Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 176 (2020) 733–742

Procedia Computer Science

www.elsevier.com/locate/procedia

24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Improving the Performance of Ontological Querying by using a Contextual Approach

Wojciech Waloszek[a], Aleksander Waloszek[a]*

[a]*Faculty of Electronic, Telecommunication and Informatics, Gdansk University of Technology, Narutowicza 11/12, 80-233 Gdansk, Poland*

Abstract

In the paper we present the results of experiment we performed to determine whether a contextual approach may be used to increase the performance of querying a knowledge base. For the experiments we have used a unique setting where we put much effort in developing a contextual and a non-contextual ontology which are as much close counterparts as possible. To achieve this we created a contextual version of a non-contextual ontology and reformulated the set of competency questions to reflect the contextual structure of the newly created knowledge base. The results of the experiment strongly suggest that using contexts might be advantageous for improving performance, and also show the further ways of development of the approach.

Keywords: knowledge bases; contexts; Description Logics; reasoning

1. Introduction

In the decade after 2000 semantic technologies saw their bloom. They were based on the conception of Semantic Web described in [1] and followed the vision of annotated information resources arranged in large graphs understandable by both humans and machines. From this field of research came many methods for systematic creation of such graphs and underlying domain models, ontologies [2]. OWL standard [3] has been refined into the

* Corresponding author: Wojciech Waloszek. Tel.: +48583472475 ; fax: +48583472727 .
E-mail address: wowal@eti.pg.edu.pl

10.1016/j.procs.2020.09.062

second version and formal methods of very powerful reasoning from ontologies (including a very expressive Description Logic $\mathcal{SROIQ}$ [4]) have been developed.

Semantic Web tasks turned out to be challenging, though, due to the sizes of graphs. The graphs frequently embraced contents of large web sites like Wikipedia [5], and for such large number of individual nodes advanced methods of reasoning proved unfeasible. Instead, many more basic, structure-related methods of processing knowledge graphs have been developed. These methods embraced rule-based closed-world reasoning and querying [6], limiting the expressiveness of underlying logics [7], or even treating graphs as vector spaces similarly like in textual settings [8].

On the other hand, the methods of inference from annotated information resources based on expressive Description Logics remains the pinnacle of processing semantic data. Open-world reasoning [9] in conjunction with soundness and completeness offer very strong guarantees for tasks like ensuring consistency or knowledge base querying and may be extremely important in many domains.

An approach which may be alternative to reducing the expressiveness and sacrificing the open-world assumption consists in using contexts. Contextual knowledge bases [10, 11] are divided into modules. The information within each module might be expressed more concisely, as every context determines the set of contextual parameters. Just like in human communication, the notion of context in knowledge bases allows for vast reduction of information being processed [12]. The sentence "John Smith is the Dean" uttered in a specific context might in fact express much more complicated piece of information that "John Smith is the Dean of Faculty of Electronics in Gdansk University of Technology in May 2020".

In this paper we undertake a task to check whether and to what extent use of contextual knowledge base can be beneficial for performance of reasoning. For this we perform an experiment with a unique setting: we use reasoning over two versions of the same ontology: the contextual one and the non-contextual one (*flat*). For the results to be meaningful, it was necessary to ensure the maximal level of similarity between the two versions. We have achieved this by executing a several-step process that consisted of limiting the domain of the original ontology, and careful review of the set of competency questions [13].

The reviewed set of competency questions has also been used to measure the performance of querying both versions of the ontology. Each competency question has been translated into a query or a set of queries for each of the versions. Execution times of the queries have been measured with use of a specially prepared application, which ensured that the conditions of executing queries against both versions of the ontologies were similar.

The rest of the paper presents the process, the experiment, and its results, and is organized as follows.

2. Preliminaries

Throughout this paper we use the language of Description Logics to express assertions and axioms that form the contents of knowledge bases and ontologies. In the field of Semantic Web Description Logics are treated as the theoretical basis for most expressive variants of Web Ontology Language, OWL.

Description Logics ontologies are usually built over a structure $S = (\mathbf{C}, \mathbf{R}, \mathbf{A})$ which is called a *signature* and represents a vocabulary (or language) used. Notions in $\mathbf{C}$ represent *concepts*, in $\mathbf{R}$ *roles* (or properties), and in $\mathbf{A}$ individual objects (*individuals*). Ontology is given its model-theoretic semantics through interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \bullet^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is the domain of objects and $\bullet^{\mathcal{I}}$ is an interpretation function mapping concepts to subsets of $\Delta^{\mathcal{I}}$, roles to subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and individuals to objects in $\Delta^{\mathcal{I}}$.

