Demonstration of KGNet: a Cognitive Knowledge Graph Platform

Hussein Abdallah, TheKien Nguyen, Duc Nguyen, and Essam Mansour

Concordia University, Canada
{fname.lname}@concordia.ca

Abstract. Knowledge graph (KG) engines lack the support for cognitive queries based on semantic affinity and classification models. Having the power of semantic search on KG engines will enable users to quickly generate deep hidden insights from their KG and enrich it. Variant vectorized representations (embeddings) techniques are proposed to encode the semantics of a word, image, graph node, and edge. User Defined Functions (UDFs) could be used to calculate semantic affinity between two entities by measuring the distance between their embeddings. We will demonstrate KGNet; a system that supports cognitive queries by transparently optimizing the queries, estimating UDF cost, automatically selecting the suitable embedding technique, executing the optimized query, and finally explaining the semantic results. During the demo, the audience will experience KGNet using four different use cases based on real datasets and variant embeddings techniques in real applications. KGNet is a step forward to enable advanced AI capabilities in KG engines. A demo video is available online here[1]

1 Introduction

Knowledge graphs (KGs) are adopted across variant application domains to integrate heterogeneous datasets via semantic information extraction from csv files, text, images, and videos. RDF engines are widely utilized to store KGs due to RDF's simplicity, powerful query language (SPARQL), and inferencing support on top of RDF Schema and Web Ontology Language. Thus, numerous applications create RDF-based KGs and enable access via online service (endpoint) receiving SPARQL queries via HTTP requests. RDF engines support SPARQL queries that apply logical, arithmetical, and set operators, such as union and join. For example, a query finds dogs of a certain weight, or a list of companies located in a given location. Cognitive queries are a new class of queries that use AI technologies to extract relevant information based on semantic similarity and classification models. There is a lack of support for cognitive queries based on semantic affinity and classification models, e,g., a query retrieving information about dogs semantically matching a given dog's image, or a list of companies whose financial growth is similar to a particular company.

[1] https://rebrand.ly/KGNET01

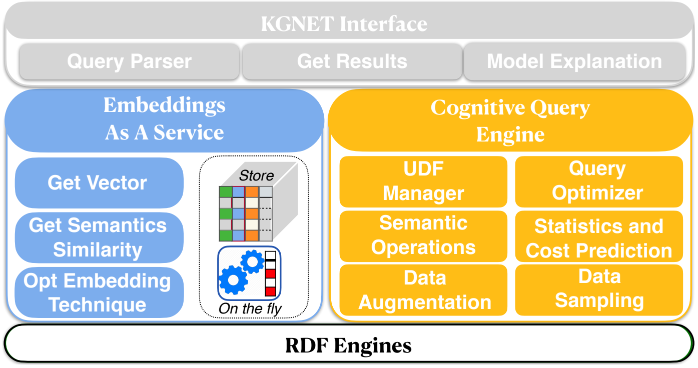

Fig. 1: The KGNet architecture.

Numerous embedding techniques can encode the semantics of heterogeneous datasets ranging from text and images to graph nodes and edges. Glove [3] is a word embedding technique capturing semantic similarity of words. VGG16 [5] is a deep CNN image classification model to generate image embeddings. FaceNet [4] obtains vector representations for face images. Moreover, HolE,TransE and others [6] are KG embeddings techniques that can embed graph nodes/edges.

The semantic similarity between two entities could be measured by calculating the distance between the entities' embeddings using distance metrics, such as Euclidean distance, Cosine, and Jaccard. The accuracy of these embedding techniques and distance metrics varies from a dataset to another. Moreover, space and time complexity of generating the embeddings vary, too. Thus, it is a time-consuming task for technical users to build AI pipelines that extract semantic information from KGs based on embedding similarity and classification models.

