Supplementary material for SEKA: Seeking Knowledge Graph Anomalies

Asara Senaratne, Pouya Ghiasnezhad Omran, Peter Christen, Graham Williams

School of Computing, The Australian National University, Canberra ACT 2600 Australia.
asara.senaratne@anu.edu.au, p.g.omran@anu.edu.au, peter.christen@anu.edu.au, graham.williams@anu.edu.au

Our Approach Visualized

Following recent work in anomaly detection in Knowledge Graphs (KGs) (Senaratne et al. 2021; Zheng et al. 2019; Jia et al. 2018), our aim is to discover abnormal triples and entities that provide contradicting, semantically incorrect, and redundant information, or that contain invalid, incomplete, or missing information. Figure 1 provides a visualization of SEKA.

Knowledge Graph Definition

We consider a directed edge-labelled KG, $G = (V, E)$ containing a set of nodes (or vertices) V and a set of labelled edges E connecting these vertices. The Resource Description Framework (RDF) is a standardised data model based on directed edge-labelled graphs[1]. The RDF model defines three types of nodes in a graph: (1) Internationalized Resource Identifiers (IRIs) I which assigns a global identifier for entities I_e and relations I_r on the web (where $I = I_e \cup I_r$); (2) literals L which represents strings and other datatype values; and (3) blank nodes B which are anonymous nodes (not a URI reference or a literal) that do not have an identifier (Hogan et al. 2021). We therefore have the node set $V = (I_e \cup L \cup B)$ and edge set $E \in V \times I_r \times V$. Each edge $t \in E \in G$ is considered as a RDF triple. A triple (also named as a triplet or fact) contains the three elements subject $s \in S$, predicate $p \in P$, and object $o \in O$. A triple is denoted as (s, p, o) where $(s, o) \in V$, and $(s \times p \times o) \in E$. Furthermore, $s \in (I, B)$, $p \in I_r$, and $o \in (I, L, B)$.

Generate Synthetic Anomalies

As the four KGs we use for experimental evaluation do not contain labelled data, we manually inject anomalies (corrupt existing triples to form anomalies). To achieve this, we introduce INK (INject anomalies to Knowledge graphs), which injects the following types of anomalies to a KG.

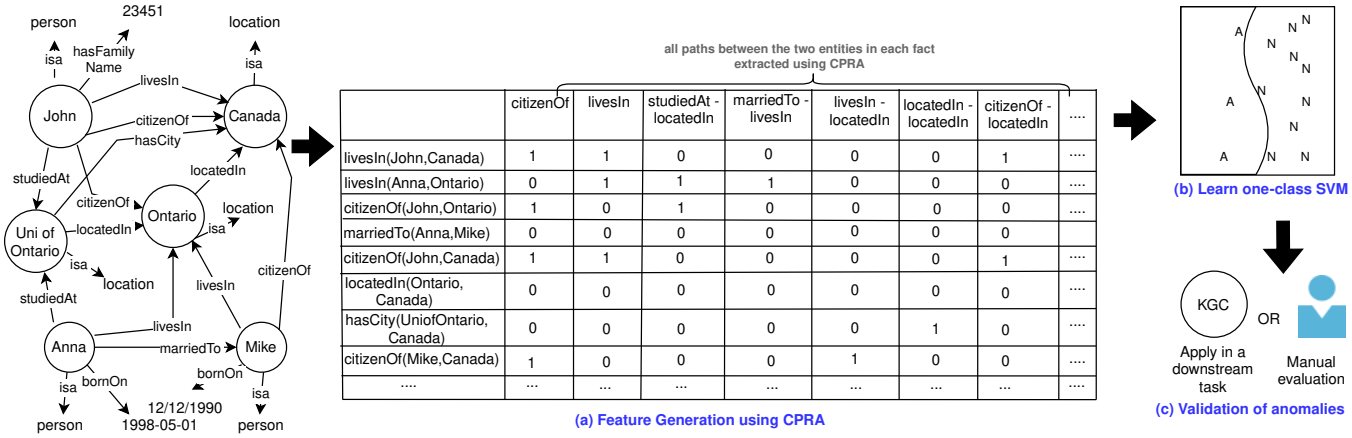
- Change the subject of a triple while preserving the entity type of the original subject. For example, (personA, isMarriedTo, personB) gets replaced by (personC, isMarriedTo, personB).
- Change the object of a triple while preserving the entity type of the original object. For example, (personA, isMarriedTo, personB) gets replaced by (personA, isMarriedTo, personD).
- Change both the subject and object of a triple while preserving the entity type of the original subject and object. For example, (personA, isMarriedTo, personB) gets replaced by (personE, isMarriedTo, personF).
- Change the subject of a triple while also changing its entity type. For example, (personA, isMarriedTo, personB) gets replaced by (TheMoon, isMarriedTo, personB).
- Change the object of a triple while also changing its entity type. For example, (personA, isMarriedTo, personB) gets replaced by (personA, isMarriedTo, London).
- Change both the subject and object of a triple while also changing their entity type. For example, (personA, isMarriedTo, personB) gets replaced by (TheMoon, isMarriedTo, London).
- Change the predicate of a triple while also changing its semantic usage. For example, replace predicates used for a person with a predicate that is not used for a person. For example, (personA, isMarriedTo, personB) gets replaced by (personA, livesIn, personB).

