IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Visualize Knowledge Graphs through semantic maps

P. CAMARILLO-RAMIREZ[1], L. F. GUTIÉRREZ-PRECIADO[1], AND F. CERVANTES-ALVAREZ.[1]
[1]Western Institute of Technology and Higher Education, Tlaquepaque, Jalisco 45601 Mexico
(pablo.camarillo@iteso.mx;lgutierrez@iteso.mx;fcervantes@iteso.mx)

Corresponding author: P. Camarillo-Ramirez (e-mail: pablo.camarillo@iteso.mx)

This work was supported in part by the National Council of Science and Technology of Mexico through grant 498322

ABSTRACT Knowledge Graphs (KGs) are one of the most novel technologies used to improve search engines and support decision making in the life sciences since they structure information in graph form by encoding concepts as nodes, and the semantics of the relationship among concepts as edges. The analysis of KGs calls for an effective strategy to visualize them. However, the increasing size of KGs makes the exploring process a big challenge. A semantic map is a visual representation of related concepts that helps humans in the learning process. In this work, we propose to generate a simplified visual representation of a KGs by generating semantic maps. We apply several clustering algorithms to group the *related* concepts in KGs. We used different semantic similarity metrics to compute the matrix consumed by the clustering algorithms.

INDEX TERMS Knowledge graphs; Knowledge graphs visualization; Semantic similarity; Semantic mapping; Big Data;

I. INTRODUCTION

Knowledge Graphs are considered one of the emerging technologies associated with Big Data by providing semantic structured information that can be interpreted by machines, and such attribute is used to speed up the production of more intelligent systems [**?**]. The core idea behind a Knowledge Graph (KG) is to represent knowledge from real world in a graph structure, where nodes represent entities of interest and edges represent relations between these entities [**?**]. Recently, academic and private organizations have constructed KGs, such as YAGO [**?**], DBPedia [**?**], Freebase [**?**], NELL [**?**], Google Knowledge Graph [**?**], Microsoft Satori [**?**], Facebook Entity Graph [**?**], and Wikidata [**?**], which contain millions of entities and billions of relationships. The main applications of KGs include the enhancement of search engines like Google [**?**] or Bing [**?**], question answering [**?**], information retrieval, recommender systems [**?**], [**?**], domain-specific KG building [**?**], [**?**], [**?**], and decision support in the life sciences [**?**], [**?**], [**?**], [**?**].

Considering the continuous increase in the use of KGs in decision-making applications, it becomes important to provide explanations for the results generated by the graph. Visualizing KGs can help to unlock insights from data and facilitate better decision making, actually visual data exploration is considered as a hypothesis-generator process by allowing users to gain a deep understanding of the data [**?**], hence producing an effective visual representation of a KG is crucial to understand relationships between entities and concepts in a domain. By representing the information in a visual format, users can quickly identify patterns, trends, and clusters or related information that may be difficult to see in a text-based representation. Existing approaches to visualize KGs are focussed on drawing the whole structure [**?**] preventing data analysts to explore the KG beyond its structural information.

Semantic maps, on the other hand, are a technique widely used to understand complex topics and consists of a categorical structuring of information in a graphic form [**?**]. Semantic maps have a central word indicating the main topic of the map and it is connected with a set of keywords that groups the rest of the vocabulary.In this paper, we hypothesize that semantics maps are useful to visualize the high level of abstraction of a KG based on the semantic closeness of entities in the KG. To generate a semantic map it is necessary to find the groups of related instances. Unsupervised learning provides clustering algorithms to classify data into one or more classes depending on a similarity or distance measure [**?**]. Theoretically, if a clustering strategy is applied over the set of entity instances

of a KG, it will group those entity resulting groups can be used to build the semantic map. Section V presents a set of experiments validating the above notion.

The main contribution of this work is to provide a formal definition of the semantic map of a KG as well as a strategy to measure the quality of these semantic maps by running a set of experiments showing that these semantic maps can be useful to provide a high-level view of a KG.

The rest of this paper is structured as follows. Section II, we introduce KGS. In Section III, we review the most relevant literature associated with knowledge graph visualization, graph clustering, semantic similarly, and semantic mapping topics. Section IV, describes the proposed method and algorithms needed to generate the semantic maps of a KG. Section V discusses the results obtained from a set of experiments evaluating the process of generating semantic maps generated from a collection of datasets extracted from DBPedia. Finally, Section VI presents final remarks and future work.

II. FOUNDATIONS OF KNOWLEDGE GRAPHS

Some definitions describe a KG as a graph-structured knowledge base [**?**], [**?**]. In this work, we consider a knowledge base a set of sentences/facts expressed in some formal language such as description logic. In other words, KGs consist in a collection of facts formed by $< subject, predicate, object >$. These collections are typically represented in languages, such as RDF (Resource Description Framework) [**?**], OWL (Ontology Web Language) [**?**], or N-Triples which is a subset of the more complex RDF/XML syntax, and is designed to be human-readable as well as machine-readable. It is a plain text format that represents RDF statements using subject-predicate-object triples, with each element separated by whitespace and terminated by a period.

