Open Access

Open Journal of Semantic Web (OJSW)
Volume 4, Issue 1, 2017

www.ronpub.com/ojsw
ISSN 2199-336X

Assessing and Improving Domain Knowledge Representation in DBpedia

Ludovic Font [A], Amal Zouaq [A, B], Michel Gagnon [A]

[A] Département de Génie Informatique, École Polytechnique de Montréal, 2500, chemin de Polytechnique, H3T1J4, Montréal, Québec, Canada, {ludovic.font@polymtl.ca, michel.gagnon@polymtl.ca}

[B] School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward Avenue, K1N6N5, Ottawa, Ontario, Canada, azouaq@uottawa.ca

ABSTRACT

With the development of knowledge graphs and the billions of triples generated on the Linked Data cloud, it is paramount to ensure the quality of data. In this work, we focus on one of the central hubs of the Linked Data cloud, DBpedia. In particular, we assess the quality of DBpedia for domain knowledge representation. Our results show that DBpedia has still much room for improvement in this regard, especially for the description of concepts and their linkage with the DBpedia ontology. Based on this analysis, we leverage open relation extraction and the information already available on DBpedia to partly correct the issue, by providing novel relations extracted from Wikipedia abstracts and discovering entity types using the dbo:type predicate. Our results show that open relation extraction can indeed help enrich domain knowledge representation in DBpedia.

TYPE OF PAPER AND KEYWORDS

Regular research paper: linked data, DBpedia, quality, knowledge domain, ontology, open relation extraction, knowledge extraction, semantic annotation, RDF, Wikipedia

1 INTRODUCTION

Linked Data, the latest paradigm for publishing and connecting data over the Web, is a significant step towards the realization of a Web that can "satisfy the requests of people and automated agents to access and process the Web content intelligently" [5]. This evolution is concretized by the development of large knowledge bases such as DBpedia [21], Yago [27] and WikiData [29]. These knowledge bases describe concepts and entities and create links to other available datasets, thus contributing to the emergence of a knowledge graph. In particular, DBpedia is usually considered as the central hub of the linked Open Data cloud (LOD). It aims at extracting an RDF representation from Wikipedia content and interlinking it to other LOD datasets.

This knowledge extraction task relies on automatic procedures. Currently, DBpedia knowledge is mainly extracted from Wikipedia infoboxes[1], which contain semi-structured information. DBpedia defines globally unique identifiers (IRIs/URIs) that represent Wikipedia pages/entities and that can be de-referenced over the Web into RDF descriptions [2]. These RDF descriptions are composed of triples of the form `<s,p,o>`, where `p` represents a relation (or a predicate) between entities `s` and `o`. For instance, DBpedia contains the following triples describing the entity *dbr:Canada:*

[1] https://en.wikipedia.org/wiki/Help:Infobox

<dbr:Canada dbo:currency dbr:Canadian_dollar>
<dbr:Canada dbo:capital dbr:Ottawa>
<dbr:Canada dbo:populationTotal 35,985,751>
<dbr:Canada owl:sameAs geodata:Canada>

DBpedia (in its 2016 English version) describes 6 million things, with 1.3 billion triples extracted from the English edition of Wikipedia[2]. Alongside the knowledge base (A-box), DBpedia is also based on an ontology that is manually created by the community to ensure its quality. This ontology contains 754 classes. Among the 6 million things described in DBpedia, 5.2 million are classified in this ontology.

In our previous work [13], we have established that DBpedia lacks terminological knowledge (T-box), especially for domain knowledge. We highlighted some quality issues in the description of domain concepts on a small subset of DBpedia, and we demonstrated a lack of linkage between the DBpedia ontology and the knowledge base. We showed that this lack of linkage is especially true for resources that describe a domain concept, such as planet, village and integer (respectively in the domains of astronomy, geography and mathematics). Without a correct and reliable schema, instances are of limited interest, especially when dealing with big data: It becomes difficult or impossible to detect incoherencies, to reason, or to answer complex queries that go beyond stated triples. In the case of DBpedia, the T-box (schema level) is represented by an ontology that is manually created by the community. This manual work ensures its quality. However, a good linkage between the T-box and the A-box is also paramount to ensure DBpedia quality and its knowledge inference capabilities.

In this paper, we first extend the quality assessment conducted in our previous work [13] by (1) studying 11 new domains, including 8 chosen randomly; (2) using semantic annotation to further extend these domains; (3) evaluating the usage along with the description of domain concepts and their linkage to an ontological schema. In this new quality assessment, we confirm the lack of important triples in the description of domain concepts and the poor linkage among domain concepts in general, and with the ontology in particular, even in domains that are "well represented" in the ontology. Secondly, we propose a solution to help alleviate these issues using semantic annotation [19] and open relation extraction (ORE) [11]. In this work, we use ReVerb [12], one of the available ORE tools, to extract relations from Wikipedia abstracts. Each relation is a triple, much like an RDF triple, except that its elements are not URIs, but instead words or groups of words extracted from text, such as `<The Milky Way, is, a galaxy>`. We associate both the subject and object to DBpedia URIs using a semantic annotator, and classify relations into groups, each corresponding to several possible existing predicates.

Overall, we attempt to answer the following research questions:

Q1: How are domain concepts described in the DBpedia knowledge base, i.e. what are the links relating a concept to other DBpedia concepts (describing the concept), both at the schema level (DBpedia ontology) and at the instance level (DBpedia facts)?

Q2: How are domain concepts used in the DBpedia knowledge base, i.e. what are the links relating DBpedia concepts to domain concepts (using these domain concepts), both at the schema level (DBpedia ontology) and at the instance level (DBpedia facts)?

Q3: What types of predicates appear in the description and usage of domain concepts, and which of them can be used for inferring domain knowledge?

Q4: Can we enhance DBpedia, by extracting novel relations between domain concepts, and by identifying potential new classes, using open relation extraction on Wikipedia abstracts?

The rest of the paper is organized as follows. The following section describes the state of the art in quality assessment and concept and relation identification. Section 3 presents the terminology used in this paper. We describe our overall research methodology in Section 4 and present our results in Section 5. Section 6 presents our work using open relation extraction. Sections 7 discusses in details our findings and Section 8 concludes this paper.

2 RELATED WORK

General Linked Open Data quality. In the first part of this paper, we provide an analysis of the quality of DBpedia for the description of domain knowledge. Several research works have been performed to assess the quality of linked open datasets in general. The usual consensus about the quality of a dataset is its "fitness for use" [18]. In our case, it means "fitness for finding and using knowledge related to a domain". More specifically, when it comes to linked open data, several quality factors have been established: Bizer [6] points out that quality must be assessed according to the task we want to accomplish, and provides 17 quality dimensions and related metrics organized in 4 categories.

Later, Zaveri et al. [33] provide an updated and extensive list of available metrics. Among this list, our work can be related to aspects of the metric "detection of good quality interlinks". However, the fitness of DBpedia for domain knowledge inference is a very specific problem, and does not fall into any of the

[2] http://wiki.dbpedia.org/dbpedia-version-2016-04

established categories, hence our need to introduce novel metrics in this paper. Other quality factors have been defined in [9] and [16], and some frameworks exist to assess the quality of a given dataset. For instance, Luzzu [9] provides a framework customizable by domain experts, and Sieve [23] provides ways to express the meaning of "quality" for a given dataset and a specific task. To the best of our knowledge, there is not any other work that focuses on the quality of domain knowledge representation in DBpedia.

DBpedia quality. The DBpedia knowledge base is a huge dataset containing information on many domains [4], [7]. However, the current method to automatically extract DBpedia data from Wikipedia is based mostly on infoboxes [7]. Even though this method has obvious advantages in terms of automatization and ensures wide coverage, it also poses some issues. According to a user-driven quality evaluation done by Zaveri et al. [32], DBpedia has indeed quality problems (around 12% of the evaluated triples have issues), that can be summarized as follows: Incorrect/missing values, incorrect data types and incorrect links. Kontostas et al. [20] provide several automatic quality tests on LOD datasets based on patterns modeling various error cases, and detect 63 million errors among 817 million triples. Mendes et al. [23] also point out issues in completeness, conciseness and consistency in DBpedia.

