A Study on the Atomic Decomposition of Ontologies

Matthew Horridge[1], Jonathan M. Mortensen[1],
Bijan Parsia[2], Ulrike Sattler[2], and Mark A. Musen[1]

[1] Stanford University, California, USA
[2] The University of Manchester, UK

Abstract. The Atomic Decomposition of an ontology is a succinct representation of the logic-based modules in that ontology. Ultimately, it reveals the modular structure of the ontology. Atomic Decompositions appear to be useful for both user and non-user facing services. For example, they can be used for ontology comprehension and to facilitate reasoner optimisation. In this article we investigate claims about the practicality of computing Atomic Decompositions for naturally occurring ontologies. We do this by performing a replication study using an off-the-shelf Atomic Decomposition algorithm implementation on three large test corpora of OWL ontologies. Our findings indicate that (a) previously published empirical studies in this area are repeatable and verifiable; (b) computing Atomic Decompositions in the vast majority of cases is practical in that it can be performed in less than 30 seconds in 90% of cases, even for ontologies containing hundreds of thousands of axioms; (c) there are occurrences of extremely large ontologies ($< 1\%$ in our test corpora) where the polynomial runtime behaviour of the Atomic Decomposition algorithm begins to bite and computations cannot be completed within 12-hours of CPU time; (d) the distribution of number of atoms in the Atomic Decomposition for an ontology appears to be similar for distinct corpora.

Keywords: OWL, Ontologies, Atomic Decomposition.

1 Introduction

The Atomic Decomposition of an ontology is essentially a *succinct representation of the modular structure* of that ontology. In this article we present an empirical study on the Atomic Decomposition of ontologies. We begin by introducing modularity in the context of ontologies and then move on to discuss the notion of Atomic Decomposition. We then present a replication study that we have performed, which thoroughly examines the performance of existing software and techniques for computing Atomic Decompositions.

Ontology Modularity. In recent years the topic of ontology modularity has gained a lot of attention from researchers in the OWL community. In the most general sense, a module of an ontology $\mathcal{O}$ is a subset of $\mathcal{O}$ that has some desirable (non-trivial) properties and is useful for some particular purpose. For example, given a biomedical ontology

P. Mika et al. (Eds.) ISWC 2014, Part II, LNCS 8797, pp. 65–80, 2014.

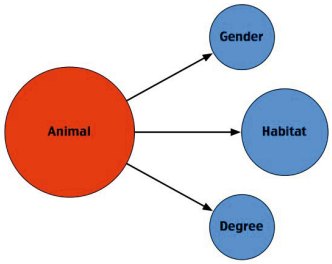


Fig. 1. The ε-connection Partition of the Koala Ontology

about anatomy one might extract a module for the class Heart. This module preserves all information about Heart from the original ontology and can therefore be used in place of the original ontology when a description of Heart is needed. In this case, the module that describes Heart is hopefully much smaller than the size of the original ontology, which makes reusing the description of Heart much easier (in terms of file size, editing and reasoning) than if it were necessary to import and reuse the original ontology in its entirety. For large biomedical ontologies, the difference in size between a module for a term and the size of ontology that the module was extracted from can be very large. For example, Suntisrivaraporn [7] determined that the average size of a module in SNOMED was around 30 axioms compared to the size of the ontology which is over 300,000 axioms. A key desirable property about the kinds of modules discussed here is that given a module $\mathcal{M}$ of an ontology $\mathcal{O}$, the entities in $\mathcal{M}$ are described exactly as they are in $\mathcal{O}$, and from the point of view of these entities, $\mathcal{M}$ is indistinguishable from $\mathcal{O}$.

From Modules to the Modular Structure of an Ontology. Although the above scenario of ontology reuse was the main driving force for the development of proper modularity definitions and practical module extraction techniques, modules have also been used for other purposes such as ontology comprehension. Here, the basic idea is that an ontology can be split up into modules that capture the different *topics* that are described by the complete ontology. Moreover, a dependency relation between topics specifies how they link together, and for a given topic, which other topics it depends upon. For example, in a medical ontology the topic (module) "diseases of the heart" may depend upon the topic "hearts", which may depend upon the topic "organs". Figure 1, taken from [10], shows how this idea could be used in a tool.[1] The circles in the diagram represent the various topics in the Koala ontology[2] with the lines between the circles representing the logical dependencies between these topics. For example, the topic Animal depends upon the topics Gender, Habitat and Degree. Each topic contains axioms that describe

[1] In this particular case, the modules are ε-connection modules, and the diagram has been produced by the ontology editor Swoop.

[2] The ontology can be found in the TONES ontology repository at `http://owl.cs.manchester.ac.uk/repository/`

entities pertaining to that topic. It is easy to imagine that such a representation would be useful for getting an overview of, and browsing, an ontology.