According to description logic terminology, knowledge bases have two types of axioms: a terminology box (TBox) and an assertion box (ABox) [**?**], hence a KG should contain these two sets of axioms to be considered as a knowledge base. To exemplify the above idea, Figure 1 shows sets TBox and ABox of a group of entities and relationships extracted from DBPedia [**?**] [1]. In KGs, the ontology classes (e.g.,`dbo:Book` [2] or `dbo:Movie`) correspond with the TBox and describe concepts hierarchies, while the ontology instances correspond with the ABox and describe entity instances (e.g.,`dbr:Lucasfilm` or `dbr:George_RR_Martin`) and their relationships. Hierarchical relationships like *is a* defines the connection between each pair of concepts in TBox. For example, axioms (`dbo:Book`, *is a*, `dbo:Work`) and (`dbo:film`, *is a*, `dbo:Work`) describe the fact that both `dbo:Book` and `dbo:film` concepts are descendants of the class `dbo:Work`. Alternatively, in ABox, axioms also indicate the list of types that one entity instance may have. For instance, in Figure 1, the axiom (`dbr:A_dance_of_dragons`, *rdf:type*, `dbo:Book`) indicates that resource `dbr:A_dance_of_dragons` is an instance of class `dbo:Book`. Another type of axioms in ABox like (`dbr:George_RR_Martin`, is *dbo:creator_of* of, `dbr:DaenerysTargaryen`) and (`dbr:George_RR_Martin`, is *dbo:author* of, `dbr:A_dance_of_dragons`) indicate that the instance `dbr:George_RR_Martin` has two semantic connections with `dbr:DaenerysTargaryen` and `dbr:A_dance_of_dragons` entity instances.

Let us propose a formal definition of KG before describing the rest of relevant topics associated with the semantic mapping process described in this paper.

Definition 1 (Knowledge Graph): Given a set of entities V, a set of property labels L, and a set of edges E a Knowledge Graph K is defined as $K = (V, L, E)$ where E is a subset of the cross product of entities and property labels defined as $V \times L \times V$. Each member of E is referred to as a triple $(subject - property - value)$.

III. RELATED WORK

This section describes the most relevant concepts associated to the proposed semantic mapping process.

A. VISUAL DATA EXPLORATION OF KNOWLEDGE GRAPHS

The idea behind the visual data exploration process is to present the data in some visual form, allowing users to draw conclusions of the analyzed phenomena [**?**]. This process, also known as the *information seeking mantra*, follows three steps: overview, zoom and filter, and details-on-demand [**?**]. In this context, ontologies are considered one of the most relevant data visualization techniques. In the field of computer science, an ontology is a model for describing the world that consists of a set of types, properties, and relationship types [**?**] and by providing an initial attempt to visualize linked data.

In regard to visual exploration of KGs, challenges include context adaptation, users input [**?**], data heterogeneity [**?**], [**?**], [**?**], supporting diverse analysis tasks (query, combination, filtering, etc.), and performance [**?**]. In this study, semantic mapping proposal is to combine and reduce the number of edges in the KG using the semantic similarity among its entities to compute clusters of related entities.

Recent applications have proven useful for large graph visualizations to understand different phenomena, such as Bitcoin transactions [**?**] and online discussions [**?**]. For big knowledge graphs, it is necessary a distributed implementation of the layout algorithms to improve the time needed to generate the visual representation [**?**]. Actually, Consalvi et. al [**?**] propose a self-contained system to compute interactive visualizations of thousands elements in a mobile browser.

In addition of recent efforts on KGs visualization, there are some commercial products enabling analysts to visual-

[1] https://dbpedia.org

[2] URIs mentioned in this document use the common prefixes described in https://prefix.cc

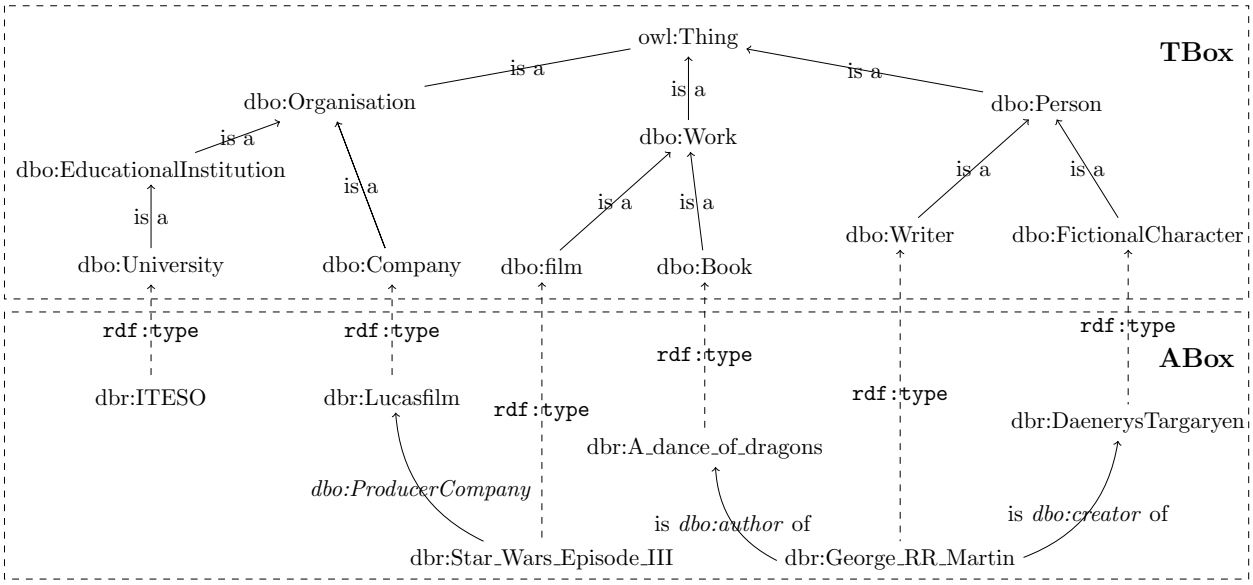

FIGURE 1: Small group of concepts and instances extracted from DBPedia.