In our previous work [13], we showed that domain concepts are often poorly described in DBpedia. We also pointed out at the low number of concepts with a (rdf) type, which is a crippling problem for the knowledge inference capabilities of DBpedia. All these issues can take origin in the extraction framework of DBpedia, the mappings wiki (which is used to create automatically the DBpedia triples), or Wikipedia itself. Some efforts have been made to locate and fix errors in DBpedia, and the Linked Data in general, using crowdsourcing approaches [1]. A crowdsourcing approach could be applied to domain knowledge quality assessment in DBpedia. Howver, given the size of DBpedia, our goal is to explore automatic methods for such a task.

Semantic annotation. Semantic annotation consists in tagging important words or groups of words in a text (entity mentions) in order to generate metadata. This process covers several aspects of text comprehension, such as named entity recognition [3], concept identification [8], sentiment analysis [22], or relation extraction [14][25][28][30]. The efficiency of these tools depends on many factors, such as the task, the type of text and the number of texts available in the corpus [14]. In this paper, given a source concept's abstract, we exploit the concept identification capabilities of semantic annotators to measure, for a given domain concept, the coverage of the Wikipedia abstract by its DBpedia RDF description and to identify concepts that should appear in relation to this domain concept.

Open relation extraction: Introduced by Banko et al. [11], open information extraction (OIE) is a paradigm to extract a large set of relational tuples without requiring any human input. We have witnessed in the past decade the development of several open relation extractors [10][12][31], and some concrete uses are emerging, such as reading news feed to quickly detect economic events [17]. The open relation extraction has recently witnessed improvements based on the usage of external sources from the Web [24] and joint inference [15][26]. In our work, we use the ORE capabilities to bridge the gap between the textual knowledge of Wikipedia and the formal RDF relations in DBpedia. For this task, we used the ReVerb system [11].

3 TERMINOLOGY

In this section, we define the terminology used in this paper.

Entity / Concept: An entity represents a resource or an individual in DBpedia that has a physical reality, such as a person, company or geographic place. At the opposite, a concept is an abstract idea such as "Arithmetic", "Orbit" or "Algorithm".

Class: A class is a set of elements described by common characteristics. For instance, *dbo:City* is a class that contains entities such as Montreal or Ottawa, which are **instances** of the class.

Ontology: An ontology is a formal structure composed of a hierarchy of classes and properties, providing relations between instances of these classes. For instance, we might indicate that every instance of the class *Person* must have exactly one birth place, which is an instance of the class *Place*, and the two must be related by the property *birthPlace*. The ontology is also referred to as the *Schema Level* or *T-Box* (T for Terminology), whereas the instances represent the *Instance Level* or *A-Box* (A for Assertion).

Namespace: In DBpedia, an entity or a concept can be represented in one or both of the following two namespaces:

- The **resource namespace** represents assertions, and corresponds to the **instance (assertional) level** (A-Box). Querying this namespace allows us to identify whether concepts are typed, i.e. whether they are related to some ontology, and whether these concepts are related to other concepts through domain-related properties. Having a concept in this namespace (with an URI of the form *http://dbpedia.org/resource/⟨concept_name⟩*, also abbreviated as *dbr:⟨concept_name⟩*) means that the concept also has a corresponding Wikipedia page (whose location is *http://wikipedia.org/page/⟨concept_name⟩*).

- The **ontology namespace** represents all concepts that have an URI of the form *http://dbpedia.org/ontology/⟨concept_name⟩*, also abbreviated as *dbo:⟨concept_name⟩*. This namespace describes the **schema/terminological level** (T-Box). Unlike the resource namespace, concepts in the DBpedia ontology are not specifically associated with a Wikipedia page and are supposed to represent classes or properties definitions.

Domain: A domain is, informally, "*A specified sphere of activity or knowledge*"[3]. In our approach, a domain D is a set of concepts in a particular subject or field: for instance, the domain "*Mathematics*" contains concepts such as "*Geometry*" and "*Algebra*".

Domain concept: In the Linked Data standards, knowledge is stored in the form of RDF triples *⟨Subject, Relation, Object⟩*. In this work, we consider only triples where the subject and the object represent either a domain concept or a named entity. Domain concepts can represent a class of domain objects, like *Integer* or *Planet*, that are usually defined by restrictions on properties in a formal ontology. They can also represent instances, such as *Saturn*, which is a specific entity in the domain of astronomy, or what is usually called a *topic* or *subject*, such as *Algebra*, in the domain of mathematics.

Concept description: A concept description contains all the triples that comprise the concept in the subject position.

Concept usage: A concept usage contains all the triples that use the concept in the object position.

Wikipedia abstract: Represents the *lead section* of a Wikipedia page, i.e. the section before the table of contents. The word *abstract* comes from the DBpedia property *dbo:abstract*, used to store the contents of the lead section of the associated Wikipedia page. For instance, the object of the triple <dbr:Canada dbo:abstract [text]> is the lead section of the page http://en.Wikipedia.org/wiki/Canada.

4 OVERALL RESEARCH METHODOLOGY

4.1 Approach Overview

In this section, we give an overview of our methodology, which consists of four steps. The first three steps concern the extraction of domain concepts from DBpedia, which are analyzed to determine how well they represent the domain. In the fourth step, we evaluate the potential of open relation extraction and knowledge mining from DBpedia to enrich the representation of domain concepts in DBpedia. These steps are the following ones:

1. **Initial dataset extraction:** In this work, we use Wikipedia Outline pages[4] to identify domain concepts. Such pages provide numerous concepts related to the domain of interest. For instance, the page "Outline of mathematics" contains links with mathematical concepts organized by subject (the subject "Space" contains the concepts "Geometry" and "Topology" for instance)
2. **Domain expansion:** Because of the low number of concepts obtained in the initial dataset extraction step, we expand this set using the Wikipedia abstracts of these domain concepts. Our hypothesis is that the most important concepts present in the abstract are **also part of the domain and should be represented (as objects) in the description of their source concept**. Thus, each source domain concept should be directly related to the concepts identified in its abstract (thereafter called related concepts).
3. **Data extraction from DBpedia:** The next step is to retrieve all the triples in the description or usage of domain concepts, i.e. all triples containing one of the previously identified concepts as subject or object. Unlike our previous study, where we focused exclusively on the description of concepts, we also examine whether the usage of a concept follows the same trend as its description.
4. **Open relation extraction and knowledge mining from DBpedia:** In the last step, we exploit the information contained in the abstracts of domain concepts to identify predicates between the source domain concept and its related concepts and then compare this information with the description of the concept in DBpedia. The extracted relations are either used to confirm the existing links in DBpedia or to learn new predicates.

4.2 Dataset Extraction from Wikipedia

In this section, we explain in detail the first three steps, which result in a dataset of domain concepts and predicates that are analyzed in Section 5.

[3] According to the Oxford dictionary: www.oxforddictionaries.com/definition/english/domain

[4] According to the definition given by Wikipedia: "Outlines on Wikipedia are stand-alone lists designed to help a reader learn about a subject quickly, by showing what topics it includes, and how those topics are related to each other". For example: https://en.wikipedia.org/wiki/Outline_of_mathematics, compared to https://en.wikipedia.org/wiki/Mathematics

4.2.1 Domain concepts identification in Wikipedia

Our set of domains (see Table 1) contains nine domains selected manually, with the objective to select fields as diverse as possible, and eight domains chosen randomly among all the "outline of" pages of Wikipedia.

To identify domain concepts, we extracted all relevant hyperlinks from their associated outline pages. We performed some filtering to remove the obviously 'non-conceptual' pages (e.g. pages describing named entities) using ad hoc rules. Some sections and hyperlinks were systematically removed, such as "*List of...*" (this kind of hyperlink is always used to list entities, and not concepts, e.g. "List of publications", "List of researchers" …), "*Table of…*", "*History*" sections, "*External links*" sections, links describing a country or nationality (e.g. "Greek mathematicians") or named entities (persons, organizations, books…). Following this filtering step, each remaining hyperlink represents a domain concept (a Wikipedia page) that has its counterpart in DBpedia (e.g. the page https://en.wikipedia.org/wiki/Artificial_intelligence is represented by http://dbpedia.org/resource/Artificial_intelligence). A domain is created by listing the set of DBpedia concepts that remain after the pruning stage.

