Towards Efficient Exploitation of Large Knowledge Bases by Context Graphs

Nada Mimouni, Jean-Claude Moissinac

Towards Efficient Exploitation of Large Knowledge Bases by Context Graphs

Nada MIMOUNI [a,1], and Jean-Claude MOISSINAC [b]
[a] *Center for Studies and Research in Computer Science and Communication, CNAM Paris*
[b] *LTCI, Télécom Paris, Institut polytechnique de Paris*

Abstract.
One challenge in utilizing knowledge graphs, especially with machine learning techniques, is the issue of scalability. In this context, we propose a method to substantially reduce the size of these graphs, allowing us to concentrate on the most relevant sections of the graph for a specific application or context. We define the notion of context graph as an extract from one or more general knowledge bases (such as DBpedia, Wikidata, Yago) that contains the set of information relevant to a specific domain while preserving the properties of the original graph. We validate the approach on a DBpedia excerpt for entities related to the Data&Musée project and the KORE reference set according to two aspects: the coverage of the context graph and the preservation of the similarity between its entities. The results show that the use of context graphs makes the exploitation of large knowledge bases more manageable and efficient while preserving the features of the initial graph.

Keywords. Knowledge base, Context graph, Similarity, DBpedia, Joconde database

1. Introduction

Developments in the semantic web and Linked Open Data (LOD) over the past decade have enabled the publication and linking of multiple structured data on the web. The *LOD cloud* [2] has grown from 12 to 1314 datasets with 16308 links and from 500 million to over 130 billion RDF triples between 2007 and 2020. These data cover several domains such as culture, life sciences, government data or geographic data. This development has demonstrated the interest of linking a dataset from a restricted application domain with an external dataset in order to get a better understanding and exploitation.In the framework of the Data&Musée project, an exploratory project aiming at improving the information systems of cultural institutions, we hypothesize that the promotion of cultural heritage can benefit from recent techniques of knowledge representation and exploration. To do so, we need to enrich datasets related to the project domain and ensure their interoperability to achieve increased visibility and accessibility by a wider audience. One of the direct consequences is the improvement of the financial conditions of cultural insti-

[1] Corresponding Author: Nada Mimouni, nada.mimouni@cnam.fr
[2] `https://lod-cloud.net/`

tutions to ensure a better preservation of this heritage. A recognized approach to ensure this type of data exploitation is the use of semantic web technologies, which have shown their power for the development of knowledge in various fields such as tourism [21,1], smart cities [7,8] or cultural heritage valorization. These techniques ensure a unified representation of heterogeneous but related data that facilitates their linking and enrichment. Large knowledge bases such as Yago, DBPedia, DBPedia-Fr and Wikidata are very useful resources because they provide a stock of semi-structured encyclopedic knowledge on LOD principles. But these resources pose several exploitation problems: access problems, performance problems, limitations on uses, etc. which are mainly related to their very large size. We are interested here in the problems of scale and performance that can arise when we want to exploit links to these large knowledge graphs. In this article, we propose a simplified, faithful and more accessible alternative representation of a knowledge base, a fortiori a large one, through a context graph. The extraction algorithm we propose constructs a context graph for a given domain, defined by a set of representative entities. The resulting graph preserves the properties of the original graph, while limiting performance and scaling problems.

We evaluate the properties of the extracted graph according to two criteria: its domain coverage and its impact on the results of a set of similarity measures between extracted entities. Indeed, assessing similarity between resources is crucial for several data-driven applications, such as link discovery, clustering or ranking. We performed a series of tests to validate our method on data from cultural institutions partners of the Data&Musée project. The results show that the use of context graphs makes the exploitation of large knowledge bases more manageable and efficient while preserving the properties of the initial graph.

In what follows, section 2 presents a state of the art on the use of contexts with knowledge bases. Section 3 recalls the basic notions on semantic graphs, gives the definitions we use in our approach and describes the process of context graph construction. Section 4 reviews similarity measures on knowledge graphs and presents our measure defined for the validation of a context graph. Sections 5 and 6 describe the experiments and validation tests performed respectively on the Paris Museums data and on the KORE reference dataset. The conclusion and perspectives are given in section 7.

2. Related work

The notion of context has been used in several works based on the semantic web for different applications such as the calculation of similarities between entities or between documents, the discovery of identity links for data binding on the LOD or the vector transformation of graphs for application to machine learning methods [19,15,2,3,20,13]. These approaches use an extract of the knowledge bases, called context, which consider it as a part of the large graph carrying semantics for one or several resources.

Semantics of context In [11], the authors describe a concept of interest C in DBpedia by a graph called a sense graph having C as its root. They propose a solution to the problem of automatic topic labeling using DBpedia. The topics are extracted by a method of *probabilistic topic modelling* (like LDA). For each concept C_i associated to a term of an identified topic, they extract a *sense graph* G_i by querying all nodes located at most two hops from C_i by recursively taking into account all links of type `nskos:broader`,

`rdfs:subClassOf`, `rdf:type` and `dcterms:subject`. The graphs G_i are then merged to obtain the topic graph G. In the same direction, the authors in [15,2] show that the use of contexts allows better description of entities to link them via identity links of type `owl:sameAs`. An identity link is valid in a context, corresponding to a subset of properties, if two instances i_1 and i_2 have the same values of these properties. They postulate that two similar instances in one context may not be similar in another with a different subset of properties. They thus show the importance of taking into account the context for the similarity computation. The idea of a knowledge base extract has also been studied in more specific domains such as IoT or environment to reduce the complexity of data manipulation in these domains. [8] proposes the LOV4IoT system for building semantic web of objects applications using domain ontologies in order to reduce search spaces and facilitate querying. The results in [24] show the positive impact of optimizations, such as domain constraints and neighborhood refinements, on reducing the complexity of the inference mechanism on animal behavior knowledge bases. These optimizations have reduced the computation time by half and thus improved the scaling.