ize RDF graphs like Data Graphs[3] or the family of tools developed by Cambridge Intelligence company: Keylines, ReGraph, and KronoGraph [4] that offer the capability to render KGs to support tasks in areas like pharmacy and bioscience research or financial analysis. In the area of free tools, there are two online tools that consumes RDF data and produce a visual representation: RDF visualizer [5] and RDF grapher [6]. The main limitation with these tools is the small amount of data they can process.

B. KNOWLEDGE GRAPH SUMMARIZATION

Recent works [**?**], [**?**], [**?**] have shown that visualizing a simplified version of a large graph is an adequate alternative. In the context of data mining, summarization is the process of facilitating the identification of meaningful data. The applications of graph summarization include reduction of data volume and storage, speedup of graph algorithms and queries, interactive analysis support, and noise elimination [**?**]. Recently, it has been proposed to summarize large graphs in order to enable an efficient visualization of their content. For example, in [**?**], the authors focus on summarizing KGs by taking advantage of individual interests to generate personalized knowledge graph summaries. In [**?**], Shen et al. propose a visual analytics tool called OntoVis, which performs both structural and semantic abstractions to offer a summarized version of a large graph and thus being able to visualize a simplified version of the graph. Another related work is presented in [**?**], which describes the VoG (Vocabulary-based summarization of Graphs) algorithm to summarize and understand large graphs by constructing and visualizing subgraph-types, such as starts, cliques, and chains. The visual abstraction presented in [**?**], transforms geo-tagged social media data into high-dimensional vectors by utilizing a doc2vec model.

Regardless of the application, one of the main challenges of graph summarization is defining which data is of interest. Every summarization strategy depends on selecting an interest criteria to extract meaningful information [**?**]. However, to achieve a concise definition of *interesting* is not an easy task. For example, the FUSE algorithm [**?**] proposes a profit maximization model that seeks to find a summary by maximizing information profit under a budget constraint. On the other hand, VoG [**?**] exploits the Minimum Description Length (MDL) principle aimed at identifying the best subgraphs by choosing those which save most bits. In the case of semantic abstraction proposed in [**?**], a dual-objective blue noise sampling model is utilized to select a subset of social media data items supporting the spatial distribution and semantic correlation for the resulting simplified geographical visualization. The personalized summaries of KGs described in [**?**], the criteria to decide which information is *interesting* for each user is determined by reviewing the users' query history. The work of M. Tasnmin et al. propose a strategy to find equivalent entities in a KG using the context of each RDF Molecule [**?**]. Semantic mapping process described in this document uses the semantic similarity between each pair of entity instances in the KG to infer the groups of related instances.

C. SEMANTIC MAPS

A semantic map is a type of graphical representation that shows the relationships between different concepts or words

[3]https://datagraphs.com
[4]https://cambridge-intelligence.com/
[5]https://issemantic.net/rdf-visualizer
[6]https://www.ldf.fi/service/rdf-grapher

within a particular domain or field of study. The purpose of a semantic map is to visually organize and display the meaning and connections between various terms or concepts, highlighting their semantic similarities and differences. In other words, a semantic map provides a visual representation of how different ideas or concepts are related to each other and how they are grouped together based on their shared meanings or semantic properties. Figure 2 shows an example of a semantic map describing the topic *Water*. It contains three node categories: (1) the central words (root), (2) the set of keywords (e.g., Usages, Living things, etc.), and (3) the vocabulary associated to each keyword, for instance, words `Cooking` and `Bathing` are associated with keyword *Usages*.

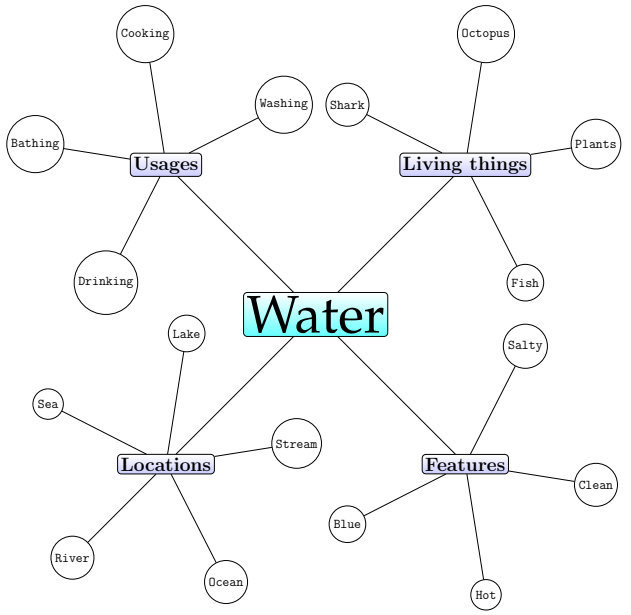

FIGURE 2: Example of a semantic map of concepts and vocabulary associated with topics *Water*.