Table 2 shows the number of concepts obtained at the end of this step. On the average, we extracted about 160 concepts per domain (with a median of 97) with the richest domains being *Geography*, *Astronomy* and *Human anatomy*.

4.2.2 Domain concept extraction using semantic annotation

As we can observe in Table 2, the number of concepts extracted from the Outline pages is quite low. For this reason, we expanded the initial set of domain concepts using a semantic annotator. A semantic annotator is a tool that takes raw text as input and identifies segments in the text that represent keywords, concepts or named entities. For each concept in the initial set, we processed its abstract with the Yahoo Content Analysis[5] semantic annotator to obtain the "important concepts". For instance, let us consider the abstract of the concept "Handwriting recognition", where the concepts detected by the semantic annotator are indicated in boldface:

> ***"Handwriting recognition*** *(or HWR) is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices. The image of the written text may be sensed "off line" from a piece of paper by optical scanning (**optical character recognition**) or **intelligent word recognition."***

Table 1: Selected domains

Selection	Domains
Manual	Artificial intelligence; Mathematics; Botany; Astronomy; Biology; Human anatomy; Music theory; Political science; Sports science
Random	Business; Construction; Geography; Health sciences; Industry; Literature; Psychology; Religion

Table 2: Number of concepts per domain based on the "outline of" pages

Domain	Number of concepts	Domain	Number of concepts
A.I.	120	Health sciences	105
Mathematics	60	Industry	100
Botany	85	Literature	139
Music theory	63	Psychology	91
Political science	59	Religion	92
Sports science	99	Astronomy	315
Business	92	Biology	97
Construction	66	Human anatomy	870
Geography	251	**Total**	**2704**

We hypothesize that those concepts are part of the **same domain** as the initial concept. We included those novel concepts in their respective domain. Table 3 provides the number of concepts in each domain after the expansion step, in the resource and ontology namespaces.

In total, we obtained 6834 domain concepts associated with a page in the *resource* namespace. We can notice in table 3 that very few of these concepts, only 100, are represented as classes in the DBpedia ontology.

[5] https://developer.yahoo,com/contentanalysis

Table 3: Number of concepts per domain after expansion

Domain	Number of concepts	
	Resource	Ontology
A.I.	352	0
Mathematics	154	0
Botany	153	1
Music theory	188	3
Political science	110	3
Sports science	245	6
Business	264	5
Construction	151	5
Geography	585	37
Health sciences	244	3
Industry	261	1
Literature	342	5
Psychology	251	2
Religion	206	1
Astronomy	880	8
Biology	350	11
Human anatomy	2098	9
All	**6834**	**100**

4.2.3 Data Extract from DBPedia

We ran a series of SPARQL queries to extract DBpedia triples that refer to our domain concepts along with a predicate of interest. Predicates of interest include:

Description Logic (DL) predicates, which are useful for inference, such as *rdfs:subClassOf* or *rdf:type*, and contain most of the predicates of the RDF, RDFS and OWL vocabularies. We also included the predicate *dbo:type* in this group, as we observed that its usage is similar to *rdf:type*.

Domain predicates, which belong to the domain of interest. For instance, the predicate *dbo:symbol* belongs to the domain *Mathematics*. The most used predicates of this group in our dataset are *dbo:genre*, *dbo:country* and *dbo:class*. Typically, we expect *DL* predicates to provide structural and domain-independent links (*Planet* ***rdfs:subClassOf*** *Astronomical_object*), whereas *domain* predicates provide domain links (*Planet* ***dbo:orbits*** *Star*).

Table 4: Distribution of the extracted triples among namespaces and modes

Namespace	**Nb. Triples**	
	Description	Usage
Resource	146,016	650,773
Ontology	329	462,571
Total	146,345	1,113,344

More specifically, let D be a domain, $DC(D)$ the set of concepts in this domain, and P the set of predicates of interest, i.e. belonging to the DL and Domain groups, as defined earlier. For each concept $c \in DC(D)$, we queried its description and its usage from DBpedia, that is all the available triples involving c in their subject or object, respectively:

$$\mathrm{DESC}(c) = \{\langle c,p,o\rangle \mid \langle c,p,o\rangle \in \mathrm{DBpedia}, p \in P\}$$

$$\mathrm{USE}(c) = \{\langle s,p,c\rangle \mid \langle s,p,c\rangle \in \mathrm{DBpedia}, p \in P\}$$

Informally, the description represents all the information available about a concept, whereas the usage represents the triples where the concept is used to describe another entity or concept. For instance, the description of *Planet* may contain the information that a planet is an astronomical body, or that a planet can be rocky or gaseous. The usage of *Planet* may indicate that the Earth is a planet, or that a moon must orbit a planet.

We refer to the first set (*DESC)* as the *description mode*, and the second (*USE*) as the *usage mode*. In total, we extracted 1,259,689 triples, distributed between namespaces and modes (description, usage) as shown in Table 4.

Here, we can already notice that triples are not equally distributed: the *usage* mode contains approximately 7.5 times more triples than the *description* mode. This difference is even more noticeable in the *ontology* namespace, with more than 1400 times more triples in the *usage* than in the *description*. This is consistent with the fact that the ontology is supposed to be widely **used** in the DBpedia knowledge base, but only **described** with few other elements of the ontology. An example of such a descriptive triple is <dbo:Galaxy rdfs:subClassOf dbo:CelestialBody>.

5 ANALYSIS OF DOMAIN CONCEPTS IN DBPEDIA

In this section, we assess the quality of the representation of domain concepts (description and usage) in DBpedia. In our analysis, we consider that the most important characteristics of a domain for knowledge inference purposes are the following ones: Domain concepts should be described by triples that relate them to other domain concepts in DBpedia, and these related concepts should represent classes from the ontology and concepts (instances) of the same domain. Subsections 5.1 to 5.3 present the three metrics used to analyze the *DL* and *domain* predicates (and hence triples) in the dataset. Subsection 5.4 presents a finer analysis of the *DL* group.

5.1 Predicates' global frequency

In this first step, our goal is to obtain global results to determine how the triples are distributed among namespaces, modes, and domains.

Given a predicate *p* and a concept *c*, we define the **frequency** *f(p,c)* as the total number of triples involving *p* and *c*, either in the description or in the usage of c, i.e. <*s*, *p*, *c*> and <*c*, *p*, *o*>. By extension, we also define $f(G, c)$, the frequency of a group *G* (where *G* is one of the two groups DL and Domain, as defined in section 4.2.3) for a concept *c*, as the sum of the frequencies of all predicates of *G* for *c*, and the **global frequency of *G*** by the sum of $f(G, c)$ on all the concepts of our dataset. For instance, the *DL* group has a global frequency of 136,605 in the description mode. This means that 136,605 triples that describe a concept in our dataset use a *DL* predicate.

Figure 2 shows the distribution and global frequency of predicates' groups for both modes in the resource namespace. The ontology namespace statistics are not shown, since all the predicates in the description or usage of a concept in this namespace belong to the *DL* group. There is an important difference in the predicates distribution in each mode. In the *description* mode, *domain* predicates are very few compared to *DL*, whereas in *usage* mode, they are almost equally balanced but far more numerous. We can conclude that *domain* concepts are widely **used** in DBpedia in relation with domain predicates, but that they themselves seldom exploit this group in their **description**.

To refine these observations, we introduce the measure of *concept coverage*, which aims at analyzing the behavior of all predicates of the group, in a given domain. We calculate, for each predicate **that represents at least 10% of the occurrences of the group**, the proportion of domain concepts in whose description or usage the predicate appears, and then average this value on the cardinality of the group.