The ontology itself consists of assertions (ABox) and axioms (TBox, terminology). Assertions are usually of the form $C(a)$, $C \in \mathbf{C}$, $a \in \mathbf{A}$ which assigns the individual a to a concept C, or $R(a, b)$, $R \in \mathbf{R}$, $a, b \in \mathbf{A}$ which relate two individuals with the role (property) R. Axioms in turn, allow for expressing interrelationships between concepts, like subsumption. An example of an axiom expressing subsumption is $\mathit{University} \sqsubseteq \mathit{Organization}$.

Description Logics are a family of formalisms that differ in the range of language construction one can use. Those constructions are usually *complex concepts or roles* built with use of constructors. Throughout the paper we will be using only a small range of complex concepts, roughly equivalent to the Description Logic $\mathcal{ALCIO}$. These complex concepts embrace: concept intersection $C \sqcap D$ ($(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$), existential quantification $\exists R.C$ ($\exists R.C^{\mathcal{I}} = \{x: (x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$), concept negation $\neg C$ ($\neg C^{\mathcal{I}} = \Delta^{\mathcal{I}} - C^{\mathcal{I}}$), role inversion R^{-} ($R^{-\mathcal{I}} = \{(y, x): (x, y) \in R^{\mathcal{I}}\}$), and individual sets $\{a, b, \dots\}$ ($\{a, b, \dots\}^{\mathcal{I}} = \{a^{\mathcal{I}}, b^{\mathcal{I}}, \dots\}$); C and D denote any concept, R any role.

Contextual Description Logics ontologies are relatively newer field of study [10, 11]. Such ontologies are usually divided into smaller pieces called *modules* or *contexts*. The axioms and assertions are typically placed within such contexts, which is denoted by prefixing them with the symbol of the module followed by a colon, e.g. c_1: *Winner*(*france*), c_2: *Loser*(*france*). To fully understand the meaning of the concept we have to consider the context it is used in, for instance we can assume that c_1 represents finals of FIFA World Championship in 2018 and c_2 in 2006.

Using such a construction saves us from specifying overly complicated axioms and assertions, by shifting some of the complexity into the structure of contexts. Expressing the same information in a flat (non-contextual) ontology would require us to introduce the concept of a *Match* and two additional roles *hasWinner* and *hasLoser*. It is therefore worth noting that in the context setting, we have "reduced the arity of required predicates", as the knowledge that has to be expressed with roles in a flat ontology is here expressed with concepts only. Moreover, introducing additional knowledge might be easier with use of contexts. For example specifying that a winner of the match cannot be at the same time a loser is trivial in the former setting (we only have to add the axiom $Winner \sqsubseteq \neg Loser$ to c_1 and c_2), while in a flat ontology would require us to use much more expressive Description Logics. This observation is the basis for our expectations of better performance of reasoning over contextual knowledge bases.

3. Experiment setup

The goal of our research was to investigate whether, and to what degree, the performance of reasoning can be increased by using contexts. To achieve this we planned to compare the performance of answering queries between the contextual and non-contextual (flat) version of the same ontology (we call the two versions test ontologies). We wanted to focus especially on ABox queries, as ABoxes (descriptions of individual objects) can easily grow to considerable sizes, and the process of querying them can benefit the most from dividing the ontology into contexts.

The main assumption we made was that the versions should be as close as possible, while still maintaining the contextual character of the former version. To achieve this, we have carried out the following several-step process.

Choosing an existing flat ontology: for choosing the ontology we assumed the following criteria: (1) the domain of interest should be general enough to be understood by non-specialists, (2) the ontology should be middle-sized and contain several hundred concepts and properties, (3) the ontology should be documented, there should be a paper and/or documentation publicly available.

Limiting the domain of the flat ontology: the rationale behind this step was to manage the resources (especially time) in our experiment, as creating contextualized ontologies is a non-normalized task, which has to be conducted basing mainly on expert knowledge and intuition. Choosing the fragment of the ontology has been done primarily by picking the base competency question, and determining the set of terms relevant to answering this question.

Preparation of the contextual version: this step consisted in designing the hierarchy of contexts. This step has been carried out mainly by identifying the parts of the ontology where complicated relationships between many entities have been described and by identifying the entities that could be treated as contextual parameters [14, 15].