Similarity in a context Most methods that compare resources, e.g. in terms of similarity, in the semantic web are based on a pre-selected set of triples. For their method of defining and computing the LCS (*Least Common Subsumer*: the most specific taxonomic ancestor that subsumes two resources) in RDF graphs, the authors show that it is important to make explicit the subgraph of the semantic web that serves as the context for computing the LCS for a resource r. The context of r, called *rooted r-graph* [6], consists of a set T_r of triples such that all resources in T_r are connected to r by a path in the RDF graph. In [5], a LOD-based inter-entity similarity measure is defined on a context (neighbors at depth N) extracted from the available dataset. For the inter-document similarity computation, the authors in [3] define the semantic context of analysis extracted from a knowledge base like DBpedia. From this context, they create a semantic context vector that outperforms classical inter-document similarity methods. In [4], the authors describe an algorithm for community detection and characterization based on knowledge graphs. They address the problem of finding the context that best summarizes the communities nodes. The algorithm uses a similarity measure that integrates the attributes of the nodes described in domain-specific hierarchical knowledge graphs (HKG). These graphs provide information relevant to a group of real-world objects.

Learning in context The use of knowledge graphs with machine learning methods has been mainly favored by the development of graph embedding techniques. This transformation preserves the relevant properties of the original graph such as topology (proximity between neighbors) or semantics. In this framework, [20] and [13] propose a knowledge graph *embedding* that creates vectors that are more representative of entities. The approach accounts for explicit (incoming and outgoing links and paths between pairs of entities) and implicit (contextual connectivity patterns) contexts between unconnected entities in this graph. An implicit context is constructed from the assumption that entities connected to the same node are generally implicitly related to each other, even if they are not directly linked in the graph.

Size of context Context size is a parameter that has been discussed in several works. In their work on automatic topic labeling with DBpedia [11], the authors use a distance of 2 jumps from the starting node. This distance was chosen following a series of tests on the expansion of nodes which showed that as from 3 jumps, the expansion produces very large graphs and introduces a lot of noise. For the definition of a LOD-based entity

similarity measure [5], the authors restrict themselves to paths of length 2 to retrieve all possible equivalent resources and enrich the instantiation space of a resource in the LOD.

These works emphasize the value of using contexts. However, in these approaches, the entire database is considered to calculate the context on the fly at the time of resource use, which poses access problems linked to the size of the database. In our approach, we propose to build a context graph, unique for all the resources in a domain, which will serve as an optimized access point for the various processing operations in a given application.

3. Domain-driven context graph

3.1. Remarks on semantic graphs

Our approach is based on knowledge bases described by an ontology in OWL and data represented in RDF. A knowledge base corresponds to a conceptual schema and a set of facts (statements).

Definition. Ontology An ontology $\mathscr{O}$ corresponds to the conceptual part of the base (schema) which structures the knowledge in a given domain. It can be represented by a triple $\mathscr{O} = (C, P_r, A)$ where C is the set of classes (concepts of a domain), P_r is the set of properties of classes and A is the set of axioms, which specify constraints on the properties of a class. In the following, we use *T-Box* to designate this conceptual part of the knowledge (see figure 1).

Definition. Facts and knowledge graph A knowledge graph $\mathscr{KG}$ is defined by a set of facts. A fact is represented by a triple of the form $\langle subject, predicate, object \rangle$. $subject$ designates an element on which we want to assert a knowledge; $predicate$ designates a property that we want to associate to the $subject$; $object$ is the value that the property takes for this $subject$. The set of facts constitutes the *A-Box* of a graph (see figure 1). This definition makes $\mathscr{KG}$ a labeled directed graph where $\mathscr{V}$ is the set of nodes (vertices) and $\mathscr{E}$ is the set of links between two nodes, links labeled by a predicate (edge). A fact described by $\langle subject, predicate, object \rangle \in \mathscr{E}$ is such that $subject, object \in \mathscr{V}$ and $predicate \in \mathscr{P}$, a set of predicates, for example chosen in a set of properties P_r defined in an ontology. $\mathscr{V}$ is the union of three disjoint sets:

$$\mathscr{V} = \{v \mid v \in \mathscr{U} \cup \mathscr{B} \cup \mathscr{L}\}$$

where $\mathscr{U}$ = set of URIs (unique resource identifier), $\mathscr{B}$ = set of blank nodes, vertices that have a technical role to group properties without associating them to a URI, $\mathscr{L}$ = set of literal values; these are typed values: strings, numerical values, dates, etc. A predicate links two URIs or blank nodes or a URI or blank node with a literal.

$$\mathscr{E} = \{(v_1, p, v_2) \mid v_1 \in \mathscr{U} \cup \mathscr{B}, v_2 \in \mathscr{U} \cup \mathscr{B} \cup \mathscr{L}, p \in \mathscr{P}\}$$

In our experiments, we use the French version of DBpedia as a generalist knowledge base, because of its wide coverage and the abundance and diversity of the links it contains and the fact that it is linked to many other bases. Figure 1 shows an example of a subgraph $\mathscr{KG}$ of DBpedia.com.

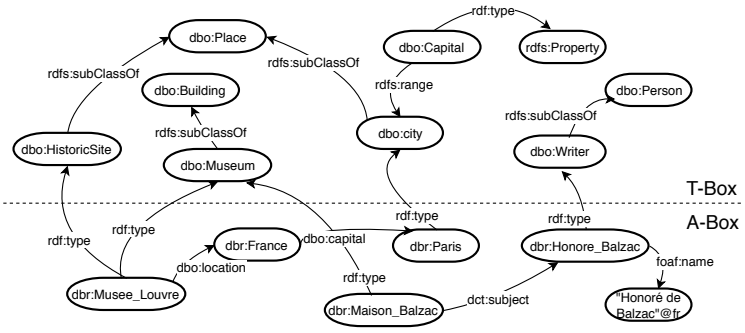

Figure 1.: Example of a subgraph $\mathscr{KG}$ of DBpedia.

Definition. Path A path $\mathscr{C}$ of length N, $\mathscr{C} = (e_i)_{1 \leq i \leq N}$, is a finite nonempty sequence of links of $\mathscr{E}$, with $N \in \mathbb{N}$, such that two consecutive links are adjacent. Two links l_1 and l_2 are adjacent when they share a node n destination for l_1 and origin for l_2. Note that links can be traversed in their normal direction or in reverse. Here is an example of a path `(Musée du Louvre, located in, Paris)(Paris, capital of, France)(France, is a, country)`.