D. SEMANTIC SIMILARITY

The semantic similarity is a metric used in Natural Processing Language (NPL) and Information Retrieval (IR) areas [**?**] that represents how related are two concepts based on their hierarchical relations [**?**], [**?**]. In a KG, the semantic similarity between two entities $e_1, e_2 \in V$ is denoted as $sim(e_1, e_2)$. Intuitively, semantic distance between two words is the most easy way to calculate semantic similarity and it is usually determined by the path connecting two entities in KG. Existing semantic similarities metrics are classified in two main groups: corpus-based and knowledge-based approaches [**?**]. Corpus-based similarity metrics are focused on learning how similar are two concepts based on the information from large corpora. Two examples of corpus-based similarity metrics are pointwise mutual information [**?**], and latent semantic analysis [**?**]. In contrast, knowledge-based similarity metrics quantify the degree to which two words are semantically related [**?**]. In KG, knowledge-based approaches, semantic similarity is determined using the information provided by the TBox. Knowledge-based approaches include path-based metrics such as those proposed by Hulpus et at. [**?**], Wu & Palmer [**?**], and Leacock & Chodorow [**?**]. Other knowledge-based measures utilize the Information Content (IC) metric like Lin [**?**], Jiang and Conrath [**?**], and Resnik [**?**] metrics. IC of concepts is a statistical measure that computes the specificity of a concept over a corpus. Higher values of IC indicate more specific concepts (e.g., `dbo:Book`) and lower values of IC are associated with more general concepts (e.g., `owl:Thing`). Hybrid knowledge-based approaches like IC-graph [**?**] or Zhou [**?**] combine IC and some other metrics to compute how related two words are. For instance, graph-based IC [**?**] uses the counting services provided by DBPedia and it is calculated using the following expression:

$$IC_{graph}(c_i) = -logProb(c_i) \tag{1}$$

Where $Prob(c_i) = \frac{freq_{graph}(c_i)}{N}$ and N is the number of entities in the KG. Let $\mathcal{E}(c_i)$ the set of entities having type of c_i, the frequency of concept c_i in the KG is defined as $freq_{graph}(c_i) = |\mathcal{E}(c_i)|$.

E. CENTROID-BASED CLUSTERING

There exist several techniques to clustering data and recent surveys summarize these clustering approaches based on the application or the type of data to group [**?**], [**?**], [**?**]. Types of clustering include Centroid-based, Density-based, Distribution-based, and Hierarchical clustering [**?**].

One of the phases of the semantic mapping process is to collocate each entity into the most appropiate cluster based on its semantic similarity. Each resulting cluster needs a node that represents all entities contained on it. We denote the set of this representing nodes as the keywords of the semantic map. Considering these keywords as centroids of clusters, the usage of a centroid-based clustering is crucial.

The main idea of centroid-based clustering is to find *k* centroids (or centers) followed by computing *k* sets of data points that minimize the proximity with each center. For instance, K-means algorithm tries to minimize the sim of the squared distance between the data points and the cluster's centroid [**?**]. A variation of K-means is the PAM (Partitioning Around Medoids) algorithm that minimizes dissimilarities between points in a cluster and the centroids [**?**]. The CLARA (Clustering Large Applications) algorithm is an extension of PAM for large datasets [**?**]. On the other hand, CLARANS (Clustering Large Applications based on RANdomized Search) is a partitioning algorithm focused on spatial data mining because it recognizes patters and relationships existing in spatial data such as topological data [**?**]. One last centroid-based clustering algorithm is the Affinity Propagation (AP) algorithm which consists on a message-passing procedure that looks for broadcasting messages of attractiveness and availability among data points [**?**].

F. CLUSTER QUALITY

Literature offers two classes of clustering validation measures: external clustering validation and internal clustering validation [**?**]. Internal validation metrics evaluate the quality of a clustering algorithm based on its intrinsic properties, while external validation methods evaluate the quality of a clustering solution based on its agreement with a known label of the data. Since there is no known label of the datasets used in the experiments described in this work, our proposal is to use internal validation measures such as Silhouette score, Davies-Bouldin score, and Calinski-Harabasz Index.

Each internal validation metric measure different aspects of the clusters. For example, Silhouette score measures how well each data point fits into its assigned cluster compared to other clusters [**?**]. Inertia of a cluster, also known as the within-cluster sum of squares (WSS) metric measures how tightly packed the data points are within each cluster. The goal is to minimize inertia, which is equivalent to maximizing the distances between clusters. On the other hand, Dunn index measures the distance between the nearest points in different clusters and the distance between the farthest points in each cluster [**?**]. Another known quality measure is the Davies-Bouldin index which measures the similarity between each cluster and its closest neighboring cluster, while also considering the cluster's internal similarity [**?**]. Finally, Calinski-Harabasz index measures the ratio of between-cluster variance to within-cluster variance [**?**].

IV. PROPOSED METHOD

The notion behind the semantic map of a KG is to produce a reduced version of the KG by exploiting the semantic similarity between each pair of entities. To illustrate this idea, let us generate a small KG from DBPedia containing the list of some fictional characters from series of fantasy novels by the novelist George R. R. Martin. Figure 3a) presents a visual representation produced by Gephi[7], which is one of the most used tools to visualize networked data. However, this kind of visualization is not visually super informative or visually appealing which may lead in an ineffective exploratory visual analysis. On the contrary Figure 3b) shows a semantic map of the KG by grouping the entity instances and showing the central concept of the map that connects the centroids of the cluster.

