledge graphs (KGs) are at the core of numerous applications and their importance is increasing. Yet, knowledge evolves and so do KGs. PubMed, a search engine that primarily provides access to medical publications, adds an estimated 500'000 new records per year—each having the potential to require updates to a medical KG, like the National Cancer Institute Thesaurus. Depending on the applications that use such a medical KG, some of these updates have possibly wide ranging impact, while others have only local effects. Estimating the impact of a change ex-ante is highly important, as it might make KG-engineers aware of the consequences of their actions during editing or may be used to highlight the importance of a new fragment of knowledge to be added to the KG for some application. This research description proposes a unified methodology for predicting the impact of changes in evolving KGs and introduces an evaluation framework to assess the quality of these predictions.

**Keywords:** Knowledge Graph Evolution · Ontology Evolution · Impact.

## 1 Relevancy

Knowledge graphs (KGs) or ontologies [1] capture the knowledge of a particular domain. They are at the core of applications, such as search, logic-based reasoning, and inductive model training. KGs represent the knowledge of a universe that evolves. Thus, such KGs capture the *current* knowledge of a particular domain: their content can change to (1) include new facts or axioms, (2) adhere to the changing world, or (3) correct wrong or imprecise knowledge. It is natural to ask ourselves how this evolution affects the services built on top of it.

Let us consider the following example: National Cancer Institute thesaurus (NCIt) [6] is a KG that relies on new research accessible through PubMed, which provides access to medical publications. Researchers use NCIt to compute recommendations and tag instances automatically [16]. However, NCIt is not static: every year PubMed receives roughly 500'000 new records, in addition to

[1] KG and ontology are used interchangeably.

updates to existing ones. Therefore, NCIt needs revisions with a subsequent need to also update the recommendation and materialization tasks built on top of it.

The results of operations built on top of KGs, e.g. search results, materialization, machine learning models, can shift strongly due to those changes. Taking actions to adapt to changes may be expensive: in some cases results can be incrementally updated, while in others they must be recomputed from scratch, leading to a potential high usage of resources or costly revisions of previous decisions. It is worth noting that not every change has the same *impact*, e.g. renaming a concept would have a minimal impact on the materialization, while removing a central node could significantly change the materialization.

Initial research, e.g. [15], focused on studying KG evolution, without considering the tasks relying on it, while later research, e.g. [9,16], focused on specific tasks or KGs. However, these studies are usually limited to one application scenario or one specific KG, hampering the generalization and comparability of their insights. As a consequence, the research community lacks a comprehensive understanding of changes, their impact, and mechanisms that help adapting to them. Hence, a framework is needed that helps to (1) understand changes on KGs leading to a general in-depth study, (2) estimate the impact of changes on a task that relies on the evolving KG, and (3) develop mechanisms for adaptation to evolutionary KG changes. Such mechanisms can include maintenance processes that are triggered when changes cause significant impact.

To enable the study of impact over different tasks and KGs, a unified methodology is necessary. This would ensure that the results are comparable to each other. Therefore, the impact prediction should be embedded in a framework enabling consistency in approach and comparison of results. Additionally, adaptation to real world scenarios would be made easier.

## 2 Problem Statement

The challenge I want to address in my PhD studies is the development of a methodology to predict the impact of KG changes on the results of respective functions. Using a unified setting, I can develop such methodologies and compare, evaluate, and improve them iteratively as done in design science approaches [13]. Inside each iteration of the methodology, I will define impact, features, selection and prediction algorithms to build a predictor of KG evolution impact. A first general methodology is introduced in Section 5. In the following paragraphs, I introduce some terminology and explain the formal setting of the 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 $p$. In Fig. 1, I 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 [18]. 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 $\delta_t$. $\delta$ indicates a set of edits that are authored by one or more agents, such as ontology engineers or maintenance bots.