Definition. Excluded predicate We define a set of predicates which will be excluded from the paths constructed on a graph $\mathscr{KG}$; we denote this set by $\overline{\mathscr{P}}$ such that $\overline{\mathscr{P}} \subset \mathscr{P}$. The set of links e labeled by predicates $p \in \overline{\mathscr{P}}$ is denoted by $\overline{\mathscr{E}}$ such that $\overline{\mathscr{E}} \subset \mathscr{E}$. For example, in the French DBpedia, the value of the predicate `dpb:WikiPageWikiLink` is the url of the wikipedia page associated with the subject. If we don't use this page, we can exclude this predicate without introducing biases.

Definition. Terminal node A terminal node is a node on which we impose the stop of the construction of a path (as if this node had no outgoing link). A path $\mathscr{C}$ built on a graph $\mathscr{KG}$ stops if it meets a terminal node; we denote this set of nodes by $\overline{\mathscr{V}}$ such that $\overline{\mathscr{V}} \subset \mathscr{V}$.

3.2. Context graph

Given a domain d and a graph $\mathscr{KG}$, our goal is to extract a subgraph of $\mathscr{KG}$ containing selected information on the domain d which we note $\mathscr{CG}(d)$. Our method is based on the parts *T-Box* and *A-Box* of $\mathscr{KG}$ by considering nodes $v \in (\mathscr{U} \cup \mathscr{L})$ and links $e \in (\mathscr{E} \setminus \overline{\mathscr{E}})$. To reduce the size of the context graph, we take into account observations made on $\mathscr{KG}$ and expert knowledge, when available, about the usefulness or uselessness of certain nodes and predicates. More precisely, the list $\overline{\mathscr{V}}$ is defined from $\mathscr{V}$ by the automatic exclusion of nodes that belong to the *T-Box* because of their very general character (ex. in DBpedia: `dbo:Building`, `dbo:Place` or `owl:Thing`) and structuring nouns (e.g. DBpedia page formatting, `<http://fr.dbpedia.org/resource/-Model:P.>`). In parallel, the list $\overline{\mathscr{E}}$ is defined from $\mathscr{E}$ by excluding links labeled by two types of predicates: predicates considered as little or not informative for the domain d (list fed by an expert) and predicates of base $\mathscr{KG}$ structuring (e.g., `<http://dbpedia.org/ontology/wikiPageRevisionID>`). These nodes and predicates introduce noise without bringing any relevant information for the considered

domain. As an example, in our experiments on DBpedia, we found 3486 nodes built on `<http://fr.dbpedia.org/\protect\discretionary{\char\hyphenchar\font}{}{}resource/Modele:****>` resulting in 2692515 links and 101235 links to `<http://www.w3.org/2004/02/skos/core#Concept>`. For the purpose of our application, we have constructed a list of excluded nodes and predicates that can be reused for applications in other domains. The list is available here [3].

3.3. Building Process

The extraction of a context graph $\mathscr{CG}(d)$ of dimension N is a recursive process that stops when the limit N is reached. The steps of the process are: (1) Seeds identification: a list of seeds $\mathscr{S}(d)$ is defined for a domain of application d. In some domains this list is obvious as in the case of museums, hotels or restaurants. In the general case, the common practice is to refer to a reference dataset (such as IMDB for the cinema domain). A list can also be drawn up with an expert in the field. (2) Construction of neighboring contexts: a neighboring context $\mathscr{CV}(a)$ is generated for any a entity of $\mathscr{S}(d)$ and the list of seeds $\mathscr{S}(d)$ is updated with the harvested neighbors. (3) Construction of the context graph: the context graph $\mathscr{CG}(d)$ is constructed as the aggregation of all the neighboring contexts $\mathscr{CV}$ of the seeds.

1. Seeds identification The seeds $\mathscr{S}(d)$ are nodes of $\mathscr{KG}$ which constitute the starting entities for the construction of the context graph $\mathscr{CG}$, $\mathscr{S}(d) = \{v | \forall v \in \mathscr{S}(d), v \in (\mathscr{V} \setminus \overline{\mathscr{V}})\}$. The list $\mathscr{S}(d)$ is defined for a domain d as the set of instances of concepts representative of the domain. For example, in our case (Data&Museum project), the starting entities correspond to the list of museums and monuments of Paris Museums and the Centre des Monuments Nationaux. *Example*. On figure 1: $\mathscr{S}(d)$ = { `dbr:Musee_Louvre`, `dbr:Maison_Balzac` }

2. Construction of neighboring contexts A neighboring context $\mathscr{CV}$ of an entity is its direct neighborhood (1-hop) in $\mathscr{KG}$. It is the local structure that interacts with the entity and reflects various aspects of that entity. More precisely, given an entity $a \in \mathscr{S}(d)$, the neighboring context of a is defined as follows:

$$\mathscr{CV}(a) = \mathscr{CS}(a) \cup \mathscr{CE}(a)$$

where

$$\mathscr{CS}(a) = \{(a,p,o) \mid \forall (a,p,o) \in \mathscr{E}, \forall p \in (\mathscr{P} \setminus \overline{\mathscr{P}}), o \in (\mathscr{U} \cup \mathscr{L})\}$$

$$\mathscr{CE}(a) = \{(s,p,a) \mid \forall (s,p,a) \in \mathscr{E}, \forall p \in (\mathscr{P} \setminus \overline{\mathscr{P}}), s \in \mathscr{U}\}$$

with $\mathscr{CS}$ a set of outgoing links from a while $\mathscr{CE}$ is a set of incoming links to a and s or o is the node neighboring a by a link labeled by p. Note here that $a \notin \overline{\mathscr{V}}$, so we do not construct neighboring contexts for the terminal nodes. The list of seeds $\mathscr{S}(d)$ is subsequently updated with the neighbors o and s collected such that $o, s \notin \overline{\mathscr{V}}$. *Example*. On figure 1 :