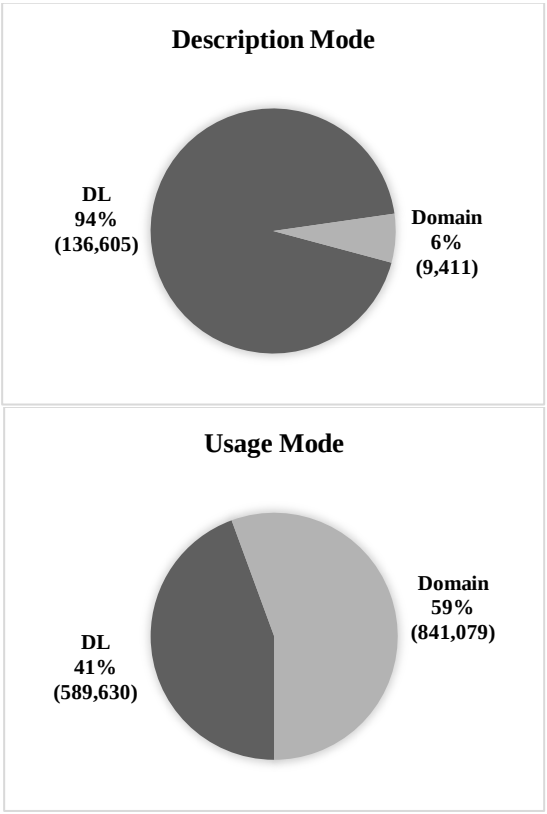

Figure 2: Distribution and global frequency of predicates in the resource namespace

$$G_{10\%}=\left\{p\in G \mid \sum_{c\in DC(D)} f(p,c) > 0.1 * \left(\sum_{c\in DC(D)} \sum_{p\in G} f(p,c)\right)\right\}$$

A predicate that belongs to $G_{10\%}$ is called a **main predicate** of the group G. We introduce this selection because each group contains a small subset of widely used predicates (typically, *DL* predicates: *rdf:type*, *owl:sameAs* and *dbo:type*) and one or more other predicates that seldom appear, meaning that, when calculating the average, an erroneous predicate that appears only a couple of times would reduce drastically the result. By taking into account only the main predicates of a group *G*, the concept coverage for a domain *D* is defined in the following way:

$$CCov(G, D)=\frac{1}{|G_{10\%}|}\sum_{p\in G_{10\%}} \frac{|\{c\in DC(D) \mid f(p,c)>0\}|}{|DC(D)|}$$

This means that, for example, if a group G has a concept coverage of 0.15 for a given domain, on average, a main predicate of the group is used in the description of 15% of the DCs.

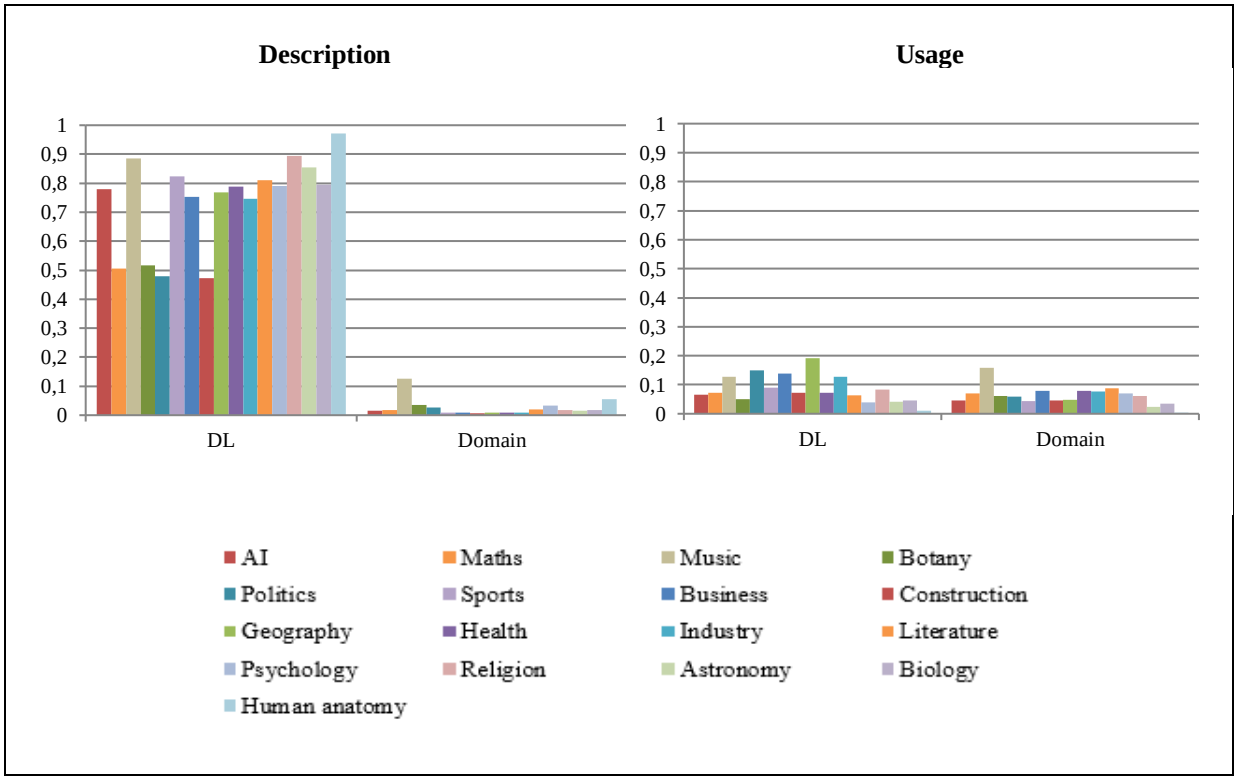

Figure 3: Concept coverage for the *DL* and *Domain* groups per domain in the resource namespace

Figure 3 shows the concept coverage values for all domains. These results confirm the observations made at the beginning of the section: *DL* predicates are widely used in the description of concepts, regardless of the domain, whereas *domain* predicates appear very rarely. The novel information, however, is that each individual *domain* predicate appears in a very low number of concepts: On average, each *domain* predicate appears in less than 6% (except in the domain Music) of the concepts' **description** and less than 15% in their **usage**.

5.2 A closer look at the DL group

As we mentioned previously, the *DL* group contains 3 main predicates that represent the majority of all the triples: *rdf:type*, *owl:sameAs* and *dbo:type*. In this section, we look more finely at the usage of this group.

Concerning the DBpedia resources' description (**Q1**), the predominant predicates are *owl:sameAs* and *rdf:type*, used respectively to indicate an URI that describes the same entity or concept, and to provide a type relation with the ontology, such as <*dbr:Barack_Obama rdf:type dbo:Person*>. These two predicates represent respectively 54.6% and 45.1% of the *DL* group in this namespace (resource) and mode (description). Only the *rdf:type* predicate is of interest here, as *owl:sameAs*'s only potential usage for knowledge inference is to indicate an equivalent resource in another LOD set, and we focus only on DBpedia in this paper.

To assess the capabilities of DBpedia for knowledge inference by using *rdf:type*, we want to know the proportion of DBpedia resources that are typed, and the origin of the type, as the object of the *rdf:type* triple can be either in the DBpedia ontology, or in another dataset. Figure 4 provides the distribution of concepts that have a type in various LOD datasets. In section 5.2, we mentioned that almost every concept uses a *DL* predicate. However, as we can notice here, many concepts are still un-typed: Depending on the domain, only 2 to 48% have a type in the DBpedia ontology, and only 25 to 74% have a type overall. On the average 81% of the concepts do not have a type in the DBpedia ontology and 55% do not have a type at all.

Concerning the resources' usage (**Q2**), the dominant predicate is *dbo:type*, representing 97.4% of the *DL* group in this namespace and mode (30,934 occurrences among 31,765). This predicate appears in the usage of 539 concepts, an average of 57.4 occurrences per concept. The way this predicate is used in DBpedia suggests that its semantics is very similar to *rdf:type*, since its object is almost always something that could be considered as a class (such as *dbr:Village*, *dbr:Town*, *dbr:Lake*). Therefore, it could be used to answer our **Q4** by identifying potential classes. We consider this in Section 6.