A. CLUSTERING ENTITIES OF KNOWLEDGE GRAPHS

The first phase of the semantic mapping process is to group the entities of the KG based on the semantic closeness between each pair of entities in the KG. The main challenge of this phase is to extract numeric data from the KG and generate a set of groups of entities. Algorithm 1 describes the process to compute the distance semantic matrix D.

Definition 2 (Semantic distance matrix): Given a Knowledge Graph $K = (V, L, E)$, and $sim(e_1, e_2)$ the semantic similarity between entities e_1 and e_2, the semantic similarity matrix $D(K)$ represents the semantic distance between each pair of entities in K i.e., the value for cell $d_{i,j} = 1 - sim(e_i, e_j)$.

Algorithm 1 Algorithm to build the semantic distance matrix

Input: List of triples $T = t_1, \ldots, t_n$
Output: Semantic distance matrix D

```
for i = 1 to n do
    for j = i to n do
        if i = j then
            D(i, j) = 0
        else
            D(i, j) = D(j, i) = sim(t_i, ts_j)
        end if
    end for
end for
return D
```

The relation between similarity and distance follows the notion that the higher is the similarity between two entities the lower is the distance between these entities. The $i - th$ row of $D(K)$ is the vector containing semantic distance values between the $i - th$ entity and the rest of entities in the KG. The semantic distance between each entity and itself is 0. Our proposal consist of using a centroid-based clustering algorithm and generate a non-overlapping set of clusters by using the semantic distance matrix D as input of the selected clustering algorithm. We can cluster these data points using two popular clustering algorithms: PAM and Affinity Propagation.

PAM is a clustering algorithm that works by iteratively selecting a set of k medoids from the data points and assigning each non-medoid point to its closest medoid. The algorithm tries to minimize the sum of distances between each data point and its assigned medoid. The algorithm can be formalized as shows the Algorithm 2.

Algorithm 2 Clustering entities in a KG using PAM algorithm

Input: D: Semantic distance matrix
Input: k: Number of desired clusters
Output: C: Set of clusters

```
Initialize k medoids randomly from the data points.
for all data points in x ∈ D do
    Compute the distance d(x, m_i) to each medoid m_i
    Assign x to the cluster with the closest medoid
end for
Add resulting clusters to C
return C
```

[7]https://gephi.org/

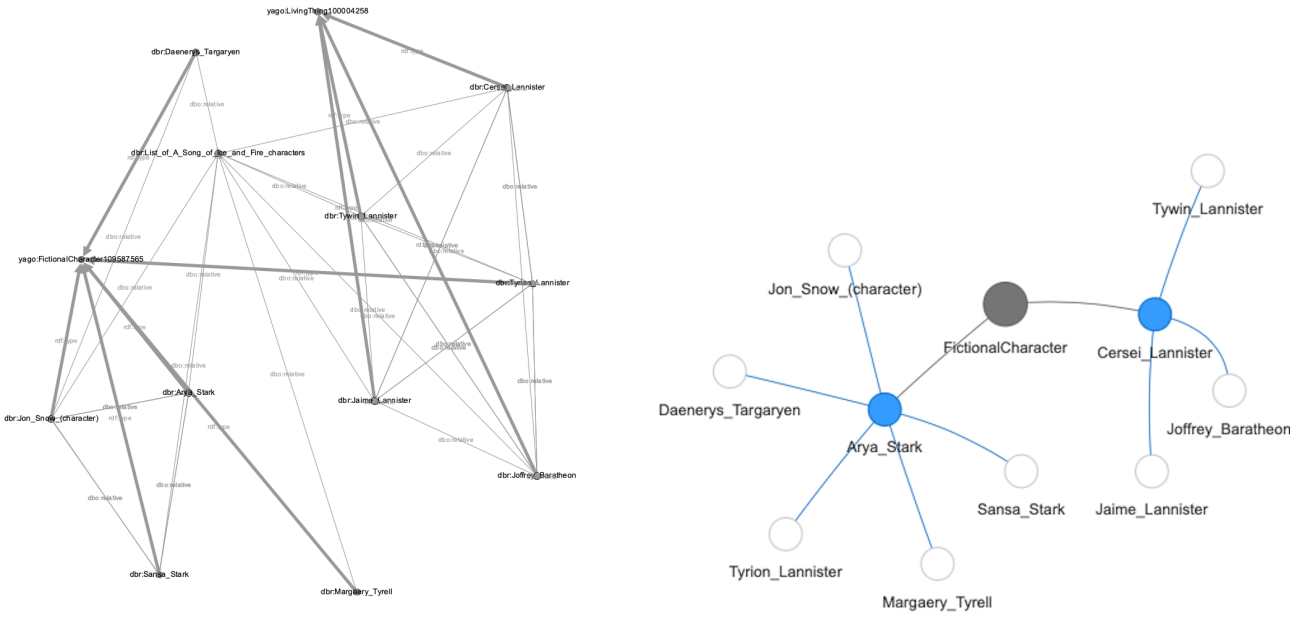

FIGURE 3: Visual representation from a small KG containing some fictional characters by George R.R. Martin. a) Contains the visual representation produced by Gephi. b) Inferred semantic map of the original KG.

On the other hand, Affinity Propagation is a clustering algorithm that works by propagating messages between data points to determine which points should be exemplars (i.e., representatives of their clusters). The algorithm is formalized in the Algorithm 3.