We can utilize User Defined Functions (UDFs) supported in RDF engines to extend a SPARQL query with semantic affinity operators and classification models. However, the query engine should address several research challenges, such as estimating UDF cost, automatically selecting the suitable embedding technique, and finally explaining the semantic results. The cognitive queries in relational databases were introduced by [1] which focused on learning representations for database tokens, i.e., words. Cognitive KG queries are more challenging due to the large varieties of embedding techniques and heterogeneous datasets.

In this demo, we purpose KGNet; a cognitive KG platform that supports numerous embedding techniques, different semantic similarity measures, and data augmentation for training classification models. KGNet addresses the above research challenges to allow efficient semantic exploration of KGs with extensible AI capabilities. Section 2 outlines the KGNet architecture. Section 3 gives a glimpse on demo scenario. Section 4 concludes.

2 The KGNet Platform

KGNet is modularized into three layers as illustrated in Figure 1. The interface layer enables users to post their cognitive queries through our GUI for easy exploration of their KG or through our web service endpoint for easy integration into data science pipelines via Jupyter Notebooks and Google Colab. The

KGNet interface layer is integrated with SHAP [2] tool to provide visual explanation of results based on classification models. The middle layer is composed of two main components, namely **Embedding As A Service** and **Cognitive Query Engine**. **Embedding As A Service** acts as an embedding store that maintains linked-data or graph embeddings and builds a catalogue of embedding techniques. It provides three main services: *get embedding vector* for an entity by choosing an embedding model from a catalogue or through providing custom trained embedding model, *get similarity score* between two entities based on a pre-defined similarity measure metrics and finally *get near-optimal embedding techniques* by choosing between different embedding techniques for the same data set [6]. Hence, selecting the near-optimal model for a cognitive query is important by sampling query results that satisfy the query conditions.

Cognitive Query Engine is responsible for providing semantic operators functionality to SPARQL queries and augmenting graph data. In data augmentation, we enrich graph nodes with several features to generate more accurate embeddings. KGNet maintains a catalogue of pre-defined UDFs covering primitive semantic similarity measures for different applications. KGNet makes it easy to write a cognitive query using the UDFs, semi-automates the execution pipeline, and collects statistics to support query optimizations.

KGNet automates the selection of the near-optimal embedding technique using our data sampling. The KGNet underlying layer is an RDF engine. In KGNet, the RDF engine should support UDFs and communicate with external endpoints through HTTP get/post requests. KGNet currently tested with Virtuoso and the Apache Jena and supports UDFs in PL/C++ or Java, respectively. KGNet design provides modular components that interact together to support scalability. Integration with KGNet is pretty simple; users have to provide the KG data, use a pre-defined (embeddings service, UDF).

3 Demonstration Overview

We developed four different use cases based on real datasets and variant embeddings techniques. Due to lack of space, this section highlights three use cases as summarized in Table 1. The first use case, **Dogs Breeds** demos semantic search in KG with image contents. We collected real datasets from Kaggel and Data World for dog breeds classification. Moreover, we augmented the generated KG with features, such as breed overview, intelligence level, breed health issues, and recommended for, collected from bowwowinsurance.com. The constructed and augmented KG is serialized as a Turtle file and loaded into a Virtuoso server. KGNet used a pre-trained VGG16 [5] deep learning (DL) model to classify these breeds and generate embeddings then link them with KG through a UDF called **getDogSimilarityScore** (see Figure 2 line 3) that returns similarity score between two images. VGG16 pre-processes the input image to generate its embeddings on the fly, i.e., the used UDF consumes more time. Hence, a simple cashing mechanism for external URLs embedding vector in memory is used to improve cognitive query response time. Building a cost-effective estimation

Table 1: Use cases Using Real Datasets.