Comparison of the two versions: the two versions have been carefully evaluated for their similarities and dissimilarities. During the evaluation various kinds of knowledge base updates (so also different kinds of knowledge that could be included in the knowledge base) were considered, and for each kind the way of its expressing in both versions was proposed.

Automated generation of ABox: in order to more thoroughly check the efficiency of ABox queries, we have created a generator of ABoxes for both the versions of the ontology. The creation of the generator allowed us to check the performance for several sizes of ABox. Moreover, the automated generation ensured that both the versions were conveying the same knowledge.

Preparation of queries and tests: a list of competency questions has been converted into two lists of queries, the first for the flat, and the second for the contextual version of the ontology. Both lists queries have been executed against their respective knowledge bases and the times (and the results, for verification) of execution have been compared.

The subsequent Sections describe the aforementioned phases in more details.

4. Preparation of the Versions

4.1. The SYNAT ontology

The ontology of our choice for the experiment is SYNAT Ontology. This ontology conforms to all the three postulates we formulated. Its domain of interest embraces scientific communities and their activities. The ontology consists of 472 classes and 296 properties and is described in details in [16]. The additional argument was Authors' familiarity with this ontology.

The ontology is internally divided into five parts, each one determined by its generic class:

1. *Agent* — the ancestor class of human agents, organizations and groups.
2. *InformationResource* — carriers of information, mainly physical, like documents, or electronic, like audio and video resources.
3. *Event* — the class representing all kinds of events associated with scientific and academic activity.
4. *Project* — describes scientific projects.
5. *Characteristic* — this class captures all kinds of abstract entities reifying properties of instances of all other concepts.

These five class are roots of separated trees of taxonomies. First four classes (hence entity classes) embrace entities that have their real-world counterparts. The last one (the characteristics class) is the root for the largest tree of subclasses defining properties of different kinds for the other classes of individuals.

An example of use of the last class is *PersonCharacteristic*. Its descendants describe basic personal data, like names and addresses, as well as all data related to the career in science and education. One of the consequences of defining characteristics in the above form is the fact that relating two instances of entity classes cannot be done directly, but with use of an instance of *Characteristic* which in fact reifies the relationships. This observation will become notable in the further part of the paper.

The division into generic concept has conceptual character, but from the deployment perspective the SYNAT ontology consists of three OWL files: *gio* which contains knowledge about geographic regions, *subject-areas* which describes the taxonomy of scientific fields of interest (from the perspective of Polish Ministry of Science), and *system* which contains the aforementioned entity and characteristic classes.

4.2. Limiting the domain of the ontology

In this part of our experiment we have chosen a fragment of the ontology that served as the basis for preparing the contextual and the flat versions. Expressing the whole SYNAT ontology in the contextualized way turned out to be too time-consuming, as every decision had to be made by experts on the basis of their experience. Therefore, we decided to limit the scope of the ontology to a chosen subdomain of interest.

Owing to participation of the Authors in SYNAT project we had a prepared list of competency questions for the ontology. For the purposes of this experiment the list has been carefully reviewed.

We decided to limit the scope of the ontology by choosing a single base competency question: which members of department x had their scientific paper published by y. This base competency question was then used to prepare a list of phenomena whose representations should be preserved in the limited ontology:

- persons' membership of an institution (department, and their parent institutions like faculty, university, etc.),
- authorship of a scientific paper,
- containment of a scientific paper within a book,
- editorship of a book by a person working for a publisher.

In the final test versions of the ontology we decided to only leave classes and properties directly related to the above list, which also resulted in removal of the imported files *gio* and *subject-areas*. The former describes geographic and lingual information, while the latter classification of field of science from the perspective of Polish Ministry of Science. Both the kinds of knowledge represent details that are not included in the list of phenomena.

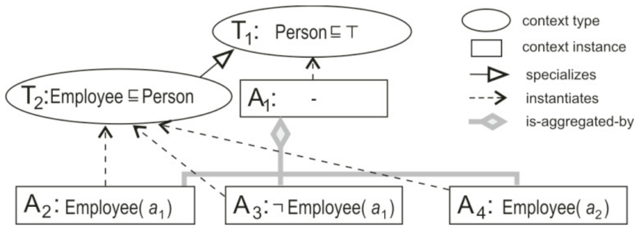

Fig. 1. An example of SIM ontology.