Algorithm 3 Clustering entities in a KG using Affinity Propagation

Input: D: Semantic distance matrix
Output: C: Set of clusters
1: **for all** data points $x \in D$ **do**
2: Assign x to the exemplar with the highest responsibility value
3: **end for**
4: Add resulting clusters to C
5: **return** C

Let $C = \bigcup C_i$ the set of clusters resulting after applying a centroid-based clustering algorithm. Each cluster C_i has a centroid element denoted by $centroid(C_i)$ and the set of centroid elements is denoted by $M = \bigcup_{C_i \in C} centroid(C_i)$.

B. CENTRAL CONCEPT OF THE SEMANTIC MAP

One of the main features of a semantic map is the **central concept** that represents the main topic of this graphical representation. In this work, we denote this central concept as α. In a regular semantic map, α is connected with a set of selected keywords (e.g., structures, characteristics, size, habitat, movie, kinds in Figure 2). These keywords are used to represent every group of words of the semantic map. This work proposes to use the centroids inferred by centroid-based clustering algorithms [**?**] as the keywords of a KG. Therefore, we denote these keywords as the set of centroids M of the entities in a KG.

To infer the central term α, we propose to compute the IC_{graph} measure for all types associated with each centroid in M. Let $types(e_i)$ to be the function to retrieves set of types associated with the entity e_i, we define $\mathcal{T}$ as the set of shared types among all centroids in μ This definition is formally described in equation 2.

$$\mathcal{T} = \bigcap_{c_i \in M} types(c_i) \tag{2}$$

Definition 3 (Central concept α): Given a set of shared types $\mathcal{T}$, the central concept α of K is the concept $c_i \in \mathcal{T}$ with maximum IC_{graph}.

Thus, the central term α represents the most specific type among all centroids in the semantic map. Algorithm 4 formalizes the process to infer the main term of the semantic map.

C. SEMANTIC MAP OF A KNOWLEDGE GRAPH

Let us define the semantic map of a KG:

Definition 4 (Semantic map of a Knowledge Graph): Given a Knowledge Graph $K = (V, L, E)$, a semantic distance matrix $D(K)$, the main term of K: α, the semantic map of K is defined as $\mathcal{SM}(K) = (\alpha, M, C, E_K)$.

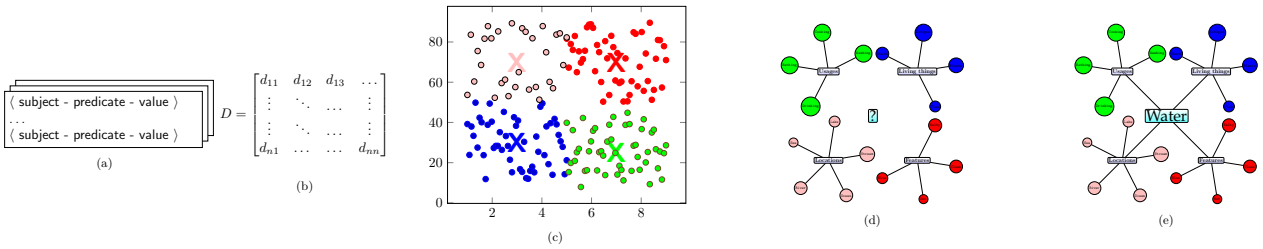

FIGURE 4: Semantic Mapping phases. (a) Consume a KG as a list of n-triples, (b) Generate the semantic distance matrix D, (c) Cluster entities using the matrix D, (d) Infer main term α, and (e) Assemble the semantic map by connecting each centroid with α.

Algorithm 4 Algorithm to infer main term α of the semantic map

Input: μ: Set of centroids of C.
Output: α: Main term of K
1: Initialize set $\mathcal{T}$ =`owl:Thing`
2: **for all** $\mu_i = centroid(C_i) \in \mu$ **do**
3: Add all elements in $types(c_i)$ to $\mathcal{T}$
4: **end for**
5: $\alpha = max(IC_{graph}(\mathcal{T}))$
6: **return** α

Table 1 describes the symbols associated with semantic maps of KGs.

TABLE 1: Symbols associated with semantic maps of Knowledge Graphs.

Symbol	Description
α	Main concept of the semantic map.
M	Set of Centroid entities, produced by a centroid-based clustering algorithm, where $M \subseteq V$.
C	Set of clusters resulting from running a centroid-based algorithm.
$\mathcal{NC}$	Non-centroids entities in clusters where $\mathcal{NC} \subseteq V$ and $\mathcal{NC} \cap \mu = \emptyset$.
E_{nc}	Set of edges connecting all members of the clusters with their corresponsing centroid, defined as $\mu \times x, \forall x \in C_i$ and $\forall \mu \in M$.
E_M	Set of edges connecting all centroids with the main term α, definded as $E_M = \mu \times \alpha$.
E_K	Set of edges connecting each all elements in the semantic map, defined as $E_K = E_M \bigcup E_{nc}$.

Semantic mapping process aggregates the process of clustering entities of KG and inferring the central term α. Algorithm 5 formalizes the process to build the semantic map of a KG. Figure 4 visually describes the phases of semantic mapping process.