[3] `https://gitlab.com/-/snippets/3709662`

$\mathscr{CV}$(dbr:Musee_Louvre) = { (dbr:Musee_Louvre, rdf:type, dbo:HistoricSite), (dbr:Musee_Louvre, rdf:type, dbo:Museum), (dbr:Musee_Louvre, dbo:location, dbr:France)}
$\mathscr{CV}$(dbr:Maison_Balzac) = { (dbr:Maison_Balzac, rdf:type, dbo:Museum), (dbr:Maison_Balzac, dct:subject, dbr:Honore_Balzac)}

3. Construction of the context graph The construction of a context graph is a recursive process, based on the following step: $\mathscr{CG}(d)$ for a domain d is constructed as the aggregation of all contexts neighboring seed nodes $a \in \mathscr{S}(d)$ such that $\mathscr{S}(d) \subset (\mathscr{V} \setminus \overline{\mathscr{V}})$,

$$\mathscr{CG}(d) = \bigcup_{a \in \mathscr{S}(d)} \mathscr{CV}(a)$$

At the end of each step, the list of seeds $\mathscr{S}(d)$ is updated with the harvested v neighbors such that $v \notin \overline{\mathscr{V}}$. The process is repeated N times for a context graph of depth N. As shown by the work in section 2, $N = 2$ is the most interesting value for a context. Indeed, the size of the graph increases exponentially with depth (for entities that have on average x neighbors, at 1-hop the size is x, at 2-hop the size is x^2, at 3-hop the size is x^3, etc.), going beyond 2 increases significantly the space and introduces a lot of noise. In our experiment, at level 3 we would arrive at a graph of the same order of magnitude as whole DBPedia. At the end of the process, $\mathscr{CG}(d)$ is completed with the *T-Box* part of $\mathscr{KG}$ (here the DBpedia ontology [4]) and for any node in $\mathscr{CG}(d)$, we ensure that a link of type `is-a` exists with a concept of the part *T-Box* (if this link exists in the original graph $\mathscr{KG}$). The **core of the context** is the graph obtained at level $N-1$. The entities added at level N are the **periphery of the context**.

3.4. CONTEXT : Algorithm for context graph construction

The algorithm `CONTEXT` (algorithm 1) constructs a context graph *context* from a knowledge graph $\mathscr{KG}$ for a domain d. For a set of entities representative of a domain, the seeds (*germsATraiter*), `NeighborContext`(g) extracts a neighbor context C_v from a knowledge graph $\mathscr{KG}$ for each g. The final context, *context*, is enriched by C_v. A list of new seeds, *newSeeds*, is updated with the new entities harvested after filtering the terminal nodes with the method `FilteredEntities`. The exploration depth *level* is incremented by 1 at each step until the desired *radius* limit is reached. At the end of the process, the resulting context *context* is enriched by the classes of the set of entities extracted from $\mathscr{KG}$ by the methods `AddClasses` and `Entities`.

4. Validation of context graphs by similarity measure

4.1. Relevance hypothesis of a context graph

The use of a context graph built from a large knowledge graph allows, as mentioned above, to gain in performance (computing time, memory usage, etc.). This gain should not penalize its use by methods usually based on the structure and content of the original

[4] `https://www.dbpedia.org/resources/ontology/`

Algorithm 1 CONTEXT BUILDER

Function `ContextBuilder(`*KG, seedsEntities, radius, filteredEntities*`)`-
Input :
A knowledge graph *KG*
A neighborhood depth to reach *radius*
A set of entities which are used as seeds *seedsEntities*
A set of entities which are excluded from the seeds *filteredEntities*
Output: Context Graph *context*

1 $level \leftarrow 0$
2 $context \leftarrow \emptyset$
3 **while** $level < radius$ **do**
4 $newSeeds \leftarrow \emptyset$
5 **foreach** $s \in seedsEntities$ **do**
6 $C_s \leftarrow$ `FindNeighbors(`*KG, s*`)`
7 $context \leftarrow context \cup C_s$
8 $newSeeds \leftarrow newSeeds \cup$ `EntityFilter(`C_s*,filteredEntities*`)`
9 $level \leftarrow level + 1$
10 $seedsEntities \leftarrow newSeeds$
11 $context \leftarrow context \cup$ `AddClasses(`*KG,* `Entities(`*context*`))`
12 **return** *context*

graph. Indeed, several algorithms use knowledge graphs as a basic structure or as a source of semantic enrichment to perform several tasks in different domains such as social network analysis (e.g. community detection), recommendation (e-commerce, tourism, music), etc. Most of these methods rely on the notion of semantic similarity or semantic relatedness between entities (class instances) to perform final processing on the original data. The computation of these measures has several direct and relevant applications for automatic language processing (disambiguation, semantic annotation, information retrieval, etc.), link discovery or classification. Starting from the use cases mentioned above, we consider that a similarity measure is a necessary condition to evaluate if the use of a context graph can be sufficient to satisfy the computational needs of the tasks related to the methods applied to the original graphs. We then speak of the relevance of a context graph.

Definition. Relevant context graph A context graph is said to be relevant for a given domain if it preserves the properties of the original graph for this domain evaluated in terms of similarity between entities.

Hypothesis. We make the assumption that our context graph is relevant to a domain if the relative similarities of two entities to a third one in the original graph are preserved in the context graph.

In knowledge graphs, the semantics describing the resources are coded according to different aspects such as neighbors or class hierarchy. Most of the existing similarity measures consider aspects in isolation, which does not allow to cover all the properties of these resources. We define in the following a more general similarity measure (section 4.3) that composes the structural (taxonomic hierarchy links) and semantic (set of

predicates) aspect describing an entity. These measures will be used in sections 5 and 6 to evaluate the relevance of a context graph.

4.2. Review of similarity measures based on knowledge graphs

In the literature, we distinguish three main families of similarity measures that are based on ontologies, the T-Box part of a knowledge base (for a detailed review see [18,9]).

Link-based measures (*edge-counting*). These measures use the number of links separating nodes (cross- cutting relationship) as a criterion of similarity. The most direct measure is the one defined by [16] which computes the shortest path between two entities in a $\mathscr{KG}$ graph by following the links `is-a`: $dis(a,b) = min_i(N_i)$, N_i length of $\mathscr{C}_i \in \mathscr{C}$, $\mathscr{C}$ is the set of paths between a and b. Several improvements of this basic measure have been proposed to take into account the depth of nodes in the hierarchy (hierarchical relations) [25,12,14]. Most of these measures are based on LCS computation which has shown interest for information extraction tasks of the web of data: disambiguation and entity linking, detection of RDF data communities or automatic extraction of shared properties between resources [6].