To replace the content of one triple, the new content is extracted from the KG itself. For example, assuming the triple (personA, isMarriedTo, personB) gets replaced by (personE, isMarriedTo, personF), the entities personA, personB, personE, and personD are all existing entities within the KG under consideration. The entities and predicates to be used as replacements get selected randomly by INK after analysing the type information associated with those entites. INK ensures that there are no duplicate triples generated.

We developed INK independent of SEKA[2], so we can use it to generate anomalies for any KG irrespective of type and size. At the run time, INK asks the user to specify the number of anomalies he/she wishes to have in the KG. For example, specifying 10% will corrupt 10% of the triples in the KG. INK randomly selects the triples for corruption. INK completes execution by displaying both the original and corrupted versions of the triples.

[1] https://www.w3.org/TR/rdf-concepts/\#section-Graph-URIref

[2] https//github.com/AsaraSenaratne/SEKA



	citizenOf	livesIn	studiedAt - locatedIn	marriedTo - livesIn	livesIn - locatedIn	locatedIn - locatedIn	citizenOf - locatedIn	....
livesIn(John,Canada)	1	1	0	0	0	0	1	....
livesIn(Anna,Ontario)	0	1	1	1	0	0	0	....
citizenOf(John,Ontario)	1	0	1	0	0	0	0	....
marriedTo(Anna,Mike)	0	0	0	0	0	0	0	
citizenOf(John,Canada)	1	1	0	0	0	0	1	....
locatedIn(Ontario, Canada)	0	0	0	0	0	0	0	....
hasCity(UniofOntario, Canada)	0	0	0	0	0	1	0	....
citizenOf(Mike,Canada)	1	0	0	0	1	0	0	....
....	...	...	...	...	...	...	...	....

Figure 1: Overview of SEKA, our anomaly detection process to identify anomalous triples in a KG.

Dataset Statistics

Table 1 provides a summary of the sizes of the four real-world KGs that we used to conduct the experiments described in the main abstract. These KGs demonstrate the applicability of our approach on KGs from different domains of varying sizes.

KG	Entity count	Triple count where the object is a literal	Triple count where the object is an entity
YAGO-1	2,215,094	21,337,521	922,741
KBpedia	62,796	534,032	227,060
Wikidata	14,036,475	53,541,372	51,559,889
DSKG	5,952	22,202	828,086

Table 1: KG summaries.

YAGO[3] contains general knowledge about people, cities, countries, movies, and organizations. This version of YAGO includes the data extracted from the categories and infoboxes of Wikipedia, combined with the taxonomy of WordNet.

KBpedia[4] is an open source KG that combines Wikipedia, Wikidata, schema.org, DBpedia, GeoNames, OpenCyc, and standard UNSPSC products and services into an integrated and computable structure. KBpedia has 98% coverage of Wikidata and nearly complete coverage of Wikipedia.

Wikidata[5] is a free, collaborative, multilingual, secondary database, collecting structured data to provide support for Wikipedia, Wikimedia Commons, other wikis of the Wikimedia movement, and to anyone in the world. It is an open knowledge base that can be read and edited by both humans and machines.

