Knowledge Graphs in Supply Chains

Supervisor: Dr. Cogan Mathew Shimizu

Presenter: Kandula Rakesh

Contents

The Construction of a Domain Knowledge Graph and Its Application in Supply Chain Risk Analysis

Construction of Domain Knowledge Graph

Knowledge graph-based information systems can be of essential use in the times of big data. The mechanisms to be taken into consideration when constructing such information systems include:

i. Knowledge representation and reasoning, which touches areas of language processing and demands the extraction of schema graph with the assistance of standard vocabularies
ii. Knowledge storage, which usually includes graph databases and repositories, and raises the problem of D2R matching.
iii. Knowledge engineering, which can be based on patterns or other mining techniques.
iv. (Automatic) Knowledge learning including schema learning and population. Ontology instance linking, ontology alignment and disambiguation needs to be considered in this phase

Community Detection

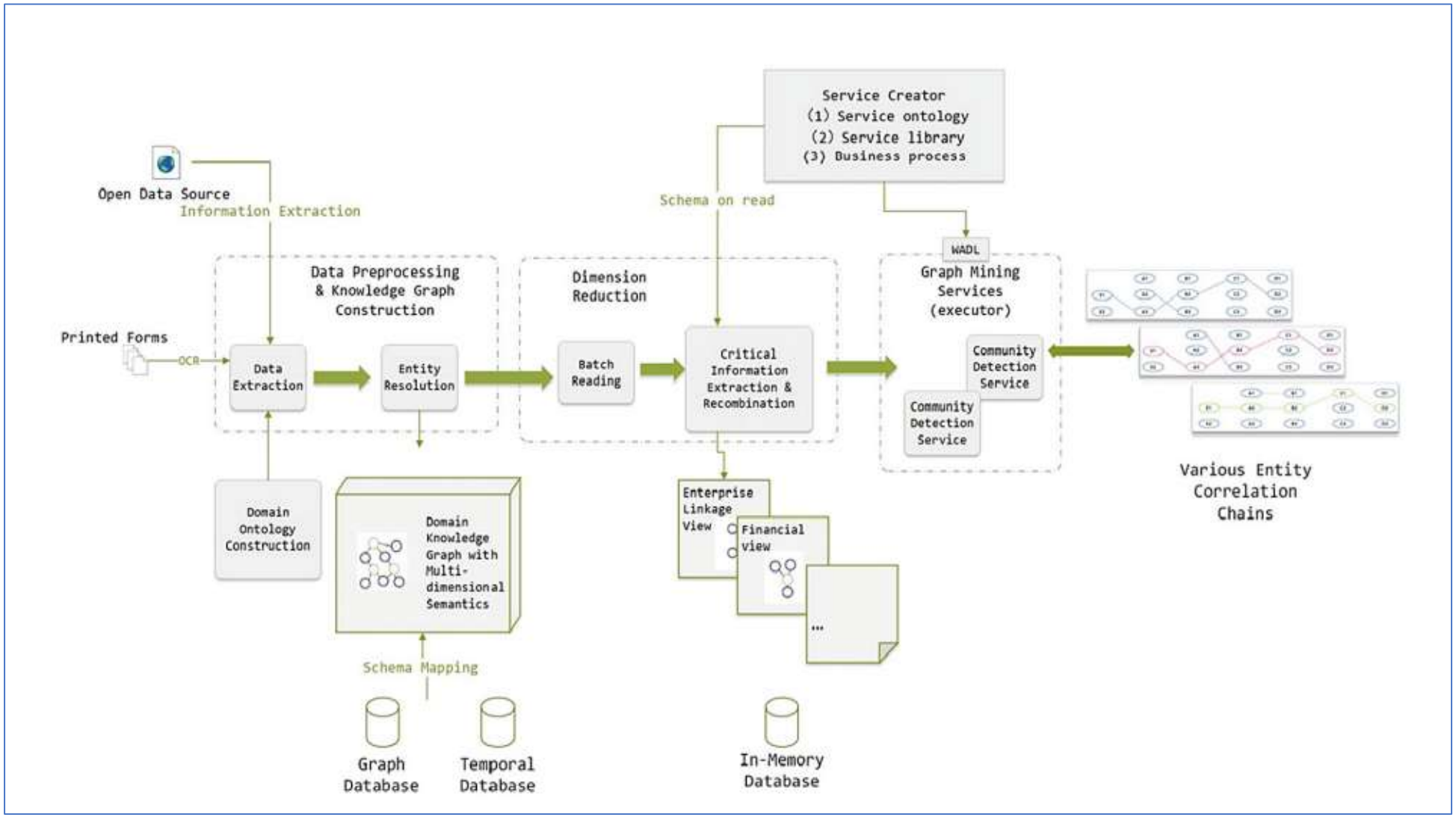
- The process of discovering cohesive groups or clusters in networks is known as community detection. It is one of the key tasks of social network analysis. A community can be defined as a set of entities related more closely with each other in some perspective than those outside the group. The closeness between entities of a group can be measured via similarity or other distance measures
- The community detection can be done by using the Label Propagation Algorithm. This works by assigning the unique labels to all the nodes in the graph and label of highest frequency of occurrence in all its adjacent nodes are assigned to it.
- However, this algorithm cannot solve problems in which one individual belongs to multiple communities. In other words, communities cannot overlap with each other.