Algorithm 5 Process of build a semantic map of a KG

Input: C: Set of clusters.
Input: M: Set of centroids of C.
Input: α: Main term of K
Output: $\mathcal{SM}(\mathcal{K})$
1: Initialize set $\mathcal{SM} = \emptyset$
2: **for all** cluster $C_i \in C$ **do**
3: $\mu = centroid(C_i)$
4: **for all** item $x \in C_i$ **do**
5: $add\_edge(x, \mu)$ to E_{nc}
6: **end for**
7: **end for**
8: **for all** $\mu \in M$ **do**
9: $add\_edge(\mu, \alpha)$ to E_M
10: **end for**
11: **return** $\mathcal{SM}$

V. EVALUATION STUDY

The goal of experiments described in this section is to validate the process of generating a visualization of a reduced version of a KG through a semantic map. We introduce a tool used to automate the process of generating semantic maps from different datasets. Then, we describe the datasets used to test the semantic mapping process and finally we present the cluster quality obtained for each experiment.

A. SEMANTIC MAPPING FRAMEWORK

Experiments are executed using a framework implemented using Python 3 language which depends on the *Sematch* framework [**?**] to perform SPARQL queries to DBPedia public endpoint and compute the similarity measure used to compute D, i.e., the function $sim(e_i, e_j)$ mentioned in Algorithm 1 is implemented through a SPARQL query to DBPedia. Once generated D, the tool produces the set of centroids M and the set of non-centroid $\mathcal{NC}$ nodes by using centroid-based clustering strategies: PAM and Affinity Propagation. We infer the main term α by implementing the Algorithm 4. These shared types are the result of a SPARQL query to DBPedia that follows the path shown in Figure 5. Finally, our tool, assembles the semantic map by implementing Algorithm 5.

s

TABLE 2: Dataset summary

Dataset	Description	Number of triples
SCI-FI-MOVIES.NT	List of triples describing sci-fi movies with a gross greater than eight billion of dollars.	188
FANTASY-NOVELS.NT	This dataset contains a set of triples describing fantasy novels published after year 2000.	693
CITIES.NT	Collection of triples describing cities with a total population greater than five millions.	127
DISEASES.NT	List of triples that enumerates infectious diseases.	36
DRUGS.NT	List that contains triples of medicines associated with infectious diseases.	54
ACTORS.NT	This collection of triples contains actors starring american sci-fsi movies.	166
MOVIES-AND-ACTORS.NT	This dataset combines a subset of `SCI-FI-MOVIES.NT` and `ACTORS.NT` datasets.	72
DISEASES-AND-DRUGS.NT	This collections of triples combining selected triples from `DISEASES.NT` and `DRUGS.NT`.	50

```
SELECT DISTINCT ?o WHERE {
<RDF Concept>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
?o .
} LIMIT 5000
```

FIGURE 5: Template of the SPARQL query to get the list of types associated with each centroid.

B. DATASETS

Datasets used to validate the semantic mapping building process are the result of performing a SPARQL query to DBPedia KG through its public endpoint [8] and results are saved in N-Triples format, i.e., each dataset is a list of subject-predicate-object triples. The intention of each dataset is to represent different knowledge domains accumulated in DBPedia and how they can be reduced and visualized using semantic maps. Table 2 reports the number of triples for each dataset.

C. HYPERPARAMETERS SELECTION

A key hyperparameter in the PAM clustering algorithm is the number of clusters we want to generate. Determining this hyperparameter is a crucial step in clustering and we determine this value by using the elbow method[9]. Table 3 describes the optimal number of clusters suggested by the elbow method for each dataset to use the PAM clustering strategy.

The *preference* parameter is a crucial hyperparameter in the Affinity Propagation clustering algorithm, which is a parameter that help to determine the number of clusters that will be generated. A higher preference value will result in more clusters, as more data points will be selected as exemplars, while a lower preference value will lead to fewer clusters, as fewer data points will be selected as exemplars. Therefore, it is often necessary to perform sensitivity analysis by trying different values of the preference parameter to find the optimal number of clusters. Our proposal is to maximize the silhouette index of resulting clustering after running Affinity Propagation with preference values that goes from 0.1 to 0.9 since this is the range of possible semantic distance values in the distance matrix.

TABLE 3: Optimal number of clusters(k) for PAM clustering suggested by selbow method.

Dataset	Optimal k	WSS
SCI-FI-MOVIES.NT	24	0.27
FANTASY-NOVELS.NT	40	1.73
CITIES.NT	16	1.69
DISEASES.NT	9	0.80
DRUGS.NT	10	8.40
ACTORS.NT	15	19.57
MOVIES-AND-ACTORS.NT	13	4.23
DISEASES-AND-DRUGS.NT	10	6.40

D. QUALITY OF SEMANTIC MAPS

The core of the semantic mapping process is to cluster the entity instances and obtain the set of centroids M. In order to provide a quantitative approach to validate semantic maps, we propose to associate the quality of clusters computed with the quality of semantic maps. With this evaluation strategy, we can learn how reliable are the groups shown in the semantic map.

Table 5 includes three columns describing the semantic map quality for two centroid-based clustering algorithms (PAM and Affinity Propagation) in terms of silhouette score,

[8] https://dbpedia.org/sparql/

[9] The elbow method is a heuristic approach to determine the optimal number of clusters in a datase and the idea behind this method is that as the number of clusters increases, the WCSS decreases, as the distance between each data point and its assigned center becomes smaller.

TABLE 4: Optimal value of *preference* hyperparameter in the Affinity Propagation clustering.