As far as the latest modularisation techniques for OWL ontologies are concerned (efficient syntactic-locality-based techniques) there can be an exponential number of modules for any given ontology with respect to the size of the ontology. However, not all modules are necessarily interesting. This gives rise to the notion of *genuine modules*. A genuine module is essentially a module that is not made up of the union (disjoint union or otherwise) of two or more other modules. Genuine modules are of interest because they can be used to generate a topicality-based structuring of an ontology.

In terms of computing genuine modules, a straight forward algorithm for obtaining the set of genuine modules for an ontology is to compute all of the modules for the ontology and then to compare them with each other in order to eliminate non-genuine modules. However, since there can be an exponential number of modules for any given ontology this is, in general, not feasible. Fortunately, it is possible to efficiently compute the *Atomic Decomposition* of an ontology as a succinct representation[3] of the modules in that ontology. Ultimately, an Atomic Decomposition can be used to generate structures similar to the structure shown in Figure 1. Moreover, it is possible to generate these succinct representations in a runtime that is polynomial (actually quadratic) with respect to the size of the input ontology.

Atomic Decomposition. In short, the Atomic Decomposition of an ontology $\mathcal{O}$ is a pair consisting of a set of *atoms* of $\mathcal{O}$ and a directed dependency relation over these atoms [10]. An atom is a maximal set of axioms (statements from $\mathcal{O}$) which are tightly bound to each other. That is, for a given module $\mathcal{M}$ in $\mathcal{O}$, either *all* of the axioms in an atom belong to $\mathcal{M}$ or else none of them belong to $\mathcal{M}$. More precisely,

Definition 1 (Atom). *let $\mathcal{O}$ be an ontology. A non-empty set of axioms $\mathcal{S} \subseteq \mathcal{O}$ is an* atom *in $\mathcal{O}$ if for* any *module $\mathcal{M} \subseteq \mathcal{O}$, it is the case that (a) either $\mathcal{S} \subseteq \mathcal{M}$, or, $\mathcal{S} \cap \mathcal{M} = \emptyset$; and (b) $\mathcal{S}$ is maximal, i.e. there is no $\mathcal{S}'$ strict superset of $\mathcal{S}$ that satisfies (a).*

For the notions of modules considered in this article, which are depleting modules, the set of atoms for an ontology $\mathcal{O}$ is *uniquely determined*, it partitions $\mathcal{O}$, and is called an *atomic decomposition* of $\mathcal{O}$.

Besides being used for end user facing tasks such as ontology comprehension, the technique of Atomic Decomposition can also be used in non-user facing services as an optimisation technique. For example, Klinov and colleagues use Atomic Decomposition based techniques for the offline computation of modules, in order to reduce memory requirements and speed up reasoning in Web services [11]. Similarly, Tsarkov et al use Atomic Decomposition based techniques for optimising reasoning in their CHAINSAW reasoner [9].

In terms of using Atomic Decomposition in 3rd party tools, there are off-the-shelf implementations of algorithms for computing the Atomic Decomposition of an ontology. These algorithms have been designed and implemented by Del Vescovo and colleagues [10], and further optimised by Tsarkov [8]. Assuming that the modularisation

[3] In this case succinct representation means a non-exponential representation that is linear in the size of the ontology.

sub-routines used by the algorithm have polynomial runtime behaviour (which is the case for the most widely used modularisation algorithms), the worst case complexity of these Atomic Decomposition algorithms is polynomial-time in the size of the input ontology.

Despite the fact that a polynomial-time algorithm is considered to be an efficient procedure, Del Vescovo points out that if a single invocation of the modularisation subroutine takes 1 ms to perform, then it would take ten years to compute the Atomic Decomposition for an ontology the size of SNOMED (300,000 axioms in size). However, Del Vescovo performed a series of experiments on a restricted subset of the BioPortal [6] corpus of ontologies and her results indicate that, in practice, the algorithm performs well and is useable in tools.

Aims and Objectives. Given the potential of Atomic Decomposition techniques for use in both user-facing and non-user-facing tools, in this article we aim to check the claims of the practicality of the optimised algorithm for computing Atomic Decompositions. Del Vescovo's original experiments were performed on a subset of 253 ontologies from the NCBO BioPortal repository. Amongst various filtering criteria, Del Vescovo excluded ontologies from the experiment that were greater than 20,000 axioms in size. Clearly, this leaves some room for verification. We therefore replicate Del Vescovo's experiments, showing that they are repeatable, and we verify the claims made by extending the experiments using a current, and complete, snapshot of the BioPortal corpus. We also bolster our results with a much larger corpus of 4327 ontologies that includes non-biomedical ontologies—specifically, we use the Semantic Web corpus described by Matentzoglu and colleagues at ISWC 2013 in "A Snapshot of the OWL Web" [4].