Community Detection

(contd..)

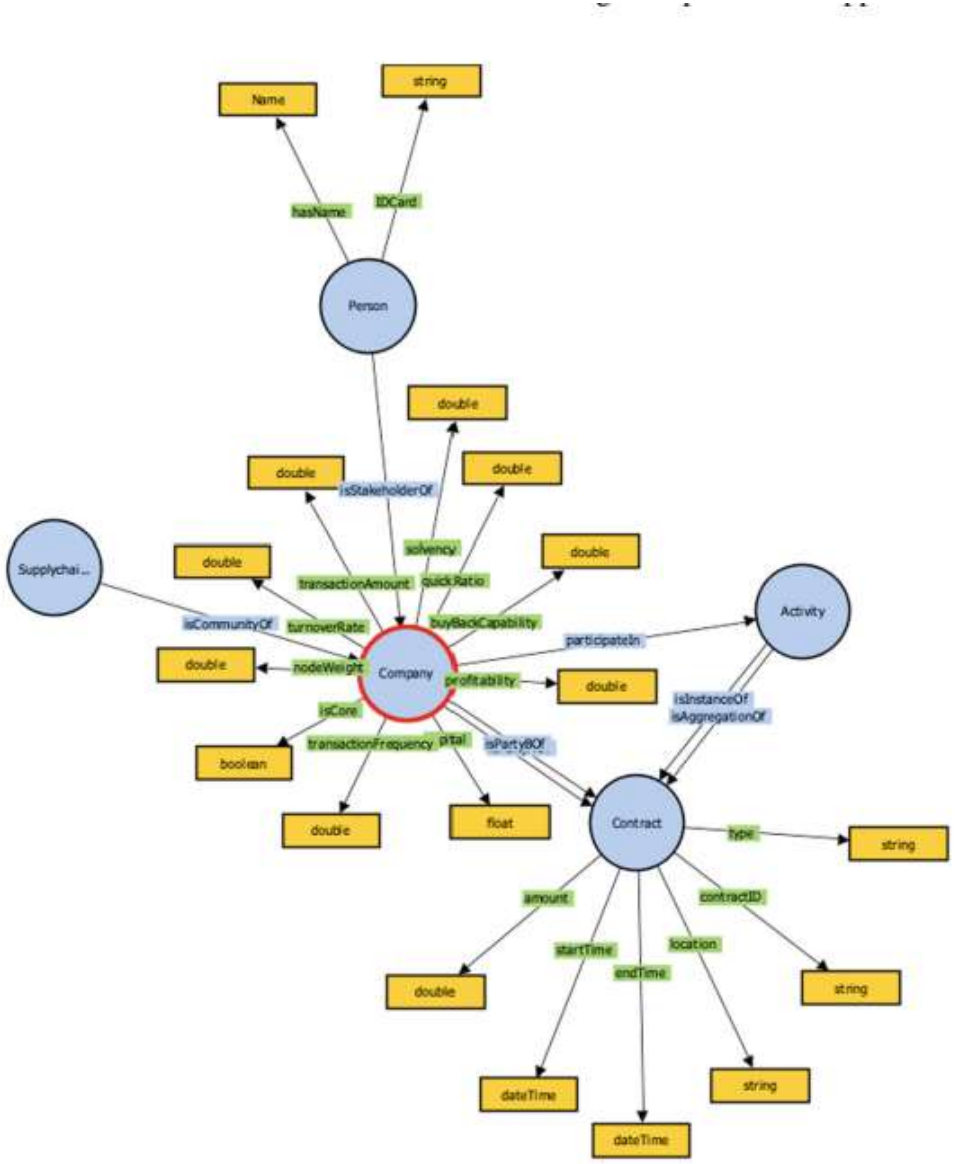
- SLPA addressees the issue in LPA, it contains the storage of a label set for each node. Although this is not directly used in the research paper, based on this there are two methodologies COPRA and BMLPA which refine the initialization and updating process of node tags.
- First, There exists a Rough Core(RC) Function is initialized to form several non-overlapping communities and each node is given a belonging coefficient and the initial value is set to one.
- Then the community labels are propagated and updated iteratively. When updating the community label of a central node, the belonging coefficients of nodes neighboring to central node are summed up and normalized by community. The labels with relatively small coefficients are removed. This process end until the network converges.
- Finally, if there are any small groups that are included in larger groups, those properly included groups are removed.