4.3. Preparation of the contextualized version – environment

The contextual version of the ontology has been prepared with use of SIM model [17]. SIM allows for creating contextual knowledge bases in which contexts are arranged into a hierarchy. The main assumption is that the contexts higher in the hierarchy describe larger portions of the domain of interest but in less detail, and naturally the opposite holds for the lower contexts.

SIM ontology is organized into two kinds of modules: terminological, which contain axioms (thus introducing concepts and relationships between them), and assertional. The former are called *context types*, and the latter *context instances*. Both context types, and context instances, form their own hierarchy (of inheritance and aggregation resp.). In addition every context instance has to *instantiate* one of context types (meaning that it inherits all the terminology from this specific module). A context instance together with the context type it instantiates form a *context*.

The paper [17] defines in detail the model-theoretical semantics of SIM. Here we will present a shortened version. Interpretation of a SIM ontology is in fact a set of interpretations of its contexts, $\mathcal{I} = (\{\mathcal{I}_j\}_{j\in J})$, where J is the set of indices over all context instances (and therefore contexts). For $\mathcal{I}$ to be a model of the ontology, each $\mathcal{I}_j$ has to (1) satisfy all the assertions in the context instance A_j, (2) satisfy all the axioms in the context type it instantiates (denoted $T_{inst(j)}$), and (3) satisfy the *aggregation conformance rules*. The aggregation conformance rules assure consistency between contexts. The general assumptions in SIM model is that the context "sees" terminologies from higher levels (as it refines them by introducing more detailed concepts and roles) but not from lower levels. The aggregation conformance rules state that all the conclusions flow down the hierarchy of context instances, and those conclusions that fit in the signature (vocabulary) of the higher context instances also flow up.

The form of the aggregation conformance rules signifies the importance of the proper choice of signatures along the context hierarchy. Figure 1 shows a fragment of a hierarchy of contexts whose goal is to store the information about employment of persons in institutions. Context instances A_2, A_3 and A_4 contain seemingly contradictory knowledge about individuals a_1 and a_2. This is possible in SIM, however, and does not make the knowledge base inconsistent, because the concept *Employee* is not present in T_1. Interpretations of this concept in A_2, A_3 and A_4 can therefore be different and still satisfy the rules, and thus the base remains consistent.

Figure 1 also shows the important aspect of contextualization in SIM base. If we assume that each of context instances A_2, A_3 and A_4 describes different organization, say o_1, o_2, and o_3, then we can express the fact that a_1 is employed in o_1 simply by formulating the unary assertion *Employee*(a_1) in the appropriate context. This information should otherwise be expressed by a binary relation (assertion), so we can say that the structure of contexts allowed for decreasing the arity of the predicate needed to express this fact.

Querying a SIM knowledge base consists in picking a context instance first and then issuing a query against the context. Continuing the example from Fig. 1, to check whether a_1 is an employee in o_1, a knowledge base user would have to pick the context instance A_2 and then ask whether a_1 is an instance of *Employee*.

SIM model has been implemented in CongloS system [18] (*https://conglos.org*). The system is a set of libraries and plug-ins to well-known Protégé [19] application. It allows for creating SIM knowledge bases, and within those bases creating context types and instances and connecting them with inheritance, instantiation and aggregation relations. With use of CongloS it is also possible to query contexts, as there is a special SIM reasoner included in the system. The reasoner uses a specific approach as it first calculates the smallest relevant set of axioms and assertions needed to fulfill the aggregation conformance rules, and then deploys a standard Description Logics inference engine to reason over them.

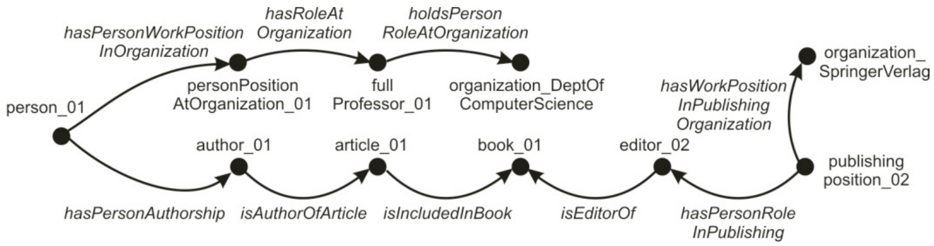

Fig. 2. The way of expressing authorship and holding a position in SYNAT ontology.

4.4. Preparation of the contextualized version – execution

We started our work over contextualizing SYNAT ontology from analyzing the base competency question. The question is about a person being an employee of a specific academic organization (Dept. of Computer Science) and being an author of works accepted by a specific publisher (Springer-Verlag). Such information can be expressed in flat SYNAT ontology in rather convoluted form (Fig. 2).