Property-based measures (*feature-based*). These measures complement path-based methods by considering the degree of overlap between the properties of the entities being compared. The similarity is calculated as a function of the properties in common and the differences between the entities. The basic measure adopted is that defined by *Tversky* [23]: $sim_t(a,b) = \alpha.f(P(a) \cap P(b)) - \beta.f(P(a) \setminus P(b)) - \gamma.f(P(b) \setminus P(a))$, with $P(a)$ and $P(b)$ respectively the properties of the entities a and b. Several methods have been proposed depending on the choice of the nature of the properties (e.g. in WordNet, synsets and glosses have been used) and the computation of the weighting parameters $alpha$, $beta$ and $gamma$.

Content-based measures. These measures rely on text corpora to compute probabilities on word occurrence and thesauri (e.g. WordNet) to compute hyponyms of concepts, an aspect that is outside the scope of this study.

Combined measures. [17] proposes a measure that combines the information content of entities and their position in the graph. It should be noted that any similarity measure that utilises information content is contingent upon the corpus from which the information content was derived and the specific methodology employed in its generation [5].

Discussion. Similarity measures based purely on the knowledge graph (links, properties) are characterized by their simplicity and efficiency. They exploit the network of labeled vertices and links, unlike content-based methods that require external data sources. However, these measures consider aspects of resources in isolation and represent less of the full information around these nodes. In [22], the authors propose a similarity measure that combines different aspects of an entity and show that it gives a better correlation with the reference values. These aspects are neighbors, hierarchy and degree of a node or its specificity. The definition of this measure is close to our goal of representing the different aspects of resources in a graph. However, as it is defined, it is not applicable to our case as, by construction of the context graph, the degree aspect of a node (number of incidental links) is not preserved (in particular for the terminal nodes belonging to the T-Box). Only the two aspects neighbors and hierarchy can be exploited.

[5] `https://www.nltk.org/howto/wordnet.html` search Resnik word, checked 24/4/2024

The validation of the graph in our approach is thus based on these two aspects by combining two types of measures: (i) Measures based on links as they allow to cover the structural aspect in a graph (local structure of a node); (ii) Property-based measures are preferred as they tap into more semantic knowledge by assessing both commonalities and differences.

4.3. A similarity measure for context graph validation

We present a new similarity measure that relies on taxonomic links and properties of entities in a knowledge graph to validate the hypothesis in section 4.1. This measure consists of two parts. The first part is used to validate the structure of the graph by following the taxonomic links (of type `is-a`) to compare two entities. We use for this the measure of *Wu and Palmer* [25].

Definition. Similarity of links Let a and b be two entities in the graph, N_1 and N_2 respectively the number of links `is-a`[6] from a and b to their LCS, N_3 is the number of links `is-a` from the LCS to the root of the graph (root of the A-Box). The similarity $sim_l(a,b)$ between a and b is calculated as follows:

$$sim_l(a,b) = \frac{2 \times N_3}{N_1 + N_2 + 2 \times N_3} \qquad (1)$$

The second part is based on properties and follows the principle proposed in the model of *Tversky* (described in the section 4.2) which considers that the similarity between two entities is a function of their common and distinctive properties. Secondly, we consider the set of (property,value) pairs. The same definition is used for both measures with the set of properties or (property,value) pairs.

Definition. Similarity properties Let a and b be two entities in the graph, $\mathscr{P}_a = p : (a,v) \mid v \in \mathscr{U} \cup \mathscr{L}$ and $\mathscr{P}_b = \{p : (b,v) \mid v \in \mathscr{U} \cup \mathscr{L}\}$ respectively the set of properties of a and b. The similarity $sim_p(a,b)$ between a and b is computed according to the cardinality of their properties as follows:

$$sim_p(a,b) = \frac{|\mathscr{P}_a \cap \mathscr{P}_b|}{|\mathscr{P}_a \setminus \mathscr{P}_b| + |\mathscr{P}_b \setminus \cap \mathscr{P}_a| + |\mathscr{P}_a \cap \mathscr{P}_b|} \qquad (2)$$

Definition. Property-value similarity Let $\mathscr{KG}$ be a knowledge graph and a and b be two entities in $\mathscr{KG}$, $\Sigma_a = \{(p,v) \mid p \in \mathscr{P}_a, v \in \mathscr{U} \cup \mathscr{L}\}$ and $\Sigma_b = \{(p,v) \mid p \in \mathscr{P}_b, v \in \mathscr{U} \cup \mathscr{L}\}$ respectively the set of property-value pairs of a and b. The similarity $sim_{pv}(a,b)$ between a and b is computed in terms of the cardinality of the set of pairs as follows:

$$sim_{pv}(a,b) = \frac{|\Sigma_a \cap \Sigma_b|}{|\Sigma_a \setminus \Sigma_b| + |\Sigma_a \cap \Sigma_b|} \qquad (3)$$

[6] usually is-a is translated by the predicate rdf:type for rdf graphs. In the case of Wikidata, this corresponds instead to the predicate wdt:P31. The developments that follow may therefore require adjustments depending on the graphs used.

Definition. Aggregate similarity measure Let $\mathscr{KG}$ be a knowledge graph and a and b be two entities in $\mathscr{KG}$, the aggregate similarity measure is defined as follows:

$$sim(a,b) = \top(sim_l(a,b), sim_p(a,b), sim_{pv}(a,b)), \qquad (4)$$

where $\top$ is the average of the previous similarities. All measures are normalized in the interval $[0,1]$, where a score of 0 means that the resources compared are dissimilar, and a score of 1 means that the resources are identical.

5. Experiments and validation on data from Data&Musée

5.1. Data&Musée

This work is conducted within the framework of *Data&Musée* project [7]. This project aims to improve the capabilities of different cultural institutions by aggregating and analyzing data from these different institutions. The data collected and processed will be used in order to broaden the scope, loyalty and better understanding of their audiences. The partner institutions are the 14 museums of Paris Musées and the 84 monuments of the Centre des Monuments Nationaux. We present in the following the constitution of a context graph for these institutions [8].