System Framework



Schema

- Knowledge Graph schema.



A Knowledge Graph Perspective on Supply Chain Resilience

A Knowledge Graph Perspective on Supply Chain Resilience

- In case of disasters to predict supply chain system only tier 1 info is available most of the time, not tier n , making it hard to predict. Information regarding suppliers is in different location, types and formats. It needs to improve how to identify the risks involved in predicting the supply chain management.
- In this paper, we aim at increasing supply chain resilience, based on data from Siemens, by addressing the challenges mentioned above in the following ways:
  i. Collect different data and create a Knowledge Graph upto tier 3
  ii. We apply state-of-the-art knowledge graph completion methods for link prediction in the knowledge graph to predict missing information.
  iii. We use graph analysis algorithms to identify critical entities in the supply network, where we focus on centrality measures to derive an importance score for each supplier

Knowledge Graph Dataset

- Supply chain knowledge graph is constructed from :
  i. siemens internal sources: to reflect internal knowledge as tier-1 suppliers, business scopes, and Siemens parts
  ii. external sources: to reflect external knowledge such as public data about smelters and substances.
- The information about tier-2 and tier-3 suppliers of Siemens is obtained mainly from public customs data, and a small part is obtained from private customs data and public media.
- We define a knowledge graph (KG) as a collection of triples $G \subset E \times R \times E$,where E denotes the set of entities, R is the set of relation types.
- Elements in E correspond to supply chain-related entities, e. g., suppliers, smelters, and components, and are represented as nodes in the graph. Every entity has a unique entity type, which is defined by the mapping $t : E \rightarrow T$, where T stands for the set of entity types.
- The entities are connected via relation types specified in R, represented as directed edges in the graph. All entity and relation types and corresponding numbers of nodes and edges are listed in the schema.