The complication of the graph is the result of modeling characteristics of base entities by using additional objects. Base entities (Agents and Information Resources) in the picture are *person_01*, *organization_DeptOfComputerScience*, *organization_SpringerVerlag*, *article_01*, *book_01*. The remaining individuals are reifications of characteristics of the objects and specify relationships between them. Reifications in ontologies often represent *n*-ary predicates [20], so we focused our approach on reducing the arity of the predicates in order to remove excessive individuals.

As a first step to contextualization of the base we divided the domain of interest of the ontology into two fragments: objects we wanted to model as individuals (like in the original ontology) and objects we wanted to reflect as elements of contextual structure. This decision has been made on the basis of analysis of competency questions. As a result we decided to reflect in the contextual structure institutions like publishers, faculties, and institutes.

In the effect we obtained the modular structure of the knowledge base depicted in Fig. 3. In the figure context types are denoted with ovals, and context instances with rectangles. The names of the context types are prefixed with *C-* and should be interpreted as referring to the contents of their context instances, e.g. the name *C-Universities* means that each instance of this context type represents a group of universities, similarly, the name *C-University* indicates that each context instance of this type represents a single university. The names of context instances are prefixed with *I-* and may carry additional information, e.g. about which university is represented by this instance.

As it can be seen in Fig. 3, the hierarchy of context types is divided into two major branches. At the top of this tree there is context type *C-Organizations*. In this contexts only very general vocabulary is introduced, namely concepts: *Person*, *Employee* (subsumed by *Person*), and *Product*. (It is worth noting, that the vocabulary does not concern organizations themselves—they are represented not as individuals but as modules—other entities important from the point of view of an organization.)

The left branch concerns publishers: in the context type *C-Publishers* we introduce a set of new terms connected with publishing, among others concepts: *Book*, *Article* (both being subsumed by *Product*), *Editor* (subsumed by *Employee*) and roles: *hasAuthor*, *hasEditor*, *includesArticle*. The context type *C-Publisher* is intended to embrace context instances representing single publishers. The most important concepts introduced here are *LocalArticle* (subsumed by *Article*), *LocalAuthor* (subsumed by *Author*), *LocalEmployee* (subsumed by *Employee*), *LocalEditor* (subsumed by *LocalEmployee* and *Editor*), denoting respectively articles published by the specific publisher, their authors, and employees and editors of the specific publisher.

Such a division into general concepts (like *Author* or *Editor*) and local ones (resp. *LocalAuthor* and *LocalEditor*) has been introduced to fulfill two goals. On one hand, we wanted to preserve the ability of the knowledge base to answer questions like "list all the editors in a knowledge base". This can be done at the level of *C-Publishers* by asking about instances of *Editor* (which is therefore a counterpart to *Person* in Fig. 1). On the other hand, we wanted to be possible for a single individual to be an editor in one publisher house but not in another (like being an *Employee* in Fig. 1). For this we needed the concept *LocalEditor* introduced at the level of *C-Publisher*.

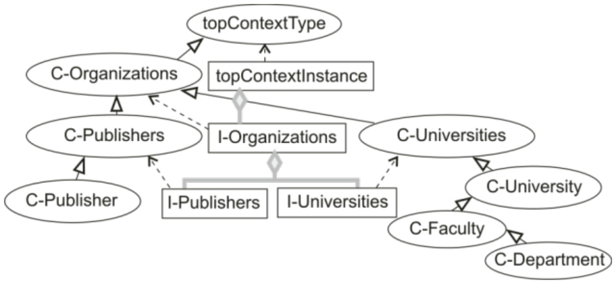

Fig. 3. The way of expressing authorship and holding a position in SYNAT ontology.

The right branch of the tree is organized similarly as the left one, the only difference being that also some aspects of organizational structure are reflected in the tree: as a consequence, here a university is understood as both a single university and a group of faculties it includes (resp. between a faculty and its departments). As the result of this difference, instead of *Local…* prefix we used here several prefixes (each on one level) like *UniversityEmployee*, *FacultyEmployee*, *DepartmentEmployee*.