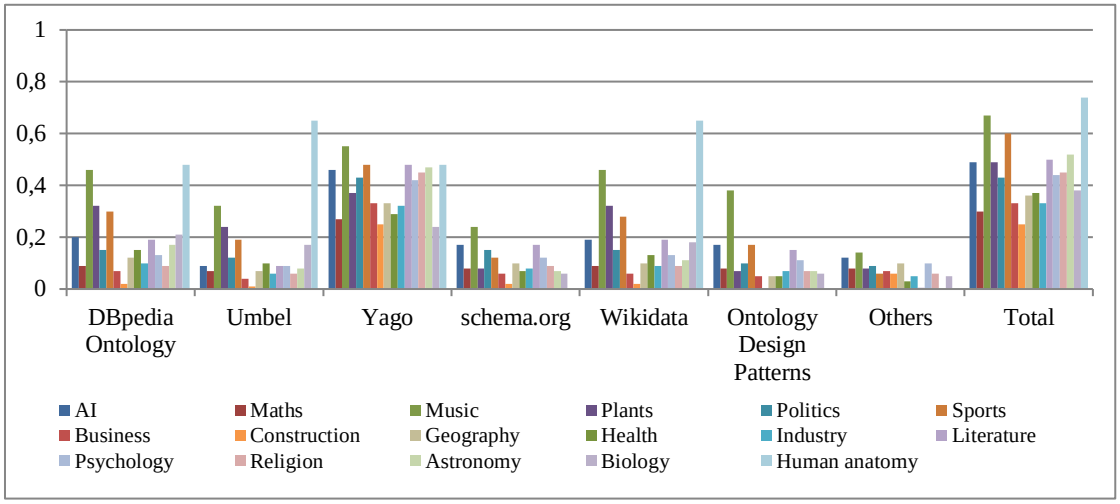

Figure 4: Typing of concepts in various Linked Open Datsets

Table 5: Ratio links / number of concepts

Domain	Resource namespace		Ontology namespace	
	Links to a resource	Links to ontology	Links from a resource	Links to ontology
Artificial intelligence	0.20	0.00	0.00	0.00
Mathematics	0.19	0.00	0.00	0.00
Music theory	2.85	0.01	0.33	0.00
Botany	0.14	0.00	0.00	0.00
Political science	0.35	0.02	0.67	0.00
Sports science	0.29	0.21	8.50	0.00
Business	0.17	0.03	1.80	0.20
Construction	0.07	0.01	0.40	0.00
Geography	0.22	0.05	0.81	0.14
Health sciences	0.33	0.11	8.67	0.00
Industry	0.18	0.00	0.00	0.00
Literature	0.25	0.38	25.80	0.20
Psychology	0.51	0.00	0.00	0.00
Religion	0.70	0.00	0.00	0.00
Astronomy	0.46	0.14	15.50	0.13
Biology	0.29	0.18	5.82	0.36
Human anatomy	1.98	0.77	179.78	0.11
Average	0.54	0.13 (*)	16.54 (*)	0.08 (*)

(*): When the *ontology* namespace is concerned, the average is calculated only on the 15 (out of 17) domains that have at least one concept in the ontology

Concerning the *ontology* (**Q1-3**), the *DL* group is mostly used to create the ontological structure using *rdfs:subClassOf, owl:equivalentClass* and *owl:disjointWith* among classes (both in their description and usage), and *rdf:type* between resources and classes. Each class in our dataset has on average 4769 instances, represented by *rdf:type* links.

5.3 Concepts Linking Among Domains

In the previous sections, we studied the linkage between domain concepts and other DBpedia resources. In this section, we focus on the links **between concepts in the same domain**.

There are three possible types of links: resource to resource (6101 links), resource to ontology (2056 links) and ontology to ontology (13 links). Table 5 provides the average number of links per concept (total number of links/total number of concepts) in each namespace and domain. In the first two columns we give the average number of outbound links per domain concept in the *resource* namespace (a link may point to another resource or a concept in the DBpedia ontology). The last two columns concern domain concepts that are in the *ontology* namespace. Note that the third column considers the inbound links, since it is the kind of link we expect to find between a resource and a class in the ontology.

As we can see, apart few exceptions, the number of links is quite low. In a well-described domain, we would expect at the very least one link to another concept of the same domain, which is the case here only for Music theory (on average 2.85 links with another resource) and Human anatomy (on average 1.98 links with another resource). Concerning the resource-to-ontology links, the situation is even worse: among more than one million triples, there are only 13 links to the 100 DBpedia classes found in our dataset (an average 0.13 links per class). Each domain concept that is present in the ontology is linked to an average of 16.5 resources (instances) of the same domain. Since each class has 4769 instances on average, this number is very low. It means that only 0.3% each class instances are in the same domain (16.5 out of 4769).

5.4 Summary of the Results

In this section, we highlighted three weaknesses in the conceptual and domain-related knowledge inference capabilities of DBpedia:

1. Poor description of DBpedia resources in general, with almost no presence of domain-related predicates to describe concepts (Section 5.1 and 5.2);
2. Poor linkage between the DBpedia T-box and A-box, with very few (2 to 48%) concepts that are actually typed in the DBpedia ontology (section 4.3);
3. Very few links between concepts of a same domain (section 4.4).

6 DBPEDIA ENRICHMENT

In this section, we propose two methods to correct these limitations. This first method relies on open relation extraction (Section 6.1) for the extraction of predicates from the abstracts associated to our domain concepts in DBpedia. We extract both domain-related predicates (Section 6.2) and *rdf:type* predicates (Section 6.3). The second method consists in analyzing the *dbo:type* predicate used in DBpedia and the hyponymy relations extracted by our first method, to identify potential classes among our domain concepts (Section 6.4).

6.1 Open Relation Extraction

Open relation extraction consists in extracting segments that express a relation from texts, without any predefined and limited set of relations. In our experiment, we ran the open relation extractor ReVerb on the Wikipedia abstracts of every concept in our dataset.

Given the following input text (the first sentence of the abstract of the concept *Handwriting recognition*):

> *"Handwriting recognition (or HWR) is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices."*

ReVerb extracts two relations:

```
⟨Handwriting recognition;
    is the ability of; a computer⟩

⟨the ability of a computer;
    interpret; intelligible handwritten input⟩
```

We lemmatized the subject, the object and the relation based on Stanford CoreNLP[6] and removed determiners from the subjects and objects, to obtain a format similar to DBpedia URIs (no plural, no article, etc.). The lemmatized forms of the two previous relations are the following ones:

[6] http://stanfordnlp.github.io/CoreNLP/

```
⟨handwriting_recognition;
   be_the_ability_of; computer⟩

⟨ability_of_a_computer;
   interpret; intelligible_handwritten_input⟩
```

Based on the set of relations extracted from all the abstracts, we only keep the relations for which **both the subject and the object are among the previously identified domain concepts**. In the example, we keep only the first one, since *Computer* is a recognized concept whereas *intelligible_handwritten_input* is not.

There were 382 unique relations extracted by ReVerb, but most of them (329) appear only once, mostly because they are very specific ("is any set of", "are very tightly bound by"), or sometimes because they are erroneous, with the inclusion of punctuation, for instance ". There are various types of". Table 7 gives the most frequent relations (at least 5 occurrences) and, for each one, the number of occurrences.

We manually classified the relations into the following categories and their associated predicates:

- **Equivalence relations:**
  owl:sameAs / owl:equivalentClass
- **Mutual exclusion relations:**
  owl:differentFrom / owl:disjointWith
- **Hypernymy/hyponymy relations:**
  rdf:type / rdfs:subClassOf / dbo:type
- **Domain relations:** Used as default if none of the preceding categories is selected.
- **None:** when the extracted relation is erroneous or nonsensical.

To perform the classification into categories, we asked four computer science Master's students at École Polytechnique de Montréal to assess the relations extracted by ReVerb. Each one assigned a category to every relation. The final category of each relation was selected by performing a majority vote. In case of equality, we asked a fifth evaluator to choose.

Table 7: Most frequent relations extracted by ReVerb in our dataset

Predicate	Frequency
is	91
is a branch of	43
is the branch of	21
is a type of	11
includes	9
is a form of	8
is a subfield of	5
is an artery of	5
is a genre of	5

Table 8 indicates the categories in which we classified the most frequent relations. We can observe that the majority of occurrences are domain relations, followed by hypernymy relations. As mentioned previously, this distribution is the result of a vote among four evaluators. The Fleiss' kappa on this evaluation is 0.59, with a 95% confidence interval of [0.57, 0.61], representing a moderate / strong agreement.

6.2Extraction of Domain-Related Predicates

In this section, we are interested in determining if the extracted relations are already represented in DBpedia in the resource namespace (Links with ontology namespace are discussed in next section). This enables us to evaluate how open relation extraction may contribute to enrich DBpedia with new triples.