As shown in Fig. 1, I define the operation $op_1(\cdot)$ as a function which accepts a KG as argument and produces a result $R$. When the operation $op_1(\cdot)$ is applied to $\mathcal{K}$, it creates a sequence of results $\mathcal{R} = (R_1, R_2, \ldots, R_t, \ldots)$, where $R_t = op(K_t)$. Examples of such operations might be the materialization, the computation of an embedding space, or some recommendations. Therefore, the Fig. 1 also shows a second operation $op_2(\cdot)$ that is defined accordingly.

Given $K_t$ and $K_{t+1}$, the respective results $R_t$ and $R_{t+1}$ can be the same, i.e. when the changes $\delta_t$ do not affect the result of $op_1(\cdot)$, or they can differ. I model this comparison through the function $impact_{op_1}(R_t, R_{t+1})$, which represents the impact that the evolution had on the results of $op_1(\cdot)$. In the case of $op_1(\cdot)$ being the calculation of embeddings the $impact_{op_1}(R_t, R_{t+1})$ would be the comparison of the embedding of $K_t$ and $K_{t+1}$ for example using the overall loss, neighborhood similarity, or link prediction hit rate. In a practical setting, it may be too expensive to compute the impact at every time instance or I may want to know the impact *before* applying a change.

I indicate the estimator with $\widehat{impact}_{op_1}(K_t, \delta_t)$, since it takes as arguments a KG and the set of changes leading to the next version. Moreover, $op_1(\cdot)$ should be considered as well, since different operations may entail different impact functions. Ideally, my research would lead to the definition of general impact estimators $\widehat{impact}(\cdot)$, which are independent of any specific operation as shown by $op_1(\cdot)$ and $op_2(\cdot)$ in Fig. 1.

Taking this formal setting, the problem of my research lies in the definition of a general methodology applicably within this setting. Inside the methodology, impact has to be defined together with features describing $\delta$, the feature selection procedure, and the prediction algorithm. As mentioned before, the methodology will be improved iteratively by applying it to use cases with adjusted factors ($impact_{op_1}(\cdot)$, $op_1(\cdot)$, $\delta$, $\mathcal{K}$). Each iteration will yield estimators, which performances I can compare and consequently refine the methodology. The performance of predictors is given with the distance between estimation $\widehat{impact}_{op_1}(\cdot)$ and real value $impact_{op_1}(\cdot)$. The comparison of methodologies is based on the performance of predictors built by them.

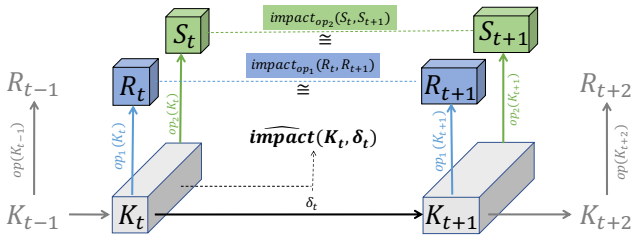


Fig. 1: General model of the problem setting with $K_t$ being the KG at time t, $\delta_t$ the changes on $K_t$ which lead to $K_{t+1}$, $op_1(\cdot)$ the operation executed on the KG, $R_t$ the result of $op_1(\cdot)$ and $impact_{op_1}(K_t, \alpha_t)$ the impact of $\delta_t$ over the result of $op_1(\cdot)$ on $K_t$. $op_2(\cdot)$ is another operation equivalent to $op_1(\cdot)$.

## 3 Related Work

Many researchers have focused on topics close to my proposed one. Essentially, they can be split in two groups, those being research on the evolution of KGs and research about the impact of KG evolution on KG-based tasks.