Knowledge Graph Schema

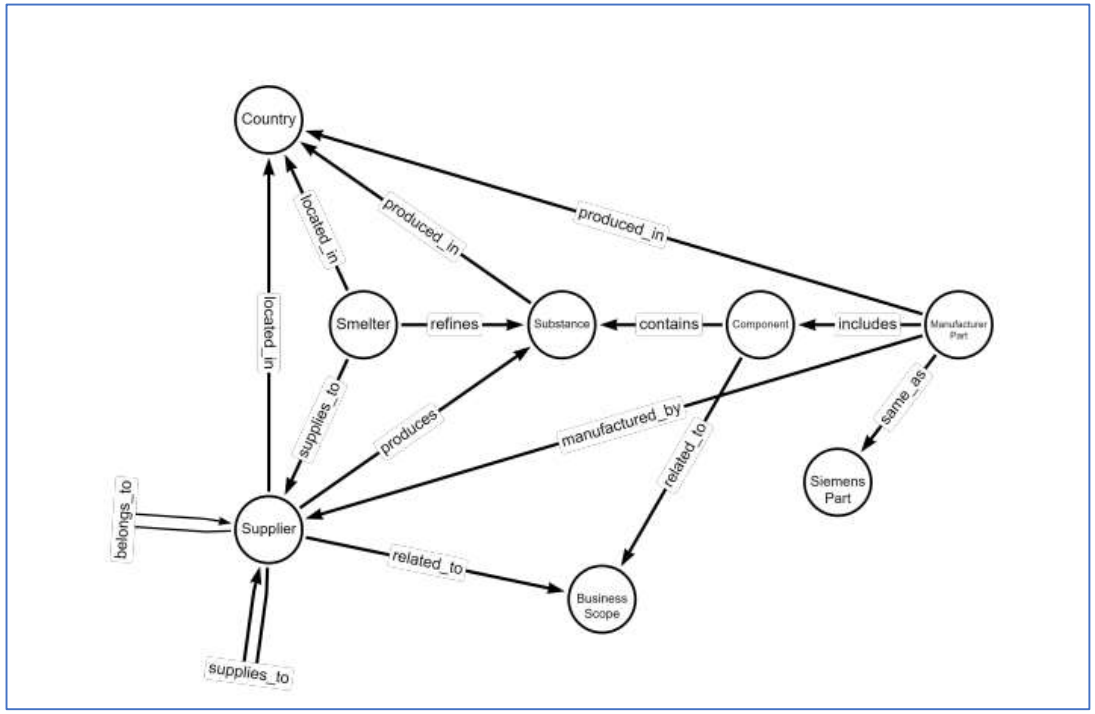


Fig: Knowledge graph schema. There are 8 entity types and 11 relation types.

Entity type	Nodes	Relation type	Edges
Supplier	61,234	supplies_to	138,197
Manufacturer Part	1,650	related_to	59,894
Siemens Part	1,295	belongs_to	56,663
Smelter	340	located_in	30,107
Substance	321	includes	10,088
Component	233	produces	7,831
Country	172	produced_in	4,381
Business Scope	32	same_as	1,847
		manufactured_by	1,564
		contains	764
		refines	340
Total	65,277	Total	311,676

Fig: Entity and relation type statistics. In the graph, there are 8 entity types, where most nodes are suppliers, and 11 relation types, where most edges are from the type supplies_to

Object Prediction Task

- Many KGs suffer from incompleteness, So the common task in graph machine learning is KG completion or link prediction. In this paper we formulate the link prediction task as an object prediction task. The triples in the KGs are in the form of (s, p, o). In object prediction task, given query in the form of (s, p, ?), the goal is to predict a ranked list of entity candidates that are more likely the correct object of the query.
- To measure the quality of the predictions, the mean reciprocal rank (MRR) and hits@k for $k \in \mathrm{N}$ are standard metrics used for link prediction on KGs. For a rank $x \in \mathrm{N}$, i. e., the position in the ranked list of entity candidates, the reciprocal rank is defined as $1/x$, and the MRR is the average of all reciprocal ranks of the correct query objects over all queries. The metric hits@k represents the proportion of queries for which the correct object appears under the top k candidates.

Knowledge graph completion methods

- There are several methods for KG completion. In this paper, we focus on graph representation learning. Based on these embeddings, a score can be calculated for each entity, indicating its likelihood to be the correct object of a query.
- We apply the following traditional and state-of-the-art methods to the supply chain KG:
  i. RESCAL
  ii. ComplEx
  iii. TuckER
  iv. TransE
  v. RotatE
  vi. ConvE
  vii. RGCN
  viii. CompGCN