To accomplish this task, the first step is to look for triples in DBpedia relating domain concepts pairs extracted by ReVerb. For instance, based on the relation ⟨*Robotics; focuses on; Robots*⟩ extracted from the abstract of "robotics", we note that the pair (*Robotics, Robot*) is linked through the triple ⟨*dbr:Robotics, rdfs:seeAlso, dbr:Robot*⟩ in DBpedia.

In the second step, we manually assessed if the extracted relations between concepts' pairs provide, at least partially, some novel information compared to the triples already in DBpedia. In the previous example, it is the case, as *rdfs:seeAlso* only indicates that the two concepts are somehow related, whereas the relation "*focuses on*" points out that *robots* is a central concept in *robotics*, providing some novel information.

Table 9 provides, for each domain, the number of novel relations, and their proportion among all extracted relations. We also provide in Table 10 the number of novel relations per category, for all domains together.

We can note that most of the extracted relations **are not represented** in DBpedia (all the ratios are close to 1, meaning that almost all extracted relations are novel). In 8 domains out of 17 (that is, where the ratio is equal to 1), DBpedia does not contain any triples between the concept pairs extracted by ReVerb. Out of the 631 extracted relations (excluding the 10 "invalid" relations, i.e. the group "none" in table 8), only 27 are represented in DBpedia (4%), and all of them are of the *Domain* category. This shows that most relations are indeed novel in DBpedia, and that open relation extractors are a suitable technology to generate new domain knowledge.

Table 8: Distribution of the most frequent relations

Relation category	Relations	Total number of occurrences
Domain	is an artery of; has; is the scientific study of; is the study of; is an approach to; arises from	347
Equivalence	is the equivalent of; is sometimes referred to as; is often used synonymously with; is also known as; is often called; is known as	10
Mutual exclusion	is neither; is distinguished from; is not to be confused with; is different from; is not synonymous with	9
Hypernymy	is; is a type of; are examples of; is a certain kind of; is a particular pattern of; is sometimes classified as; is the type of	254
Hyponymy	includes; consists of; can include activities such as	11
None	is substantially altered.It is difficult to find absolutely; are dwarf; There are various types of	10

Table 9: Number of novel relations

Domain	Extracted relations	Novel relations	Ratio
Artificial intelligence	35	34	0.97
Astronomy	144	143	0.99
Biology	3	3	1.00
Botany	24	22	0.92
Business	21	21	1.00
Construction	10	10	1.00
Geography	52	50	0.96
Health sciences	42	41	0.98
Human anatomy	126	110	0.87
Industry	9	8	0.89
Literature	34	34	1.00
Mathematics	32	32	1.00
Music theory	27	26	0.96
Political science	10	10	1.00
Psychology	39	37	0.95
Religion	18	18	1.00
Sports science	5	5	1.00
All domains	**631**	**604**	**0.96**

Table 10: Number of novel relations

Category	Extracted relations	Novel relations	Ratio
Hypernymy	254	254	1.00
Hyponymy	11	11	1.00
Mutual exclusion	9	9	1.00
Equivalence	10	10	1.00
Domain	347	320	0.92
None	10	N/A	N/A

Table 11: Number of hyponymy relations for which the subject is in the ontology and hypernymy relations for which the object is in the ontology

Domain	Nb. relations extracted	Links with the ontology	
		Subject	Object
Artificial_intelligence	35	0	0
Astronomy	144	3	3
Biology	3	0	0
Botany	24	0	3
Business	21	1	0
Construction	10	0	0
Geography	52	2	3
Health_sciences	42	0	12
Human_anatomy	126	0	2
Industry	9	0	0
Literature	34	3	3
Mathematics	32	0	0
Music_theory	27	0	0
Political_science	10	0	0
Psychology	39	0	0
Religion	18	0	0
Sports_science	5	1	0
Total	631	10	26

6.3 Extraction of rdf:type Links

In this section, our objective is to assess if the Open Relation Extraction paradigm can be used efficiently to relate DBpedia resources with the DBpedia ontology. For each relation, we queried DBpedia to find if the subject or the object has a corresponding concept in the ontology. For instance, given the relation ⟨*flat_bone, is, bone*⟩, we find that the class *dbo:Bone* exists in the DBpedia ontology and we know that *flat_bone* already exists in the DBpedia resources *dbr:Flat_bone*. Thus *dbo:Bone* should be related to the entity *dbr:Flat_bone* through an *rdf:type* link (since "is" designates a hypernymy relation). If this link is not present in DBpedia, our approach highlights that it should be.

Table 11 provides the number of concept pairs present in the ontology where the relation represents hypernymy or hyponymy. In the 36 cases where a correspondence is found (out of 631), only the subject or the object is mapped to the ontology and never both.

An important point is that **all these relations are novel**. We highlighted before the lack of linkage between the A-box and the T-box in DBpedia, and especially the poor typing of domain concepts. We prove here that ORE tools are relevant to partly correct this issue. An example of such an extracted relation is <*Milky Way, is, galaxy*>, allowing us to infer that <*dbr:Milky_Way rdf:type dbo:Galaxy*>, which is not present in DBpedia. We manually assessed the extracted relations and concluded that for 14 out of 36 cases there is indeed an instance/class relationship between the concepts that is not represented by a *rdf:type* in DBpedia.

6.4 Domain Class Identification

In this section, we present an approach to identify domain concepts that represent classes, but that do not appear in the DBpedia ontology. To accomplish this, we propose two methods. The first one is based solely on the information present in DBpedia, more specifically on the predicate *dbo:type*. The second uses the hypernymy relations extracted by ReVerb.

6.4.1 Identification by *dbo:type*

In this approach, we hypothesize, based on our observation of its usage, that the *dbo:type* predicate has a similar role to *rdf:type*, i.e. to indicate an instance/class relationship between two DBpedia entities. Therefore, the object of such a predicate is potentially a class. For example, if we have the triple <*dbr:Seattle dbo:type dbr:City*>, we consider that *dbr:City* is a potential class, even though it is not present in the ontology.

Table 12. Results of the evaluation for the *dbo:type*-based method

Result	Accepted	Refused	Questionable
Number of concepts	112	66	18
Percentage	57%	33%	9%

In our dataset, we identified 539 potential classes (that are the object of at least one *dbo:type* triple), with an average of 54.24 instances per potential class. However, 196 among the 539 potential classes have the biggest number of instances (at least 5 instances) and represent more than 95% of the occurrences. Because we conducted a manual evaluation of whether these candidates are indeed classes, we focus on these 196 potential classes.

We relied on a vote between four evaluators, who assessed the validity of each of those 196 candidates. The Fleiss' kappa for this evaluation is 0.43, with a 95% confidence interval of [0.40, 0.46], representing a moderate agreement.

Table 12 provides the results of this vote. A candidate can be accepted (it is a class that should be in the ontology), refused (it is not a class) or questionable (for instance, *Research* can be considered as a class, but the *dbo:type* triples present in DBpedia are nonsensical, such as <*dbr:University_of_Oregon dbo:type dbr:Research*>).

As we can see, this method yields moderately good results, with a precision of 57% (66% when we also consider the questionable classes).

6.4.2 Identification by Hypernymy Relations

In this method, we exploit the relations extracted in Section 5.1. In our classification of the extracted relations, we determined that some of them represented hypernymy links. Because of the nature of such links, the object is a potential candidate class. We extracted 254 hypernymy relations. Some have the same object, leading to a total of 143 candidates.

Following the same approach, we evaluated each candidate to assess if it should be a class by performing a vote between four evaluators. The Fleiss' kappa for this evaluation is 0.59, with a 95% confidence interval of [0.55, 0.63], representing a strong agreement.

Table 13 provides the results of this evaluation. Like before, a candidate can be accepted, refused or questionable.

Table 13: Results of the evaluation for the ORE-based method

Result	Accepted	Refused	Questionable
Number of concepts	93	20	30
Percentage	65%	14%	30%

This second method yields better results than the first one, with a precision of 65%. Besides, there is a low number of firm refusals (14%), with 30% of questionable cases. These cases represent candidates that could arguably be classes depending on the context, and therefore the precision in practice could be as high as 86%.

Overall, the first method, based on *dbo:type*, provides 112 concepts that should be classes out of 196 candidates. The second method, based on ORE, provides 610 novel relations and identifies 93 concepts that should be classes.

7 DISCUSSION

In this section, we refer to the elements highlighted previously in order to answer our research questions, presented in Section 1.