5.2. Creation of a context graph for Data&Musée

As we have seen in section 3, the context graph is built from a list of entities, represented by their URIs. The list of entities is either chosen by a domain expert, or is made up of obvious entities (e.g., for the museums of Paris Musées, we manually search for an entity from DBpedia-fr corresponding to each museum). The context graph extraction process is parameterized by the dimension N of the graph. It is a recursive process for collecting neighboring nodes that depends on the choice of N. Based on the work described in the previous section and on the observations made during the experiments on our data, we consider that $N = 2$ is a good choice for the dimension of a context graph. The context for Paris Museums was thus constructed with a depth of 2. The core of the context has therefore a depth of 1. Studying the impact of this depth value choice is beyond the scope of this work. A blacklist has been created including essentially all the elements of the T-Box, considered as terminal nodes. Indeed, for example, if a node brought us to `owl:Thing` and we followed the links from there, we would bring back 1527645 entities not necessarily related to our domain.

Table 1 gives a description of a context graph extracted from DBpedia-fr for the depth $N = 2$. We test different settings for the constitution of such a graph. The numbers in this table are therefore only an indication of an example.

On the observed values, it is normal that there are fewer links per node in $\mathscr{CG}$ than in DBPedia-fr, since by construction we have eliminated certain links that are not very

[7] Data&Musée project was selected in the 23rd call for projects of the Fonds Unique Interministériel (FUI) and certified by Cap Digital and Imaginove. `https://imtech.imt.fr/2017/10/12/datamusee-data-institutions-culturelles/`, checked 24/4/2024.

[8] Data and code are available here : `https://gitlab.telecom-paris.fr/jean-claude.moissinac/contextgraph`.

Table 1.: Description of a context graph $\mathscr{CG}$ extracted from DBPedia-fr for $N = 2$

	$\mathscr{CG}$	DBPedia-fr	%
Distinct nodes	451653	10515624	4,29
Distinct predicates	2310	20322	11,36
Links	5150179	185404534	2,78
Links per node (average)	11,4	17,6	

informative in our application framework as explained above. We thus have a number L of links 36 times lower and a number S of vertices 23 times lower in the $\mathscr{CG}$ than in the $\mathscr{KG}$. On an algorithm which is in $O(L+S)$ -such as the breadth-first search-, we can thus anticipate a gain of a factor of the order of 30, which can strongly contribute to the applicability of some methods. The gains can become considerable on algorithms such as those for searching the shortest path between two nodes if we wish to give a weight to the links where we can be in $O(S^2)$. We will have to look further into these issues of contribution to scalability, but these first indications are favorable. We will see in the following other indicators that argue in favor of the context graph.

At the time of writing, we do not have comparative execution time measurements. However, we can judge that with the significant reduction in the size of the processed graph (see Table 1), we can achieve significant reductions in processing time due to the computational complexity of the task. Furthermore, it is possible to operate locally on the graph, whereas a complete installation of DBpedia is a time-consuming, complex and resource-intensive operation. Consequently, if such an installation is not carried out, all queries on the graph have to be made via the network, which necessarily has a significant impact on processing times.

5.3. Validation of the obtained context

5.3.1. Domain coverage by the context graph

To evaluate the relevance of our context, we wanted to see if it maintains a good coverage of the main elements concerning us in the Joconde database. The Joconde database is made up of metadata concerning nearly 600,000 Works of French heritage and is available in Open Data. Each work is mainly described by its location (city and institution), its creator(s), its title, the techniques of which it is part and time information. This database is important in our project since it brings a considerable amount of authoritative data for the description of the French heritage. Table 2 illustrates the coverage of Joconde database by our context graph. We took cities; to ensure that they cover the majority of the entities of interest to us -the artworks-, we have selected 10 cities associated with the greatest number of works in Joconde database and found them in DBPedia-Fr and verified that they are in our context graph. We proceeded in the same way for the creators, the domains and the museums associated with works. Entities are found by label matching, then verified by a human to ensure that the entity corresponds to the element sought.

We see that the cover is excellent. Only one creator has not been found, *J.B.Barla*: he is a naturalist who made many drawings of plants, but is not well referenced by DBpedia-Fr on this subject and therefore is not recognized as an element of our domain. For the coverage in number of works, we have, for example, 333114 works associated to the mentioned cities, that is to say more than half of the captured works for only 10

Table 2.: Coverage of the Joconde database by our context graph

	List	In $\mathscr{CG}$
Cities	Paris, Saint-Germain-en-Laye, Marseille, Strasbourg, Sèvres, Chantilly, Bordeaux, Montauban, Communauté urbaine Creusot-Montceau, Rennes	10/10
Domains	Dessin, Archéologie, Peinture, Ethnologie, Estampe, Sculpture, Photographie, Céramique, Costume, Néolithique	10/10
Creators	A.Rodin, H.Chapu, E.Boudin, G.Moreau, JB.Barla, Y.Jean-Haffen, T.Chassériau, Manufacture nationale de Sèvres, JBC.Corot, E.Delacroix	9/10
Museums	Louvre, Musée d'Archéologie nationale, Cité de la céramique, Musée Rodin, Musée Condé, Musée Ingres, Musée des beaux-arts(Strasbourg), Musée des beaux-arts(Rennes), Musée des beaux-arts(Angers), Musée Gustave-Moreau	10/10

cities. This demonstrates that our context graph provides good coverage of the Joconde database, which is one of the most important for the French cultural heritage domain.

5.3.2. Pertinence of a context graph by similarity measure

In this section we show that properties of the original graph, important for our work, are preserved in the constructed context graph.