	KG-Image	**Economics**	**QA**
Dataset	Dog Breeds	Forbes-2013	DBPedia-2016-10
Data Source	Kaggel + Data.World	Mannheim Data and Web Science Group	DBPedia
Data Augmentation	Scraping data from bowwowinsurance.com	Categorical Features and Wikidata URIs	NA
UDF	getDogSimilarityScore	getSimilarCompanies	getSimilarKeywords
Embed. Type	Image Embedding	KG Embedding	KG Embedding
Embed. Models	VGG16	RDF2Vec ,TransE	GloVe
Classification	VGG16	Random Forest	NA

```
prefix ns1:<https://www.dog_breeds.com/>
SELECT ?dog_image ?breed_class ?breed_overview
(sql:getDogSimilarityScore(?dog_image,?external_image_url)) as ?Score
WHERE {?s  ns1:img_folder_name ?breed_class. ?s  ns1:img1 ?dog_image.
       optional {?s  ns1:breed_overview ?breed_overview} }
ORDER BY DESC(xsd:float(?Score))
```

Fig. 2: Cognitive SPARQL query that select dogs who looks-like external image.

model for these UDFs is mandatory to optimize user queries and predict the shortest execution pipeline.

The second use case, **Economics KG**, demos semantic search based on the KG structure and the interconnection among KG nodes. In this use case, we use structural KG embedding techniques to group similar companies in the Forbes-2013 dataset together based on companies attributes and neighbourhood location in the graph. We used off-the-shelf KG embedding techniques, such as *RDF2Vec*, *HolE*, *TransE*, *DistMul* [6], to generate embeddings, and evaluated the accuracy using Random Forest ML models. This dataset contains a list of attributes like company name and market value. To augment this graph, we converted attributes, such as market value, profit, and rank, into three categorical classes (low, medium and high) based on quantile values of these features.

The final use case is related to **KG Question-Answering**. In this use case, we use text embedding techniques to get a similar keywords list to a certain predicate and use this list to improve QA results. We used DBPedia KG to query for a subject based on predicate similar embedded words retrieved from GloVe [3] pre-trained model and linked with KG subject.

It is a challenging task to select a near-optimal embedding technique for a specific dataset. It is tedious for expert users to decide which embedding technique to use, especially with different dependant dataset attributes. Figure 3 shows the accuracy of using three different KG-embedding techniques to predict the market value attribute using a random forest classifier. DistMul embedding technique achieved the best prediction accuracy with a score of 0.978. KGNet finds a near-optimal embedding technique for a specific cognitive query by sampling graph data and applying ML prediction task on it. KGNet uses this mechanism

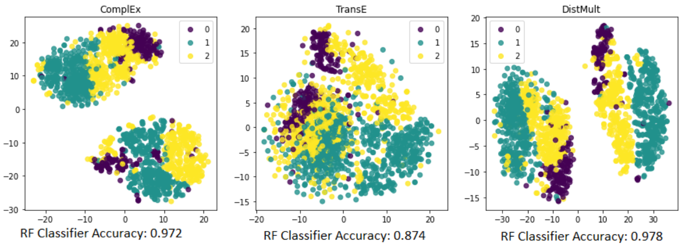

Fig. 3: Economics KG-embeddings clustered by Market-value attribute [low-mid-high] and predicted using Random Forest Classifier.

to decide the best embedding technique on the fly and generate the results with the highest accuracy.

4 Conclusion

KGNet bridges the gap between KG engines and AI pipelines to provide built-in cognitive query support. Thus, there is no need to reformat and migrate KG data from RDF engines to data science or AI platforms. Our proposed cognitive query extension enables the invocation of a user-defined semantic function based on different embedding techniques. KGNet optimizes the query execution pipeline, estimates UDF's cost, and automatically opts for the near-optimal embedding techniques. In KGNet, a user needs only to provide data and customize UDF for semantic search or use an existing one. This enables semantic discover on KGs for users without prior knowledge on embedding techniques and AI pipelines.

References

1. Bordawekar, R., Shmueli, O.: Enabling cognitive intelligence queries in relational databases using low-dimensional word embeddings (2016)
2. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 4765–4774. Curran Associates, Inc. (2017)
3. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
4. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
5. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
6. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering **29**(12), 2724–2743 (2017). https://doi.org/10.1109/TKDE.2017.2754499