In such a knowledge base university faculties and departments as well as publishers are represented by context instances. Therefore the base competency question can be executed by issuing two queries against the SIM knowledge base, the first query about members of the concept *DepartmentEmployee* in the context instance *I-DeptOfComputerScience*, and the second about members of *LocalEditor* in the context instance *I-SpringerVerlag*, and then intersect the results. In practice, to avoid burdening the user with the necessity of asking two queries, we took advantage of CongloS function which allowed for reasoning over more than one context instance at the same time.

5. Experimental evaluation

5.1. Automated generation of ABox

SYNAT ontology comes with only very small exemplary ABox (embracing 7 persons and 3 organizations). For that reason, to execute more extensive performance tests, we decided to remove the individuals and generate larger ABox with the aid of computer. For this task we prepared a generator written in Java and using JavaFX, OWL API, and CongloS libraries (for generating contextual ABoxes).

The generator has been used to create three sets of increasingly larger ABoxes (each containing a flat ABox and contextual ABox of the same contents) that have been later used for tests.

5.2. Preparation of queries

For the tests we have prepared a list of 31 questions that covered the limited ontology domain. We arranged them into 7 categories, depending on which part of the ontology they concerned:

1. general questions (related to the ontology as a whole),
2. questions related to a specific publisher *pubX*,
3. questions related to a specific department *depX*,
4. questions related to a specific faculty *facX*,
5. questions related to a specific university *univX*,
6. questions related to a specific person *perX*,
7. questions related to a specific book *bookX*.

Each question was expressed in natural language, and each has been converted into queries for the flat and the contextual version of the ontology. The examples of questions (about members of a concept) and the results of conversions are presented in Tab. 1. Some of the queries are close variants of each other (like about various posts at a selected department), and in these cases the queries have the same number but ended with different letters.

Table 1. Examples of Questions and their translations into Queries (CI = Context Instance).

Question (category.number)	Flat query	Contextual query
List all the persons (1.1)	*Person*	CI: *I-Organizations* Query: *Person*
List all the persons who edited any book (1.2)	$\exists$*hasPersonWorkPositionInPublishing* .$\exists$*hasPersonRoleInPublishing.Editor*	CI: *I-Publishers* Query: *Editor*
All the books published by *pubX* (2.3)	$\exists$*hasEditor*.$\exists$*hasPersonRoleInPublishing*$^{-}$.$\exists$*hasWorkPositionInPublishingOrganization*.{*pubX*}	CI: *pubX* Query: *LocalBook*
All the persons who hold any position in department *depX* (3.1)	$\exists$*hasPersonWorkPositionAtOrganization* .$\exists$*hasRoleAtOrganization* .$\exists$*holdsPersonRoleAtOrganization*.{*depX*}	CI: *depX* Q: *DepartmentResearch AndTeachingEmployee*

5.3. Tests

The tests have been carried out with use of a dedicated Java application. The application executed queries against the flat and contextual version of the ontology, measuring the times of execution. To reason over both the versions the same reasoner has been used (in the case of the contextual test ontology it was used for reasoning over the calculated set of sentences). For this task we chose Hermit 1.3.8 [21], because of it being written in pure Java and behaving very stable with Protégé 4 (the version for which the original CongloS was written).

The computer used for testing was equipped in Intel i5-4300M processor and 8GB of RAM. The application repeated the execution of each query several times (each time restarting the reasoner to eliminate the caching effect). For the queries from categories 2-7 three different parameters (publishers, department, faculties, etc.) have been randomly drawn. The times of execution have been averaged for each query, and gathered.

Table 2. The results of the performance tests, time of execution, ctx = contextual, times in millseconds

Query		1.1	1.2	1.3	1.4a	1.4b	1.4c	1.4d	1.4e	1.5	1.6	2.1	2.2	2.3
ABox1	flat	96	2856	1902	2926	2931	2990	2974	2994	100	96	2856	3259	2111
	ctx	1349	1868	1873	273	281	276	276	283	1811	1349	1868	308	299
	ratio	0.07	1.53	1.02	10.72	10.43	10.83	10.78	10.58	0.06	0.07	1.53	10.58	7.05
ABox2	flat	230	15732	9012	16416	15592	16157	16267	16215	237	230	15732	17149	9759
	ctx	5637	8287	8566	603	618	617	621	594	8011	5637	8287	862	802
	ratio	0.04	1.90	1.05	27.22	25.23	26.19	26.19	27.30	0.03	0.04	1.90	19.89	12.17
ABox3	flat	747	115167	67199	113403	113049	118346	119405	119197	725	747	115167	123106	66210
	ctx	39396	61740	61725	1736	1688	1703	1688	1714	61513	39396	61740	4185	3877
	ratio	0.02	1.87	1.09	65.32	66.97	69.49	70.74	69.54	0.01	0.02	1.87	29.42	17.08