Similarity links First, let us note that since we retrieve the types (links `is-a`) of all the entities present in the $\mathscr{CG}$, by construction, the LCS of two entities computed on the $\mathscr{KG}$ (DBpedia-Fr) and on our $\mathscr{CG}$ are identical for all entities. This constitutes a first index of relevance of the $\mathscr{CG}$. This first property allows us to assert that, for each pair of entities of our $\mathscr{CG}$, the similarity measure of *Wu-Palmer* (formula (1), section 4.3) obtained on our $\mathscr{CG}$ is identical to the one obtained on the $\mathscr{KG}$, since it depends only on (i) the LCS measurements that are identical on both graphs, (ii) the distance from the LCS to the root of the used *T-Box*, which is identical for both graphs since we have included in our $\mathscr{CG}$ the *T-Box* of DBPedia-Fr. We can therefore use this similarity measure on our $\mathscr{CG}$ without losing information. This constitutes a second index of relevance of our $\mathscr{CG}$.

Similarity properties We used the similarity measure of the properties of the entities defined by the *Tversky* measure (formula (2), section 4.3). As not all properties are conserved in our $\mathscr{CG}$, this similarity measure gives different results on the $\mathscr{CG}$ and on the $\mathscr{KG}$. In this case, using rank correlation as a metric to evaluate the relationship between two variables is a good indicator of preserving this measure. We thus define a rank correlation property and we have verified its validity on $\mathscr{CG}$.

Property. Rank correlation Let a, b and c be three entities of $\mathscr{CG}$ such that $a,b,c \notin \overline{\mathscr{V}}$ [9]. A rank correlation exists between the entity pairs (a,b) and (a,c) if:

$$sim_p^{\mathscr{KG}}(a,b) > sim_p^{\mathscr{KG}}(a,c) \Rightarrow sim_p^{\mathscr{CG}}(a,b) > sim_p^{\mathscr{CG}}(a,c) \qquad (5)$$

[9] defined in 3.1

with $sim_p^{\mathscr{KG}}$ and $sim_p^{\mathscr{CG}}$ are the respective similarity measures on $\mathscr{KG}$ and $\mathscr{CG}$. To verify property (5), we have applied the following algorithm on a set of entities a and e chosen randomly among the set of central nodes:

- choose $\{a_1,..a_m\} \notin \overline{\mathscr{V}}$ and $\{e_1,..e_n\} \notin \overline{\mathscr{V}}$.
- $\forall l \in \{1,2,..m\}, \forall i \in \{1,2,..n\}$, calculate $sim_p^{\mathscr{KG}}(a_l,e_i)$ and $sim_p^{\mathscr{CG}}(a_l,e_i)$
- $\forall (e_i,e_j) \mid i,j \in \{1,2,..n\}, i \neq j$, check the condition:

$$sim_p^{\mathscr{KG}}(a_l,e_i) > sim_p^{\mathscr{KG}}(a_l,e_j) \Rightarrow sim_p^{\mathscr{CG}}(a_l,e_i) > sim_p^{\mathscr{CG}}(a_l,e_j)$$

  and count the number of times it is verified.

We chose $m = 100$ and $n = 10$ then $m = 20$ and $n = 20$ and we performed $\frac{n(n+1)}{2} \times m$ checks to validate the rank correlation measure. Table 3.(a) shows the results of a series of tests for these different values of m and n:

Table 3.: (a) Rank correlation $\mathscr{KG}$ et $\mathscr{CG}$. (b) Spearman's correlation for similarity measure on $\mathscr{KG}$ and $\mathscr{CG}$

m	n	Nb. verifications	Nb. success	%
100	10	5500	5070	92,18
100	10	5500	5137	93,40
20	20	4200	3778	89,95
20	20	4200	3873	92,21

m	n	Spearman correlation
20	10	0.944
20	20	0.949
20	10	0.964
20	20	0.934

Thus, when we use this similarity to compare items and propose items to a user, in 90% of cases or more, the proposal we can make will be identical to the one we would have made on the full $\mathscr{KG}$.

Rank-order correlation coefficient of *Spearman*. We also calculated the correlation coefficient of *Spearman* which is the classical metric used in the literature to evaluate similarity measures. This correlation evaluates the monotonic relationship between two variables. Similarly, we computed this coefficient for $sim_p^{\mathscr{KG}}(a,e)$ and $sim_p^{\mathscr{CG}}(a,e)$ on a set of randomly selected a and e entities. We repeated this process several times and computed the overall measure $sim^{\mathscr{KG}}(a,e)$ and $sim^{\mathscr{CG}}(a,e)$. The results, described in Table 3.(b), show very good correlation values. The global similarity $sim(a,e)$ is computed as the average of $sim_l(a,e)$ and $sim_p(a,e)$. Experiments on the similarity $sim_{pv}(a,e)$ defined by the formula (3) give less good results. This is due to the fact that not all value-properties are preserved in $\mathscr{CG}$ (as described above). The first results seem very encouraging and persuade us to pursue the exploitation of context graphs in the Data&Musée project.

6. Experiments and validation on KORE data

In a general framework, benchmark data exists to evaluate similarity between entities. In order to compare the use of $\mathscr{CG}$ context graphs with the use of $\mathscr{KG}$ graph, we use the

KORE [10] benchmark dataset. It contains 21 main entities in 5 different domains: *IT companies, Hollywood celebrities, Television series, Video games* and *Chuck Norris*. For each of the main entities, it contains 20 entities ranked by similarity to it, with the most similar ranked first. This results in 420 pairs of entities ranked from most to least similar. We use Spearman's correlation as the evaluation metric. We have semi-automatically identified the set of KORE entities in DBpedia. For each of the 5 domains of KORE we have created a context graph using as seeds the set of its entities that we pass as input to the `CONTEXT` algorithm. As output we have 5 context graphs on which we evaluate the similarity for the pairs of entities of the dataset. We performed similarity calculations between KORE entities on these graphs and on DBPedia, in order to compare the obtained results. Table 4 gives the results of the Spearman correlation between $\mathscr{KG}$ and $\mathscr{CG}$ on the reference dataset. Each row of the table describes the correlation values for the corresponding similarity measure. The last row corresponds to the correlation on the ranking of the similarity measure calculated as the average of the previous three. The column 'Average' describes the correlation values for all entities of the considered KORE domains (the treatment of the domain *Video Games* had to be postponed for technical reasons).