**Evolution of KGs.** Zablith et al. [23] survey various evolution processes. In addition, Hartung et al. [12] show the different tools for managing, exploring, and propagating changes on knowledge graphs. These studies focus on how ontologies are maintained. They do not investigate the consequences of updates. In contrast, I would like to focus on the impact of KG evolution on the operations that rely on that KG. Rashid et al. [17] use the evolution of a knowledge graph to asses the quality of the current version by examining consistency, completeness, persistence, and historic persistence. In my study, those quality metrics can be seen as impact measures to be predicted rather than observed measures for quality assessment. Another related area aims at detecting ontology changes and classify them. OntoDiff [22] is a tool that enables the user to detect changes between two versions of the same graph. It works by identifying semantically equivalent elements between the ontologies. COnto-Diff [11] and the integrated CODEX [10] both detect changes and group low level changes into high level change actions. They provide a simple classification and a rich action semantics. Klein and Noy [14] developed an ontology describing 80 basic changes. Similarly to COnto-Diff and CODEX, they also introduce a notion of complex changes, showing how they help in the interpretation of consequences for data and entities.

**Impact of KG Evolution.** SemaDrift [19] calculates various semantic drift measures between versions of ontologies. An alternative notion of semantic drift (assuming that code can be seen as a kind of ontology describing functionality) has also been investigated in the context of code repositories and bug fixing [3]. The authors discovered a semantic drift that had a grave influence on the defect prediction quality. Thus, the evolution of the repository showed an impact on the prediction. Chen et al. [1] discuss how learned models become less accurate as a stream evolves semantically. Impact is measured with accuracy loss and changes are addressed using concept drift. In Know-Evolve [21], the authors apply machine learning over the graph and predict re-occurrence of events. The time component directly affects the results of the reasoning from which an impact could be derived. Goncalves et al. [7] investigate the influence of change classification on the set of entailed axioms in the next version. Gross et al. [9] examine how the changes in an ontology impact previously conducted functional analysis. Osborne et al. [16] present an analysis of the selection of concepts for a new version by evaluating the performance of four different tasks. Gottron and Gottron [8] implemented various indexing methods and evaluate how the index is affected by the KG evolution. All of these mentioned studies focus either on a specific task or a specific knowledge graph. One of the goals of my research is to define a general methodology that would also capture previous studies. This means, that the approach of other researchers could be mapped to my proposed methodology making their results comparable.

## 4 Research Questions and Hypotheses

The related work shows that it is possible to predict the impact of changes for specific tasks and knowledge graphs. My first research question builds on those studies, and generalizes them to define a methodology. The goal is to predict the impact on tasks of KG changes using a general methodology that can be applied on different tasks and KGs.

***RQ-I:*** *Can I define a unified methodology to develop predictors for the impact of a task on an evolving knowledge graph?*

The development of the methodology is a design science task [13]. It requires the iterative development of a series of improving frameworks that get evaluated in a practical context. Using the insights from the evaluation, the next iteration of the framework development can be started. A first version of the framework will be introduced in Section 5.

The implementation of such a methodology has various requirements. It should be general enough to be applicable over different tasks and KGs by allowing the definition of impact metrics for every task. In addition, the definition of features is another factor inside the methodology that can be compared and therefore also improved between interations.

I plan to investigate this question in practical settings by studying: (1) KGs of different topics, maintenance processes, size, and structure; (2) features portraying different aspects of a change using detailed change information (change action features) or general graph measures; (3) operations $op_1(\cdot)$ that differ in complexity, stochasticity, and incremental compatibility; as well as (4) impact measures varying in value range, e.g., Boolean, categorical, or continuous, and nature, e.g., structural or semantic. Implementing and improving the methodology using a broad range of possibilities would set the requirements for a generalized methodology, applicable to a wide number of cases.