Dataset	Number of clusters k	Preference
SCI-FI-MOVIES.NT	5	0.8
FANTASY-NOVELS.NT	6	0.8
CITIES.NT	5	0.5
DISEASES.NT	3	0.6
DRUGS.NT	52	0.8
ACTORS.NT	3	0.0
MOVIES-AND-ACTORS.NT	4	0.7
DISEASES-AND-DRUGS.NT	3	0.7

Davies-Bouldin score, and Calinski-Harabasz index. The silhouette score measures how similar each entity is to its own cluster compared to other clusters, **with scores closer to 1** indicating better cluster quality. The Davies-Bouldin index measures the ratio of the within-cluster scatter to the between-cluster separation, **with lower scores** indicating better cluster quality. The Calinski-Harabasz index measures the ratio of between-cluster variance to within- cluster variance, **with higher scores** indicating better cluster quality.

E. INFERRED MAIN TERMS

For each experiment, semantic mapping process infers the main term α based in the IC_{graph} metric. Table 6 describes the inferred α for each dataset.

F. DISCUSSION

The quality of semantic map results indicate that the PAM algorithm outperforms the Affinity Propagation algorithm. The PAM algorithm achieved a higher silhouette score for all datasets, indicating that the clusters are more well-defined and separated compared to the Affinity Propagation algorithm's score. Additionally, the PAM algorithm's Davies-Bouldin scores suggests that the clusters are compact and well-separated for 5 of 8 datasets, while the Affinity Propagation algorithm's scores indicates that the clusters have significant overlap and are not well-separated. The Calinski-Harabasz index also supports the superiority of the PAM algorithm to generate good semantic maps, as its scores are significantly higher than the Affinity Propagation algorithm's scores for all datasets. Therefore, the PAM algorithm is a better choice than the Affinity Propagation algorithm, as it produces higher quality and better-separated semantic maps.

VI. CONCLUSIONS

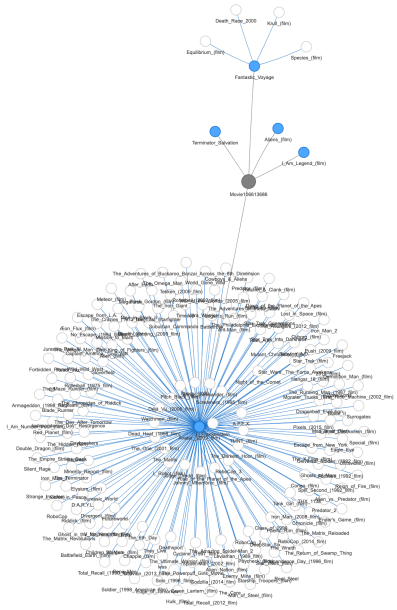

FIGURE 6: Semantic map obtained using AP clustering algorithm for MOVIES_SCIFI dataset.

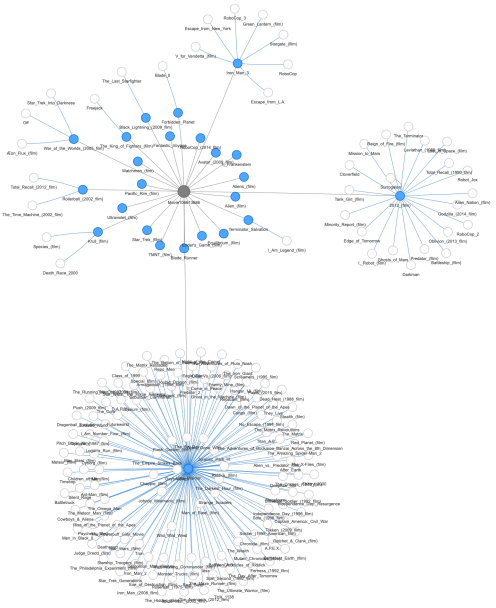

FIGURE 7: Semantic map obtained using PAM algorithm for MOVIES_SCIFI dataset.

TABLE 5: Semantic mapping quality results

	PAM			Affinity Propagation		
Dataset	Silhouette score	Davies-Bouldin score	Calinski-Harabasz Index	Silhouette score	Davies-Bouldin score	Calinski-Harabasz Index
MOVIES_SCIFI	**0.86**	**0.28**	**1366.07**	0.45	2.33	17.38
FANTASY_NOVELS	**0.66**	3.73	**133.56**	0.38	**1.53**	10.93
CITIES.NT	**0.69**	**0.46**	**281.70**	0.47	1.27	77.45
DISEASES.NT	**0.44**	**0.68**	**46.56**	0.43	2.84	5.52
DRUGS.NT	**0.33**	1.32	**11.06**	-0.02	**0.71**	0.63
ACTORS.NT	**0.40**	**1.26**	**52.87**	0.13	2.44	37.63
MOVIES-AND-ACTORS.NT	**0.55**	**0.67**	**103.26**	0.54	1.41	85.44
DISEASES-AND-DRUGS.NT	**0.54**	0.94	**91.39**	0.42	**0.57**	60.25

TABLE 6: Inferred main terms.

Dataset	**Inferred** α
SCI-FI-MOVIES.NT	`yago:Movie106613686`
FANTASY-NOVELS.NT	`yago:WikicatFantasyNovels`
CITIES.NT	`yago:City108524735`
DISEASES.NT	`yago:Disease114070360`
DRUGS.NT	`dbo:Drug`
ACTORS.NT	`yago:Actor109765278`
MOVIES-AND-ACTORS.NT	`yago:Whole100003553`
DISEASES-AND-DRUGS.NT	`yago:Abstraction100002137`