The Data Set Knowledge Graph (DSKG)[6] is a RDF dataset that contains information about datasets and the publications that use those datasets. The metadata of the datasets are modeled in the standard vocabulary data catalog (DCAT) and are based on datasets registered in OpenAIRE and Wikidata.

[3]https://yago-knowledge.org/downloads/yago-1

[4]https://kbpedia.org/

[5]https://www.wikidata.org/wiki/Wikidata:Main_Page

[6]http://dskg.org/

Parameter Settings

We now discuss the parameter settings used in SEKA. To eliminate the need of having user involvement during the anomaly detection process, our approach is designed in such a way that it has a minimal number of parameters that need to be manually set.

In the step of constructing features, to determine the path length to traverse in depth-first search, we use n where it is by default set as $n = 0.5, 1, 2$. Hence, no user involvement is required to determine the value of n.

In the step of identifying anomalies, the ν-SVM requires configurations for the two parameters, its kernel and ν. In order to obtain robust results, we train four ν-SVMs with the four kernels RBF, poly, linear, and sigmoid. This ensures our classification outcomes are not biased due to the use of a single kernel function. Hence, the user does not have to configure the kernel for training. The ratio of normal to abnormal classified feature vectors can be specified using the ν parameter by a user, according to the budget of how many abnormal feature vectors she/he can investigate further. By default, we set ν to 10%, which a user can increase or decrease based on the capacity for manual evaluation.

Additional Results

We used the two measures, precision and recall, to evaluate SEKA in performing fact anomaly detection on the four experimental datasets. We conducted the evaluation under three different percentages of anomalies injected into the KGs.

As the results in Table 2 show, our approach performs well in identifying anomalies in all four KGs under all three anomaly percentages. SEKA performs the best in DSKG with 10% and 20% anomalies, while it performs the best in YAGO-1 with 30% anomalies.

KG	10% anomalies			20% anomalies			30% anomalies		
	Anomalies count	Precision	Recall	Anomalies count	Precision	Recall	Anomalies count	Precision	Recall
YAGO-1	92,274	0.99	0.99	184,548	0.90	0.89	276,822	0.86	0.87
KBpedia	22,706	0.82	0.81	45,412	0.70	0.70	68,118	0.72	0.72
Wikidata	5,155,989	0.81	0.80	10,311,978	0.79	0.78	15,467,967	0.72	0.71
DSKG	82,809	0.99	0.99	165,617	0.95	0.94	248,425	0.82	0.81

Table 2: Results obtained by SEKA for fact anomaly detection with 10%, 20% and 30% anomalies in each KG.

Manual Evaluation of Anomalies

From the output of the ν-SVM, we can extract the abnormal triples or entities and validate them manually with insights from a domain expert. This manual validation can aid in correcting the identified anomalies to improve KG quality. Furthermore, we can remove these anomalous triples from the KG and subject the refined KG for a downstream task such as Knowledge Graph Completion (KGC) to observe performance improvement.

References

Hogan, A.; Blomqvist, E.; Cochez, M.; d'Amato, C.; Melo, G. D.; Gutierrez, C.; Kirrane, S.; Gayo, J. E. L.; Navigli, R.; Neumaier, S.; et al. 2021. Knowledge graphs. *ACM Computing Surveys (CSUR)*, 54(4): 1–37.

Jia, B.; Dong, C.; Chen, Z.; Chang, K.-C.; Sullivan, N.; and Chen, G. 2018. Pattern Discovery and Anomaly Detection via Knowledge Graph. In *International Conference on Information Fusion*, 2392–2399. IEEE.

Senaratne, A.; Omran, P. G.; Williams, G.; and Christen, P. 2021. Unsupervised Anomaly Detection in Knowledge Graphs. In *International Joint Conference on Knowledge Graphs*, 161–165. New York, USA: ACM.

Zheng, L.; Li, Z.; Li, J.; Li, Z.; and Gao, J. 2019. AddGraph: Anomaly Detection in Dynamic Graph Using Attention-based Temporal GCN. In *IJCAI*, 4419–4425. USA: IJCAI.