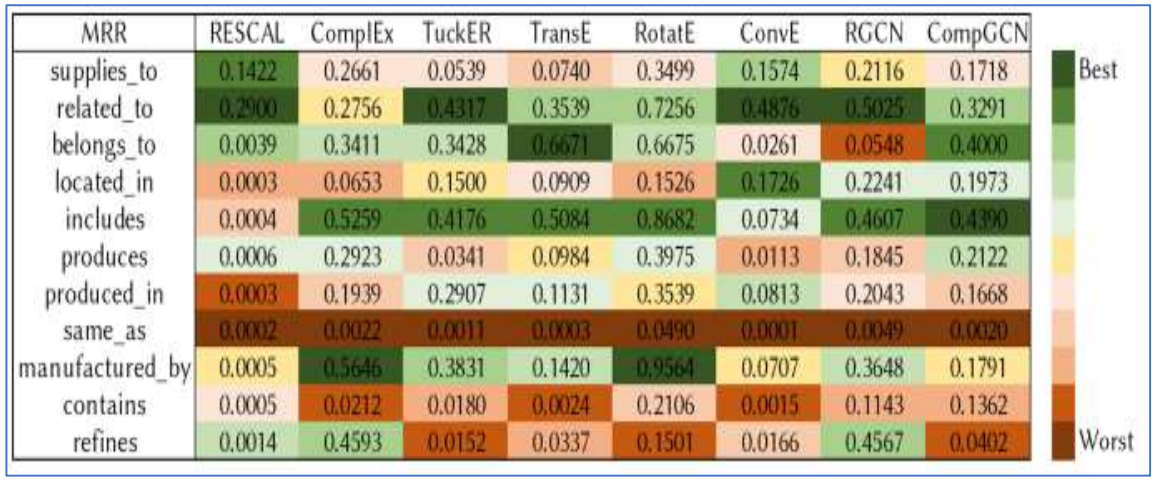
Experimental Setup

- For all KG completion methods, we use the implementations provided by the Python library PyKEEN.
- We split the graph into training, validation, and test dataset, where we operate under the transductive setting, i. e., all entities and relation types from the validation and test set are also included in the training set.
- The training set consists of 65,277 nodes and 249,340 triples, while both the validation and test set have 31,168 triples. The number of nodes for the validation and test set is 22,212 and 22,213, respectively
- All three datasets include all entity and relation types. We use the optimizer Adam and optimize the margin ranking loss with a margin of 1, where one negative triple is sampled for each training triple.
- We tune the hyperparameters embedding size in the range {16, 32, 64, 256, 512, 1024} and learning rate in the range {0.0001, 0.001, 0.01}
- The training of the model is stopped early if there is no improvement regarding the metric hits@10 on three subsequent evaluations on the validation set, where the evaluation takes place every 10 epochs.

Results

Method	MRR	Hits@1	Hits@3	Hits@10
RESCAL	0.1476	0.0684	0.1809	0.2772
ComplEx	0.2535	0.1793	0.2850	0.3949
TuckER	0.1738	0.0749	0.1878	0.4033
TransE	0.1595	0.0873	0.1733	0.3164
RotatE	**0.4377**	**0.3686**	**0.4733**	**0.5627**
ConvE	0.2289	0.1549	0.2438	0.3875
RGCN	0.2911	0.1784	0.3379	0.5195
CompGCN	0.2223	0.1271	0.2486	0.4229

Fig: Results on the test dataset for the object prediction task. The best results are displayed in bold.



MRR	RESCAL	ComplEx	TuckER	TransE	RotatE	ConvE	RGCN	CompGCN
supplies_to	0.1422	0.2661	0.0539	0.0740	0.3499	0.1574	0.2116	0.1718
related_to	0.2900	0.2756	0.4317	0.3539	0.7256	0.4876	0.5025	0.3291
belongs_to	0.0039	0.3411	0.3428	0.6671	0.6675	0.0261	0.0548	0.4000
located_in	0.0003	0.0653	0.1500	0.0909	0.1526	0.1726	0.2241	0.1973
includes	0.0004	0.5259	0.4176	0.5084	0.8682	0.0734	0.4607	0.4390
produces	0.0006	0.2923	0.0341	0.0984	0.3975	0.0113	0.1845	0.2122
produced_in	0.0003	0.1939	0.2907	0.1131	0.3539	0.0813	0.2043	0.1668
same_as	0.0002	0.0022	0.0011	0.0003	0.0490	0.0001	0.0049	0.0020
manufactured_by	0.0005	0.5646	0.3831	0.1420	0.9564	0.0707	0.3648	0.1791
contains	0.0005	0.0212	0.0180	0.0024	0.2106	0.0015	0.1143	0.1362
refines	0.0014	0.4593	0.0152	0.0337	0.1501	0.0166	0.4567	0.0402

Fig: Results of the best models for each relation type. For each model, the performance with respect to the MRR is colored from best (green) to worst (red).

Uses in cybersecurity knowledge graphs

- From the paper domain knowledge graph construction, we can use the data processing framework for preprocessing the data and extracting the information from the printed contracts and PDF's by using the OCR.
- We can also apply the community detection mechanism to the KG, to detect the communities and divide the network into groups. We can use this topic on the cybersecurity knowledge graph, to apply the community detection mechanism to the CKG and group it into based on the Vendor, manufacturer or any other entity.
- From the other paper we can use the schema of the knowledge graph and by using the same techniques as Object prediction task or by applying the state-of-the-art algorithms to complete the knowledge graph.


thank
you