7.1 Assessing the Quality of Domain Knowledge in DBpedia (Q1-Q3)

The first three research questions concern three aspects of the quality of domain knowledge. Q1 and Q2 ask whether domain concepts are well *described* and *used* in DBpedia respectively, whereas Q3 concerns the predicates that are present in the description and usage of domain concepts.

In Section 4, we confirmed some of the conclusions drawn in our previous work [13]. Even for the domains that are the most represented in the DBpedia ontology (Astronomy, Biology, Geography, Human anatomy), we noticed a serious lack of connection between the ontology and the resources, with only 48% of concepts typed in the DBpedia ontology in the best case, and 2% in the worst. We also noticed that concepts are much more used than they are described. This means that, when exploring DBpedia as a graph, many concepts represent "Domain sinks", i.e. nodes with only inbound *Domain* links (Q1, Q2). We also noticed a disparity in the *domain* (i.e. *dbo*) predicates: Some of them are much more used than others, to the point where some predicates only appear once in the entire dataset, such as *dbo:governor* or *dbo:musicBy* (Q3). We have not investigated this further, as this is not the point of this paper, but we suspect that there could be room for improvement here. For instance, the predicate *dbo:musicBy* appears only 1,402 times in all of DBpedia and could be replaced in most cases by the predicate *dbo:musicComposer* (62,034 occurrences).

Concerning the linking among concepts of the same domain in the *resource* namespace, we confirmed the extremely low number of links (less than 1 per concept to another concept in the same domain, for all but two domains). There is also a low number of links towards domain concepts present in the ontology: Even though between 25 and 74% of concepts are typed (depending on the domain), only 13% on average are typed **within** the domain. The conclusion that DBpedia lacks domain knowledge is however tempered by the fact that our method to create domains is still incomplete and probably misses many concepts, which should be in the domain.

Another important point concerns the ontology. We already knew from our previous work that the DBpedia ontology is poorly linked to the domain concepts. In this study, we noticed a new crucial point: There are several classes in our dataset (33 out of 100), which appear in the ontology and have **no instance at all**, like "psychologist" or "law". Unlike most of our other conclusions, this lack of linkage applies to all DBpedia resources, and not only to our relatively small set of domain concepts: These 33 classes do not have any instance in **all of DBpedia**. Given the small size of the DBpedia ontology as a whole (685 classes[7]), these classes still represent 5% of the ontology that is completely unlinked to the A-Box.

However, in all cases, our point is that the *domain* group is almost never present to **describe** concepts. This point is even stronger as this group arguably contains more predicates than it should. Many predicates occur very rarely, indicating a lack of reuse across DBpedia.

7.2 Predicate and Class Discovery Using Relation Extraction and *dbo:type* (Q4)

In the second part of this study, we used open relation extraction to identify relations in Wikipedia abstracts that could enrich the DBpedia description of domain concepts. We also used the particular predicate *dbo:type* and the extracted hypernymy relations to identify potential classes to be added to the DBpedia ontology.

Even if ReVerb did not provide a high number of new relations, we proved that most of the extracted relations were not already present in DBpedia, with only 4% of redundancy. This means that 96% of the extracted relations were entirely novel, or at least provided some novel information compared to the triple(s) already present in DBpedia.

[7] http://wiki.dbpedia.org/services-resources/ontology

We also pointed throughout this paper that the links between resources and the ontology are rare, and that the DBpedia ontology only contains a few domain concepts. Some of the extracted relations could be used to suggest DBpedia resources that should be ontology classes or to provide a type to a resource in the DBpedia ontology (14 relations). One limit of our approach is that these numbers represent only a small proportion of the extracted relations. In fact, a limitation of our work comes from the approach used to identify domain concepts. This method is by no means exhaustive, so we cannot consider that we were able to identify all the concepts relevant to a particular domain. Because we only consider relations where both the subject and object are part of a domain, an enrichment of the recognized domain concepts could help further expand the set of applicable relations.

When it comes to relations between resources, the small number of identified relations can also be considered as a limit of our approach. We have a total of 631 extracted relations that link two domain concepts, for a total of 6835 concepts in our dataset. This represents approximately one new relation for every 11 concepts, or 0.089 relation per concept. This could be mitigated by exploring other open relation extractors or by parsing all Wikipedia texts mentioning concept pairs rather than only the abstract of each domain concept.

Additionally, we have classified the extracted relations into categories that contain at least two predicates (*Mutual exclusion* for instance), and at most a very high number of predicates (*Domain*). This is sufficient for a first coarse-grained analysis of the results. However, a finer-grained analysis would be to associate the extracted relations to predicates automatically. This is left for future work.

Concerning the potential classes identification, our two methods obtained respectively a precision of 57% for the first one (with 112 new classes), and 65% for the second (with 93 new classes). However, these results do not take into account the granularity of the DBpedia ontology. Several of our identified classes are probably too precise to be included in the DBpedia ontology as such. One potential idea would be to create several fine-grained domain ontologies related to the upper-level DBpedia ontology.

Altogether, we showed the relevance of open relation extraction for the task of improving DBpedia, both at the assertion-level and at the schema-level.

8 Conclusion and Future Work

In this paper, we confirmed the conclusion drawn in our previous work [13] on a larger set of domains, highlighting the lack of domain knowledge representation in DBpedia, especially at the ontology level. We also enhanced our method to answer the question "What are the concepts that should belong to a given domain?", notably by exploiting the information contained in the abstracts of a small number of reliable concepts. We extended our analysis of the current state of DBpedia by also considering the linkage with the ontology and the usage of concepts. We concluded that improvements are still to be made on DBpedia to represent more extensively the knowledge contained in Wikipedia, essentially for the description of concepts and their linkage to the ontology.

We also proposed a method to exploit Wikipedia abstracts to infer relations between domain concepts. This method proved quite effective although limited in terms of the number of discovered relations. In parallel, we exploit these relations to discover new classes. This approach proved more effective than the method based on a direct exploration of DBpedia RDF triples.

The approach we propose here is still in development, but already provides interesting results. Our future work will consist of providing automatic methods to classify the extracted relations to compare more finely the redundancy between the results of open relation extraction and the triples already present in DBpedia.

Acknowledgments.

This research has been funded by the NSERC discovery grant program.

References

[1] M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, S. Auer, and J. Lehmann, "Crowdsourcing Linked Data Quality Assessment," *12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013.*

[2] A. P. Aprosio, C. Giuliano, and A. Lavelli, "Extending the Coverage of DBpedia Properties Using Distant Supervision over Wikipedia," *ResearchGate*, vol. 1064, Jan. 2013.

[3] S. Atdağ and V. Labatut, "A Comparison of Named Entity Recognition Tools Applied to Biographical Texts," *2nd International Conference on Systems and Computer Science*, Villeneuve d'Ascq, France, 2013.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A Nucleus for a Web of Open Data," in Proc. *6th International Semantic Web Conference and 2nd Asian Semantic Web Conference*, pp. 722–735, 2007.

[5] T. Berners-lee, D. Connolly, L. Kagal, Y. Scharf, and J. Hendler, "N3Logic: A Logical Framework for the World Wide Web," *Theory Pr. Log Program*, vol. 8, no. 3, pp. 249–269, May 2008.

[6] C. Bizer, *Quality-Driven Information Filtering-In the Context of Web-Based Information Systems*. Saarbrücken, Germany: VDM Verlag, 2007.

[7] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, S. Hellmann: "DBpedia - A Crystallization Point for the Web of Data," *Web Semant*, vol. 7, no. 3, pp. 154–165, Sep. 2009.

[8] C. De Maio, G. Fenza, M. Gallo, V. Loia, and S. Senatore, "Formal and Relational Concept Analysis for Fuzzy-Based Automatic Semantic Annotation," *Appl. Intell.*, vol. 40, no. 1, pp. 154–177, 2014.

[9] J. Debattista, C. Lange, and S. Auer, "Luzzu Quality Metric Language -- A DSL for Linked Data Quality Assessment," *ArXiv150407758 Cs*, Apr. 2015.

[10] L. Del Corro and R. Gemulla, "ClausIE: Clause-based Open Information Extraction," in *Proceedings of the 22Nd International Conference on World Wide Web*, New York, NY, USA, 2013, pp. 355–366.