Within the settings of these variations and the proposed methodology, I will then investigate the following hypotheses to answer if the said methodology can serve as candidate answer to *RQ-I*. Please note that this list is not exhaustive:

***H-I:*** *The impact prediction performance (AUC) of a methodology using* ***change action features*** *is higher than the performance with general graph features.*

***H-II:*** *The impact prediction performance (AUC) of a methodology using a* ***small KG*** *is higher compared to the performance using a KG with twice as many axioms.*

***H-III:*** *The impact prediction performance (AUC) of a methodology using a* ***stochastic operation****, such as embedding calculation, is lower compared to the performance with a logical operation, such as materialization.*

***H-IV:*** *The impact prediction performance of a methodology using $\delta_t$* ***of size one*** *is higher compared to the performance with using $\delta_t$ of larger size.*

The results leading to the answering of the hypotheses and of *RQ-I* will support my second research question. The methodologies set the foundation to study how features correlate among different knowledge graphs, tasks, and impact measures. This is captured in the second research question:

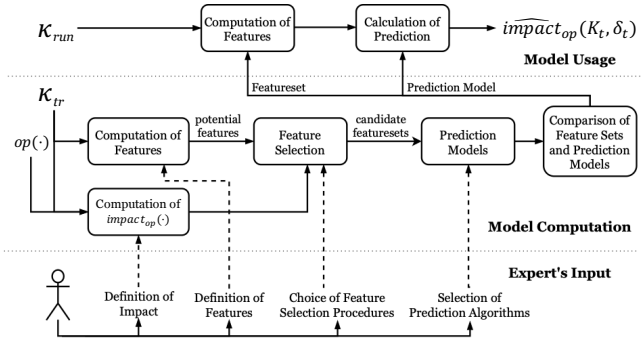


Fig. 2: Framework usage. The upper part shows how $\widehat{impact}_{op_1}(\cdot)$ is estimated and the bottom part illustrates the procedure for learning the estimator.

***RQ-II:*** *Can I build a generalized model for predicting impact that is as accurate as models specifically built for one knowledge graph, task and impact measure?*

To arrive at a generalized model, investigating the relationship between change features and the impact is necessary. It would reveal which impact measures interact with which change features. In this interaction, the direction of the relationship between features and impact measures is important. Using a feature that correlates with the first impact measure positively but correlates with the second impact measure negatively would be counterproductive in prediction of a general impact. Therefore, I can identify features that are common among the prediction models of different task impacts.

***H-V:*** *A feature showing the same coefficient direction* ***between KGs*** *produces a better performance compared to features with coefficients going in opposite directions.*

***H-VI:*** *A feature showing the same coefficient direction* ***between task****s produces a better performance compared to features with coefficients going in opposite directions.*

***H-VII:*** *A feature showing the same coefficient direction* ***between tasks and KG*** *produces a better performance compared to features with coefficients going in opposite directions.*

## 5 Approach

I propose a first methodology, which I call *CHIMP*, to initiate investigation of *RQ-I*. The name is derived from change impact, since at its core it predicts the impact of KG changes on a task. CHIMP will be refined in future iterations. In Fig. 2, CHIMP provides a workflow to predict the impact using the setting shown. It is applicable to individual pairs of KG and operation without imposing requirements.

The bottom section of Fig. 2 shows decisions. These need to be made by one (or more experts) and used as input for CHIMP. An KG expert needs to define how the impact between task results should be measured. Further, a KG expert defines the features later used as input for the learning models. However, the decisions about feature selection and learning algorithm require knowledge about data science and its algorithms.

The middle section of Fig. 2 shows relevant steps of CHIMP: the evolving KG $\mathcal{K}$ is used to determine the set of potential features, while the operation $op_1(\cdot)$ sets the basis to define $impact_{op_1}(\cdot)$. First, a training subsequence $\mathcal{K}_{tr}$ of $\mathcal{K}$ is used to identify the relevant features and train the prediction model. Secondly, CHIMP computes both (1) the impact $impact_{op_1}(\cdot)$ for each pair of consecutive KGs in $\mathcal{K}_{tr}$ and (2) the values of the potential features using as input the changes that update every version of $\mathcal{K}_{tr}$ to the successive one. Subsequently, CHIMP selects a set of candidate features by exploiting impact and potential feature values. Each candidate feature set is used to build several prediction models. Finally, CHIMP identifies the best model and feature set to be used for prediction. The upper part of Fig. 2 shows how to estimate $impact_{op_1}(\cdot)$ at runtime: the model takes an evolving KG $\mathcal{K}_{run}$ and the sets of changes as input. The selected features are calculated and fed into the already trained prediction model. For CHIMP to work, it requires an domain or task expert's guidance with regard to the (1) definition of impact. The (2) definition of features, (3) choice of feature selection procedure, and (4) selection of prediction algorithms are steps that in future could also be automated. For now, these steps require the knowledge of data mining and machine learning.