Query		2.4	3.1	3.2a	3.2b	3.2c	3.3	4.1	4.2a	4.2b	4.2c	6.1	7.1
ABox1	flat	8607	3187	3208	3180	3171	3287	3194	3186	3219	3197	8568	3260
	ctx	355	231	231	233	234	2905	218	212	212	219	1216	563
	ratio	24.22	13.80	13.87	13.65	13.53	1.13	14.67	15.01	15.21	14.58	7.05	5.79
ABox2	flat	50590	16923	16950	16832	16964	17404	17090	16979	16820	16788	50015	17269
	ctx	813	377	391	372	374	13511	364	361	359	364	4654	1958
	ratio	62.23	44.89	43.39	45.25	45.32	1.29	46.99	47.03	46.81	46.16	10.75	8.82
ABox3	flat	476396	120306	122872	123142	122375	127468	123995	121407	120648	121709	483022	123914
	ctx	3752	764	775	812	800	91612	764	790	772	783	29962	8232
	ratio	126.96	157.54	158.48	151.71	152.97	1.39	162.23	153.61	156.21	155.51	16.12	15.05

The whole process has been repeated for three gradually larger ABoxes. Each ABox contained 3 universities, 3 faculties per university, 3 departments per faculty, 8 publishers, and additionally:

- ABox1: 160 people, between 100 and 120 books, 4-5 articles (with up to 4 authors) per book,
- ABox2: 320 people, 180 books, 4-7 articles (with up to 6 authors) per book,
- ABox3: 640 people, 340 books, 6-9 articles (with up to 8 authors) per book.

The results are presented in Table 2.

5.4. Discussion

While analyzing the Table 2 we can divide the queries into three categories:

- those whose times of execution are substantially longer for the contextual ontology (1.1, 1.5, and 1.6),
- those whose times of execution are similar for both the test ontologies (1.2, 1.3, and 3.3),
- finally the rest of the queries, whose times of execution are substantially shorter for the contextual ontology.

The queries in the first category are very specific, as they retrieve all the Persons (1.1), Books (1.5), and Articles (1.6) from the ontology. We currently investigate the reasons why the execution of these queries is longer for the contextual ontology. However, these queries are extremely non-contextual in their nature, which is very easy to determine. It is not difficult to imagine that in a deployed system they could be easily be redirected to a non-contextual version of the ontology.

The majority of queries fall in the last category. They (like 2.3 and 3.1) concern a selected context, and this feature let them be executed several times faster in the contextual test ontology. What is also notable is the fact that the speed-up is growing with the growing ABox, which is an indication that contextual mechanisms can improve the execution times of significant number of queries for ontologies with large ABoxes.

While the presented results are encouraging, one has to bear in mind that the increase in performance does not come without cost. In the contextual version publishers, universities, faculties, and departments have been excluded from the set of individuals, and now constitute the structure of the ontology. This means that the user needs to know the structure in order to ask a question, and that it is impossible to include in the ontology an organization about which we do not know where in the hierarchy it should be placed. In the terms of query answering it also means that the questions about organizations have to behave in closed-world fashion (so we can ask questions about the organizations we know, but we cannot draw a conclusion that some additional organization exists). From the point of view of assumed use of SYNAT ontology it did not pose a problem, but the situation might be different in other settings.

6. Summary

In the paper we presented the results of the experiment, whose goal was to determine whether it is possible to improve performance of query answering with use of contextual techniques.

The results of the experiments are notable, because they seem to indicate that contextual methods have the potential to improve the performance of reasoning especially over knowledge bases with large numbers of individuals. Such knowledge bases are becoming prevalent with the advent and development of interest in Knowledge Graphs.

While studies on contextual knowledge bases performance exist [18], the setting of the experiment was specific, as much care has been devoted to preparation of analogous versions of test ontologies: the flat one and the contextual one. Moreover, use of SIM model gave the unique opportunity to use in fact the same reasoner (Hermit) for final delivery of the results to an end user.

The experiment showed that there are relatively many queries which may benefit from being executed against a contextual ontology. Further development of SIM model may result in improving the performance of answering broader ranges of queries, and in creation of methods that will be able to contextualize existing ontologies more easily, and possibly on-demand. Additional experiments which we plan to execute, including contextualizing other ontologies and use of various reasoners, may in turn give more information about the possible range of applications of the proposed approach.