[11] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld, "Open Information Extraction from the Web," *Commun ACM*, vol. 51, no. 12, pp. 68–74, Dec. 2008.

[12] A. Fader, S. Soderland, and O. Etzioni, "Identifying Relations for Open Information Extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, USA, 2011, pp. 1535–1545.

[13] L. Font, A. Zouaq, and M. Gagnon, "Assessing the Quality of Domain Concepts Descriptions in DBpedia," in *2015 11th International Conference on Signal-Image Technology Internet-Based Systems*, Bangkok, 2015, pp. 254–261.

[14] M. Gagnon, A. Zouaq, F. Aranha, F. Ensan, and L. Jean-Louis, "Semantic Annotation on the Linked Data Cloud: A Comprehensive Evaluation," *Journal of Web Semantics*, submitted-2016.

[15] S. Groppe, D. Heinrich, S. Werner, "Distributed Join Approaches for W3C-Conform SPARQL Endpoints", Open Journal of Semantic Web (OJSW), 2(1), Pages 30-52, 2015.

[16] A. Hogan, J. Umbrich, A. Harth, R. Cyganiak, A. Polleres, and S. Decker, "An Empirical Survey of Linked Data Conformance," *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 14, pp. 14–44, Jul. 2012.

[17] A. Hogenboom, F. Hogenboom, F. Frasincar, K. Schouten, and O. van der Meer, "Semantics-Based Information Extraction for Detecting Economic Events," *Multimed. Tools Appl.*, vol. 64, no. 1, pp. 27–52, 2013.

[18] J. M. Juran and J. A. De Feo, *Juran's Quality Handbook*: The Complete Guide to Performance Excellence, McGraw Hill Professional, Sep 2010.

[19] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff, "Semantic Annotation, Indexing, and Retrieval," *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 2, no. 1, pp. 49–79, Dec. 2004.

[20] D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen, "Test-driven Evaluation of Linked Data Quality," in *Proceedings of the 23rd International Conference on World Wide Web*, New York, NY, USA, 2014, pp. 747–758.

[21] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendesf, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, C. Bizer., "DBpedia – A Large-Scale, Multilingual Knowledge Base Extracted from Wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.

[22] B. Liu, "Sentiment Analysis and Opinion Mining," *Synth. Lect. Hum. Lang. Technol.*, vol. 5, no. 1, pp. 1–167, 2012.

[23] P. N. Mendes, H. Mühleisen, and C. Bizer, "Sieve: Linked Data Quality Assessment and Fusion," in *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, New York, NY, USA, 2012, pp. 116–123.

[24] K. Narasimhan, A. Yala, and R. Barzilay, "Improving Information Extraction by Acquiring External Evidence with Reinforcement Learning," *ArXiv160307954 Cs*, Mar. 2016.

[25] R. Peixoto, T. Hassan, C. Cruz, A. Bertaux, N. Silva, "Hierarchical Multi-Label Classification Using Web Reasoning for Large Datasets", Open Journal of Semantic Web (OJSW), 3(1), Pages 1-15, 2016.

[26] H. Poon and P. Domingos, "Joint Inference in Information Extraction," in *Association for the Advancement of Artificial Intelligence*, 2007, vol. 7, pp. 913–918.

[27] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A Core of Semantic Knowledge," in *Proceedings of the 16th International Conference on World Wide Web*, New York, NY, USA, 2007, pp. 697–706.

[28] M. Tatu, M. Balakrishna, S. Werner, T. Erekhinskaya, D. Moldovan, "A Semantic Question Answering Framework for Large Data Sets", Open Journal of Semantic Web (OJSW), 3(1), Pages 16-31, 2016.

[29] D. Vrandečić and M. Krötzsch, "Wikidata: A Free Collaborative Knowledgebase," *Commun ACM*, vol. 57, no. 10, pp. 78–85, Sep. 2014.

[30] J. Weston, A. Bordes, O. Yakhnenko, and N. Usunier, "Connecting Language and Knowledge Bases with Embedding Models for Relation Extraction," *ArXiv13077973 Cs*, Jul. 2013.

[31] F. Wu and D. S. Weld, "Open Information Extraction Using Wikipedia," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, USA, 2010, pp. 118–127.

[32] A. Zaveri, D. Kontokostas, M. A. Sherif, L. Bühmann, M. Morsey, S. Auer, J. Lehmann, "User-driven Quality Evaluation of DBpedia," in *Proceedings of the 9th International Conference on Semantic Systems*, New York, NY, USA, 2013, pp. 97–104.

[33] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer, "Quality Assessment for Linked Data: A survey," *Semantic Web*, vol. 7, no. 1, pp. 63–93, 2015.

APPENDIX: ACRONYMS

LOD: Linked Open Data. Contains all datasets following W3C standards for Linked Data[8], in which the data is available to the public.

RDF[9]: Resource Description Framework. Specifications proposed by the W3C to describe information in a structured format <subject, predicate, object>. It is the main data model used on the LOD.

RDFS[10]: RDF Schema. Extension of RDF that aims at providing a data-modelling vocabulary. For instance, it allows to indicate that the property *birthPlace* must link a *Person* and a *Place* (i.e. if we have the triple <John, birthplace, Montreal>, *John* and *Montreal* must be a *Person* and a *Place*, respectively).

OWL[11]: Web Ontology Language. A language built on RDFS, but that allows a finer representation of more complex knowledges. Whereas RDF is mostly used to describe *facts*, OWL is instead used to define classes in *ontologies*.

SPARQL[12]: SPARQL Protocol and RDF Query Language. Query language used to retrieve and manipulate RDF data present on the Web. All LOD datasets provide a SPARQL endpoint.

URI: Uniform Resource Identifier. Character string used to uniquely identify a resource, for instance http://dbpedia.org/resource/Canada. In the LOD, everything (entity, predicate, class…) is identified by an URI, which does not necessarily correspond to a webpage.

IRI: Internationalized Resource Identifier. Extension of URIs that allows the use of Unicode characters, such as Chinese, Cyrillic or accentuated characters. It is not currently supported by all LOD implementations.

[8] https://www.w3.org/standards/semanticweb/data
[9] https://www.w3.org/RDF/
[10] https://www.w3.org/2001/sw/wiki/RDFS and https://www.w3.org/TR/rdf-schema/
[11] https://www.w3.org/OWL/
[12] https://www.w3.org/TR/rdf-sparql-query/

AUTHOR BIOGRAPHIES

Ludovic Font is a Master's thesis student at the École Polytechnique de Montréal since September 2014, under the supervision of Michel Gagnon and Amal Zouaq. He also graduated from the École Nationale Supérieure d'Informatique et de Mathématiques Appliquées de Grenoble in September 2016. His research interests are semantic Web technologies, artificial intelligence and mathematics. He plans on starting a PhD at the École Polytechnique de Montréal in January 2017 in didactic of mathematics.

.

Amal Zouaq is an Associate Professor at the University of Ottawa and an Adjunct Professor at Ecole Polytechnique de Montreal. Previously she was at the Royal Military College of Canada in Kingston, Ontario.
Her research interests include natural language processing, Semantic Web, ontology engineering, knowledge extraction and technology-enhanced learning. She obtained her M.SC. and Ph.D. degrees at the University of Montreal (Canada). She was also a Postdoctoral Research Fellow at the Ecole Polytechnique de Montreal in 2009 and at Athabasca University and Simon Fraser University, funded by the FQRNT (Fonds Québécois de Recherche sur la Nature et les Technologies) in 2010. She serves as a member of the program committee and as a reviewer in many conferences and journals in knowledge and data engineering, natural language processing, eLearning and the Semantic Web.

Michel Gagnon is professor at the Computer Engineering Department of Polytechnique Montreal since 2002. Previously, he worked as a team leader at Machina Sapiens inc., a company which at that time was a leader in the development of grammar checkers, and as a professor at the Univerdade Federal do Parana, in Brazil. He received his Ph.D. degree in computer science in 1993 from the Université de Montreal. Since then, he has been working on natural language processing, with a special attention to semantics. Since 2002, his research activities also include the semantic web, especially its industrial applications. He was co-chair of TALN 2010 Conference, the main scientific event in French for researches in natural language processing. Currently, he is co-leader of WeST lab, whose main activities are related to the extraction of knowledge from texts.