To gain insights towards answering *RQ-II*, predictions from CHIMP over the different operations and KGs have to be analyzed in detail. By looking at the interaction between the impact and selected features, similarities between task-specific and KG-specific predictions models can be identified and used. Ideally, I would find some features that are used across the models and show the same relationship towards the different task-specific impact measures. Therefore, the main step in answering the second research question is the thourough analysis of the feature selection inside each implementation of CHIMP.

## 6 Evaluation of Predictors

The first part of the evaluation of the predictors built by CHIMP is a closer look at the distribution of the impact. The distribution is a vital contributor in the decision of which prediction approach should be used. For a normal distribution linear regression is appropriate. If normality can not be detected a transformation, like the log or square-root transformation, needs to be used on the impact measure. However, I will also apply non-linear regressions, like polynomial or principal component regressions that do not require a normal distribution, to widespread data. If there are distinct peaks close to 0 and 1 impact, assuming that the impact is a value in the range [0,1], I can treat the impact as a binary value and use classification. Methods like Support Vector Machine with radial kernel, K-Nearest Neighbors and Random Forest can be applied in this case.

Before evaluating the prediction models, I take preventive measures against over-fitting. I split the data into a training and testing datasets, where the training dataset contains 70% of the data points. Additionally, for learning, I use 10-fold cross validation. This is especially beneficial for small datasets. Testing of the learned models is then done on the remaining 30% of the dataset. These are decisions made within *Selection of Prediction Algorithm* in CHIMP.

I compare the performance of the different models and approaches using the area under the receiver operating characteristics curve (AUC). To calculate the AUC for regression, I determine a threshold for the observed impact. I then treat the predicted values as probabilities and calculate the AUC as if it was a classification task. The goal is to only signal the impact (value of 1) when a recalculation of the task is necessary. The detailed comparison of models is part of the last step of CHIMP's middle section.

## 7 Preliminary Results: First Experiments

I began my research by selecting two tasks, materialization and embeddings. Materialization is the calculation of a finite set of additional axioms that can be inferred from the ones that are already present. An embedding is the representation of the entire graph in a vector space of multiple dimensions. The tasks are differ significantly. The former creates a graph following a deterministic process, while the latter produces vectors according to a stochastic process. It follows that comparison is not directly possible. Subsequently, I describe potential impact measures for the two tasks; next, I describe two experiments I performed on them, considering two different datasets. Finally, I describe the analysis of the features I extracted for the scenarios.

**Impact for Materialization.** Since the result of a materialization is a finite graph, I can define the impact in terms of a graph distance measure. Dehmer, Emmert-Streib, and Shi [2] propose the usage of the topological indexes as an input for the graph similarity distance $D$:

$$impact_{mat}(K_t, K_{t+1}) = D(K_t, K_{t+1}) = 1 - e^{-(\frac{I(K_t)-I(K_{t+1})}{\sigma})^2} \tag{1}$$

where $I(K_t)$ and $I(K_{t+1})$ are the zeroth order Randić indices of $K_t$ and $K_{t+1}$, and $\sigma$ is a parameter set by the user. The zeroth order Randić index ${}^0R(\cdot)$ is a topological index and is defined as:

$$I(K) = {}^0R(K) = \sum_{u \in V(K)} \frac{1}{\sqrt{d(u)}}, \tag{2}$$

where $u$ is a node from the set of nodes $V(K)$ in the graph $K$ and $d(u)$ is the degree of $u$. This measure ranges from 0 to 1. I have also considered the Randić and Wiener impact, which are also proposed in [2]. However, my preliminary experiments did not lead to good results and I have omitted them.