Table 4.: Spearman correlation for similarity measures on $\mathscr{KG}$ and $\mathscr{CG}$(KORE)

Measure	IT Companies	Hollywood Celebrities	Television Series	Chuck Norris	Average
$sim_l(a,b)$	1.0	0.999	0.995	0.898	0.973
$sim_p(a,b)$	0.997	0.998	0.994	0.998	0.997
$sim_{pv}(a,b)$	0.590	0.807	0.646	0.806	0.712
$sim(a,b)$	0.994	0.996	0.957	0.986	0,983

We observe on the table that the measures $sim_l(a,b)$, $sim_p(a,b)$ and $sim(a,b)$ give very good values of correlation between the rankings obtained on $\mathscr{KG}$ and those on $\mathscr{CG}$, which is an element of confirmation in favor of the use of context graphs. As in the case of the context graphs of Paris Musées (section 5.3.2), on the context graphs of KORE the measure $sim_{pv}(a,b)$ gives less good results. Tests are underway to improve the results of this measure.

7. Conclusion and outlook

In this article we introduced the notion of a context graph for a domain. We define it as an extract of a larger graph and which targets the knowledge on this domain. We have shown that this graph can be constructed simply by starting from a few important entities of the domain using a starting dataset or with little expert knowledge if available. We have also shown that the obtained graph presents characteristics that allow it to be substituted to the large graph for classical exploitations of the knowledge graph (study on the similarity between elements). In the near future, we intend to apply this technique to other domains and to exploit the obtained context graphs to apply learning techniques on graphs.

References

[1] Al-Ghossein, M., Abdessalem, T., Barré, A.: Open data in the hotel industry: leveraging forthcoming events for hotel recommendation. J. of IT & Tourism **20**(1-4), 191–216 (2018). , https://doi.org/10.1007/s40558-018-0119-6

[2] Beek, W., Schlobach, S., van Harmelen, F.: A contextualised semantics for owl:sameas. In: Proceedings of the 13th European Semantic Web Conference. LNCS, vol. 9678, pp. 405–419. Springer (2016)

[3] Benedetti, F., Beneventano, D., Bergamaschi, S., Simonini, G.: Computing inter-document similarity with context semantic analysis. Information Systems **80**, 136 – 147 (2019)

[4] Bhatt, S., Padhee, S., Sheth, A., Chen, K., Shalin, V., Doran, D., Minnery, B.: Knowledge graph enhanced community detection and characterization. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. pp. 51–59. WSDM '19 (2019).

[5] Cheniki, N., Belkhir, A., Sam, Y., Messai, N.: Lods: A linked open data based similarity measure. In: 2016 IEEE 25th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE). pp. 229–234 (2016)

[6] Colucci, S., Donini, F.M., Giannini, S., Sciascio, E.D.: Defining and computing least common subsumers in rdf. J. Web Semant. **39**, 62–80 (2016)

[7] Consoli, S., Mongiovì, M., Nuzzolese, A.G., Peroni, S., Presutti, V., Recupero, D.R., Spampinato, D.: A smart city data model based on semantics best practice and principles. In: WWW'15 (2015)

[8] Gyrard, A., Atemezing, G., Bonnet, C., Boudaoud, K., Serrano, M.: Reusing and unifying background knowledge for internet of things with lov4iot. In: 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud). pp. 262–269 (2016)

[9] Harispe, S., Sánchez, D., Ranwez, S., Janaqi, S., Montmain, J.: A framework for unifying ontology-based semantic similarity measures: A study in the biomedical domain. Journal of Biomedical Informatics **48**, 38 – 53 (2014)

[10] Hoffart, J., Seufert, S., Ba Nguyen, D., Theobald, M., Weikum, G.: Kore: Keyphrase overlap relatedness for entity disambiguation (11 2012).

[11] Hulpus, I., Hayes, C., Karnstedt, M., Greene, D.: Unsupervised graph-based topic labelling using dbpedia. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining. pp. 465–474. WSDM '13 (2013)

[12] Li, Y., Bandar, Z.A., Mclean, D.: An approach for measuring semantic similarity between words using multiple information sources. IEEE Transactions on Knowledge and Data Engineering **15**(4), 871–882 (July 2003).

[13] Luo, Y., Wang, Q., Wang, B., Guo, L.: Context-dependent knowledge graph embedding. pp. 1656–1661 (01 2015)

[14] Christian Paul and Achim Rettinger and Aditya Mogadala and Craig A. Knoblock and Pedro A. Szekely: Efficient Graph-Based Document Similarity. International Semantic Web Conference (ISWC) 2016.

[15] Raad, J., Pernelle, N., Saïs, F.: Détection de liens d'identité contextuels dans une base de connaissances. In: IC 2017 - 28es Journées francophones d'Ingénierie des Connaissances. pp. 56–67. Caen, France (2017)

[16] Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man, and Cybernetics pp. 17–30 (1989).

[17] Philip Resnik: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. International Joint Conference on Artificial Intelligence (1995), https://api.semanticscholar.org/CorpusID:1752785

[18] Sánchez, D., Batet, M., Isern, D., Valls, A.: Ontology-based semantic similarity: A new feature-based approach. Expert Syst. Appl. **39**, 7718–7728 (2012)

[19] Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: Issues, techniques, and solutions. IEEE Transactions on Knowledge and Data Engineering **27**(2), 443–460 (2015)

[20] Shi, J., Gao, H., Qi, G., Zhou, Z.: Knowledge graph embedding with triple context. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 2299–2302. CIKM '17 (2017)

[21] Soualah Alila, F., Coustaty, M., Rempulski, N., Doucet, A.: Datatourism: designing an architecture to process tourism data. In: IFITT and ENTER 2016 Conferences (02 2016)

[22] Traverso, I., Vidal, M.E., Kämpgen, B., Sure-Vetter, Y.: Gades: A graph-based semantic similarity measure. In: Proceedings of the 12th International Conference on Semantic Systems. pp. 101–104. SEMANTiCS 2016 (2016)

[23] Tversky, A.: Features of similarity. Psychological Review **84**(4), 327–352 (1977).
[24] Wannous, R., Malki, J., Bouju, A., Vincent, C.: Trajectory ontology inference considering domain and temporal dimensions—application to marine mammals. Future Generation Computer Systems **68**, 491 – 499 (2017)
[25] Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics. pp. 133–138. ACL '94 (1994)