Contextual methods give hope for better arrangement of ontologies, and for better exposure of assumptions behind their design. While the ability to flexibly manipulate contexts could prove itself very useful in ontology engineering tasks like ontology alignment. We hope that with further development of the methods, they will become more popular and mature enough to be used broadly throughout many Semantic Web-related tasks.

Acknowledgements

The authors of this paper want to warmly thank Ms. Marta Żuławska for her work on the experiment.

References

[1] Berners-Lee, T., Hendler, J., Lassila, O. (2001) "The Semantic Web." *Scientific American*, May 2001.

[2] Gruber , T. R. (1993) "A translation approach to portable ontologies." *Knowledge Acquisition*, **5 (2)**:199–220

[3] "OWL 2 Web Ontology Language Document Overview", https://www.w3.org/TR/owl2-overview/, Accessed May 2020

[4] Horrocks, I., Kutz, O., Sattler, U. (2006) "The even more irresistible $\mathcal{SROIQ}$", in *Proceedings of KR'06*

[5] Lehmann, J. et al. (2012) "DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia", *Semantic Web* **1 (1–5)**

[6] Herman, J. ter Horst (2005) "Completeness, Decidability and Complexity of Entailment for RDF Schema and a Semantic Extension Involving the OWL Vocabulary" *Journal of Web Semantics* **3**: 79–115

[7] Baader, F., Brandt, S., Lutz C. (2005) "Pushing the EL Envelope", in: *Proceedings of IJCAI'2005*

[8] Ristoski, P., Rosati, J., Di Noia, T., De Leone, R., Paulheim, H. (2018) "RDF2Vec: RDF graph embeddings and their applications" Semantic Web **10**: 1-32.

[9] Reiter, R. (1978) "On Closed World Data Bases" in: Gallaire H., Minker J. (eds) *Logic and Data Bases*

[10] Homola, M., Serafini, L. (2012) "Contextualized knowledge repositories for the Semantic Web." *Journal of Web Semantics* **12–13**: 64–87

[11] Klarman S. (2013) "Reasoning with Contexts in Description Logics." Doctoral Thesis, Free University of Amsterdam

[12] Sowa, J.F. (1995) "Syntax, semantics, and pragmatics of contexts", in: Ellis G., Levinson R., Rich W., Sowa J.F. (eds) *Conceptual Structures: Applications, Implementation and Theory*, Springer, Berlin

[13] Gruninger, M. Fox, M.S. (1995) "Methodology for the Design and Evaluation of Ontologies", in: *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing*, IJCAI-95

[14] McCarthy, J. (1993) "Notes on Formalizing Context", in: *Proc. of 13th International Joint Conference on Artifical Intelligence*

[15] Bar-Hillel, Y. (1954) "Indexical Expressions" Mind, **63**: 359–379

[16] Wróblewska, A., Podsiadły-Marczykowska, T., Bembenik, R., Rybiński, H., Protaziuk, G. (2013) "SYNAT System Ontology: Design Patterns Applied to Modeling of Scientific Community, Preliminary Model Evaluation", in: *Intelligent Tools for Building a Scientific Information Platform: Advanced Architectures and Solutions*, Springer, 2013

[17] Goczyła, K., Waloszek, A., Waloszek, W. (2007) "Contextualization of a DL knowledge base", in: *Proceedings of DL Workshop DL'2007*

[18] Goczyła, K., Waloszek, A., Waloszek, W., Zawadzka, T. (2013) "Theoretical and Architectural Framework for Contextual Knowledge Bases", in: *Intelligent tools for Building a Scientific Information Platform : Advanced architectures and solutions*, Springer.

[19] Musen, M.A. (2015) "The Protégé project: A look back and a look forward" in *AI Matters. ACM Specific Interest Group in Artificial Intelligence* **1(4)**

[20] "N-ary relations ODP", http://www.w3.org/TR/swbp-n-aryRelations/, Access 2019-12-06.

[21] Glimm, B., Horocks, I., Motik, B., Stoilos, G., Wang, Z. (2014) "HermiT: An OWL 2 Reasoner", in *Journal of Automated Reasoning* **53 (3)**

[22] Bozzato, L., Serafini, L. (2014) "Knowledge Propagation in Contextualized Knowledge Repositories: An Experimental Evaluation", in: *Proceedings of EKAW'2014 (Satellite Events)*, Springer.