**Impact for Embeddings.** With a stochastic process, like embeddings, defining impact becomes more complex. To evaluate embeddings of two snapshots, I

perform the comparison using neighborhoods. Taking the 100 closest neighbors of a particular node, I 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 mean:

$$impact_{emb-avg}(K_t, K_{t+1}) = \frac{\sum_{u \in V(K)} N(u_t) \cap N(u_{t+1})}{|V(K)|} \tag{3}$$

where $u_t$ is the node $u$ in $K_t$ and $u_{t+1}$ is the same node in $K_{t+1}$. $N_t(u)$ is the neighborhood of the node $u$ in the embedding of $K_t$. $|V(K)|$ is the number of nodes in $K$. The measure is normalized to the range $[0, 1]$.

I investigated the overall loss of embedding and link prediction performance. In addition, there are common measures to compare embeddings, but they are usually used to compare different embedding approaches over the same KG. I identified the extension of such approaches to compare embeddings created by the same techniques on two snapshots of the same KG as future work.

**Experiments.** I considered three datasets: Bear-B-instant (BB) [5], the Gene Ontology (GO) [20] and an anonymized and de-identified WebProtege Ontology (WP). Other ontologies and KGs also need to be considered to be able to confirm preliminary results and draw general conclusions. The criterion for data selection are the following: The ontology or KG (1) has at least 2'000 snapshots, (2) was edited by more than two users, and (3) has less than 40k nodes and 80k edges. This number of snapshots is a trade of between the time it would be necessary to compute impact and features and the considered time frame of the evolution of the KGs. The limitation on the size of the KG is due to calculation time of impact and features and also to have KGs that are comparable in size. The initial list of features was comprised using measures from social network analysis that show the structure of the KG. Spareness and entropy were added to include features outside of social networks.

I applied two feature selection procedures: Pearson correlation and ridge regression. Significance was the indicator for a feature to be used in prediction. For materialization, classification was used because of the impact distribution of BB and WP datasets showing peaks at 0 and close to 1. I classified the impact larger than 0.5 as 1 and smaller than 0.5 as 0. The choice of the threshold is based on the meaning of impact. When doing classification, the model decides if there will be impact (1) or if there won't be any impact (0). Including numbers below 0.5 would give the same attention to changes with lower impact compared to changes with very high impact. Therefore, the cutoff was decided at 0.5. However, this number is for now arbitrary and further investigation is necessary to determine if lower or higher thresholds would be suited better.

SVM with a radial kernel, k-Nearest Neighbors, and Random Forest were the three prediction models built using the features sets selected by correlation and ridge regression. With all three algorithms and two feature sets an AUC value of over 0.97 was achieved. For the impact on embeddings, regression was necessary. For the BB dataset, performance does not exceed an AUC of 0.64. On the other hand for WP, prediction models shows a performance of AUC = 0.85.

Table 1: Selected features for the different prediction models for the two operations (*mat* and *emb*). (+,−) indicate the (positive, negative) direction of the coefficient using Pearson correlation (*corr*) and ridge regression (*ridge*). (.) are not selected features.

| | **Corr-** BB | | WP | | **Ridge-** BB | | WP | |
|---|---|---|---|---|---|---|---|---|
| Features | mat | emb | mat | emb | mat | emb | mat | emb |
| number of nodes | . | − | . | . | + | − | − | . |
| number of edges | + | + | . | . | + | . | . | . |
| mean degree | . | . | − | . | − | + | − | . |
| mean degree centrality | − | − | . | . | + | − | − | − |
| mean closeness | − | − | . | . | + | + | − | + |
| mean degree connectivity | . | . | − | . | − | − | + | . |
| assortativity | . | . | − | . | − | − | − | − |
| cluster coefficient approx | . | . | . | . | + | − | − | − |
| cluster coefficient | − | . | − | − | − | + | + | . |
| transitivity | − | . | + | − | + | . | − | − |
| number of components | . | . | + | . | . | . | − | + |
| strong components | . | − | . | . | + | − | − | . |
| mean SP length | . | . | . | . | + | . | − | + |
| longest SP length | . | . | . | . | . | . | + | + |
| centrality entropy | + | . | . | . | − | + | + | . |
| closeness entropy | + | . | . | . | + | + | + | − |
| sparseness | − | . | . | . | + | − | − | − |

The prediction of the impact for embeddings performed significantly worse than for materialization. The stochastic nature of embedding computation results in less predictable outcomes compared to deriving a materialization. Hence, impact prediction of comparable consistency cannot be expected.

**Analyzing Selected Features.** Table 1 shows the selected features for sets that were used in the predictions. As mentioned before, the features are common social network features with the addition of entropy and spareness to also include feature outside of this domain. For all these features, the semantics inside the KG was completely ignored and the KG was turned into a directional unnamed graph. The first column shows the names of the features. For Pearson correlation (*Corr*), significance was the selection criterion and the correlation value was not considered. However, the +/- sign indicates the direction of the correlation value. For Ridge regression (*Ridge*, a coefficient determined selection as well as +/- entry in the table. The comparison between the two feature selection procedures *Corr* and *Ridge* is not advisable, because the respective approaches are very different in nature. In both selection approaches along with both datasets, features between the two operations show the same direction of influence on the impact. Therefore, it can be concluded, that the indicated features describe the impact partially in the same way. For Corr in BB, three features are in common as shown in Table 1. For Corr in WP, only the cluster coefficient is in common

between the tasks. Investigating the ridge models, three features are in common for BB and six for WP. These findings are of great importance. They show that the impact measures have common indicators, which could be used in a real world application. Focusing on fewer features allows potential advantages for implementation in e.g. impact detector inside an ontology editor.

I have also analyzed the features which are more concerned with the nodes that are directly affected by a change. However, I was only able to calculate such features for GO using COnto-Diff [11]. The algorithm inside COnto-Diff is only applicable on ontologies following the OBO standard. These features showed great influence on the impact measures in both cases, materialization and embeddings. The calculation of the change action features also takes less time than considering features that need calculation over the entire graph, because the calculation of graph features grows exponentially. It is thus of great interest to investigate change action features.

## 8 Reflections

As with the butterfly effect, a small change in a KG can lead to large differences in operation results. The goal is therefore to decide when an operation needs to be recalculated due to the KG evolution. In this research description, I propose the iterative development of a methodology, which can be used to build a predictor for the impact of changes of KGs. This is a crucial step towards understanding the magnitude of changes, making KG engineers aware of the possible impact.

So far, I have used a first version of the methodology to build various prediction models across three different datasets and two operations. The experimental results confirmed that CHIMP can be used to study the impact of changes. I will continue by revising this first methodology and by applying it to further operations, different impact measures, new features and prediction algorithms.

The biggest challenge so far has been the definition of an appropriate impact measure given an operation. Looking into practical use cases will help to distinguish the respective understanding of impact. At the same time, obtaining suitable datasets has proven difficult. There are many KGs of different contexts and sizes, each of them recording and reporting the evolution history in a different way. This requires an adaptation of KGs and their evolution before usage.

The recent advances by related studies have shown that the desired research is indeed feasible. In addition, my current results are promising and provide preliminary answers to both research questions. In future, I want to refine the introduced methodology and broaden the research to encompass additional KGs, operations and features.

**Acknowledgments.** I want to recognize the help of my supervisor Prof. Abraham Bernstein, PhD and Dr. Daniele Dell'Aglio of the University of Zurich.

## References

1. 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)

2. Dehmer, M., Emmert-Streib, F., Shi, Y.: Interrelations of Graph Distance Measures Based on Topological Indices. PLOS ONE **9**(4), e94985 (Apr 2014)
3. Ekanayake, J., Tappolet, J., Gall, H.C., Bernstein, A.: Tracking concept drift of software projects using defect prediction quality. In: MSR. pp. 51–60. IEEE Computer Society (2009)
4. Fernandez, J.D., Schneider, P., Umbrich, J.: The DBpedia wayback machine. In: SEMANTICS. pp. 192–195. ACM (2015)
5. Fernandez, J.D., Umbrich, J., Polleres, A., Knuth, M.: Evaluating Query and Storage Strategies for RDF Archives. In: SEMANTICS. pp. 41–48. ACM (2016)
6. Gonçalves, R.S., Parsia, B., Sattler, U.: Analysing the evolution of the NCI Thesaurus. In: 2011 24th International Symposium on Computer-Based Medical Systems (CBMS). pp. 1–6 (Jun 2011)
7. Goncalves, R.S., Parsia, B., Sattler, U.: Categorising logical differences between OWL ontologies. In: CIKM. pp. 1541–1546. ACM (2011)
8. Gottron, T., Gottron, C.: Perplexity of Index Models over Evolving Linked Data. In: ESWC. vol. 8465, pp. 161–175. Springer (2014)
9. Gross, A., Hartung, M., Prüfer, K., Kelso, J., Rahm, E.: Impact of ontology evolution on functional analyses. Bioinformatics **28**(20), 2671–2677 (2012)
10. Hartung, M., Gross, A., Rahm, E.: CODEX: Exploration of semantic changes between ontology versions. Bioinformatics **28**(6), 895–896 (Mar 2012)
11. Hartung, M., Gross, A., Rahm, E.: COnto-Diff: Generation of complex evolution mappings for life science ontologies. JBI **46**(1), 15–32 (Feb 2013)
12. Hartung, M., Terwilliger, J.F., Rahm, E.: Recent Advances in Schema and Ontology Evolution. In: Schema Matching and Mapping, pp. 149–190. Springer (2011)
13. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Q. **28**(1), 75–105 (2004)
14. Klein, M., Noy, N.F.: A component-based framework for ontology evolution. In: Workshop on Ontologies and Distributed Systems at IJCAI. vol. 3, p. 4 (2003)
15. Noy, N.F., Klein, M.: Ontology Evolution: Not the Same as Schema Evolution. Know. Inf. Sys. **6**(4), 428–440 (Jul 2004)
16. Osborne, F., Motta, E.: Pragmatic Ontology Evolution: Reconciling User Requirements and Application Performance. In: ISWC (1). vol. 11136, pp. 495–512. Springer (2018)
17. Rashid, M., Torchiano, M., Rizzo, G., Mihindukulasooriya, N., Corcho, Ó.: A quality assessment approach for evolving knowledge bases. SemWeb **10**(2), 349–383 (2019)
18. Ren, Y., Pan, J.Z.: Optimising ontology stream reasoning with truth maintenance system. In: CIKM. pp. 831–836. ACM (2011)
19. Stavropoulos, T.G., Andreadis, S., Kontopoulos, E., Kompatsiaris, I.: SemaDrift: A hybrid method and visual tools to measure semantic drift in ontologies. Journal of Web Semantics (Jun 2018)
20. The Gene Ontology Consortium: Gene Ontology Consortium: Going forward. Nucleic Acids Research **43**(D1), D1049–D1056 (Jan 2015)
21. Trivedi, R., Dai, H., Wang, Y., Song, L.: Know-Evolve: Deep Temporal Reasoning for Dynamic Knowledge Graphs. In: ICML. vol. 70, pp. 3462–3471. PMLR (2017)
22. Tury, M., Bielikova, M.: An approach to detection ontology changes. In: ICWE Workshops. vol. 155, p. 14. ACM (2006)
23. Zablith, F., Antoniou, G., d'Aquin, M., Flouris, G., Kondylakis, H., Motta, E., Plexousakis, D., Sabou, M.: Ontology evolution: A process-centric survey. Knowl. Eng Rev. **30**(1), 45–75 (2015)