We make the following contributions:

- We replicate Del Vescovo's Atomic Decomposition experiments. We show that they are repeatable and we verify the runtime performance results on the exact corpus used by Del Vescovo.
- We use the same methodology and software to extend the experiments on the complete BioPortal corpus. This includes ontologies that are an order of magnitude larger than the paired down corpus used by Del Vescovo. We do this to investigate the claims that the techniques are practical.
- We carry out another round of experiments on a third corpus of 4327 ontologies obtained from a Web-crawl. As well as being larger than the BioPortal corpus, this Web-crawl corpus contains non-biomedical ontologies, which may reflect different styles of modelling. We examine both the runtime performance of the Atomic Decomposition algorithm and also the number of atoms per ontology, comparing these result to the results from Del Vescovo's corpus.
- We discuss how the nature of the ontologies affects the results of the second and third experiments and make some recommendations for future work.

2 Preliminaries

In the work presented here, we deal with a corpus of ontologies written in the Web Ontology Language (OWL), and more specifically OWL 2, its latest version [5]. Through-

out the rest of this article we refer to OWL 2 simply as OWL. In this section, we present the main OWL terminology that is useful in the context of this article. We assume that the reader has basic familiarity with ontologies and OWL.

OWL: Entities, Class Expressions, Axioms and Ontologies. An OWL ontology is a set of *axioms*. Each axiom makes a *statement* about the domain of interest. The building blocks of axioms are *entities* and *class expressions*. Entities correspond to the important terms in the domain of interest and include *classes*, *properties*, *individuals*, and *datatypes*. The *signature* of an ontology is the set of entities that appear in that ontology. OWL is a highly expressive language and features a rich set of class constructors that allow entities to be combined into more *complex class expressions*. As a convention, we use the letters A and B to stand for class names and the letters C and D to stand for (possibly complex) class expressions. We also use the word *term* (or *terms*) as a synonym for entity (entities).

Syntactic-Locality-Based Modularity. The most widely implemented form of modularity in available tools, and the type of modularity used by Del Vescovo and thus in the experiments in this article, is *syntactic-locality-based modularity*. Given an ontology $\mathcal{O}$ and a signature Σ which is a subset of the signature of $\mathcal{O}$, a syntactic-locality-based module $\mathcal{M} = \mathsf{Module}(\mathcal{O}, \Sigma) \subseteq \mathcal{O}$ can be extracted from $\mathcal{O}$ for Σ by inspecting the syntax of axioms in $\mathcal{O}$. Syntactic-locality-based-modules have the desirable property that given an entailment α expressed using terms from Σ, $\mathcal{M}$ behaves exactly the same as $\mathcal{O}$. That is, $\mathcal{M}$ entails α if and only if $\mathcal{O}$ entails α. Given $\mathcal{O}$ and Σ, it is possible to extract three three main types, or *notions*, of syntactic-locality-based modules: the $\bot$-module (pronounced "bottom module"), the $\top$-module (pronounced "top module") and the $\top\bot^\star$-module (pronounced "star module"). To take a *very rough*, *over-simplistic view*, a $\bot$-module includes axioms that define relationships between terms in Σ and more general terms in $\mathcal{O}$, a $\top$-module includes axioms that define relationships between terms in Σ and more specific terms in $\mathcal{O}$, and a $\top\bot^\star$-module includes axioms that define and preserve relationships between terms in Σ.

Atomic Decomposition. An atomic decomposition of an ontology $\mathcal{O}$ is a pair $(\mathcal{A}(\mathcal{O}), \succ)$, where $\mathcal{A}(\mathcal{O})$ is the set of *atoms* induced by the genuine modules of $\mathcal{O}$, and $\succ$ is a partial order (dependency relation) between the atoms. An atom is a set of axioms (from $\mathcal{O}$) all of which, for a given Σ and corresponding genuine module $\mathcal{M}$, are either contained within $\mathcal{M}$ or are not contained within $\mathcal{M}$. An atomic decomposition can be computed within a period of time that is polynomial with respect to the size of the ontology. For a given ontology $\mathcal{O}$ and each notion of syntactic-locality it is possible to compute an Atomic Decomposition of $\mathcal{O}$. This gives us a $\bot$ Atomic Decomposition (or $\bot$-AD for short), a $\top$ Atomic Decomposition (or $\top$-AD for short), and a $\top\bot^\star$ Atomic Decomposition (or $\top\bot^\star$-AD for short). The $\bot$-AD highlights dependencies of more specific atoms on more general atoms, the $\top$-AD highlights the dependencies of more general atoms on more specific atoms, and the $\top\bot^\star$-AD highlights differences between atoms.

Class Expression, Axiom and Ontology Length. In line with the reporting of results in Del Vescovo's work [10], we use the notion of the "length" of an ontology to report

the results in this article. In essence the length of an ontology is the number of steps required to parse the symbols in an ontology and reflects the number of operations required to compute a module for some signature. For example, the length of $C \sqsubseteq D$ is the length of C plus the length of D. The length of $C \sqcap D$ is the length of C plus the length of D. The length of the class name A is 1. The length of an ontology $\mathcal{O}$ is the sum of the lengths of the axioms in $\mathcal{O}$. For the sake of brevity we do not give a complete definition of length here. Instead we stick with the intuitive meaning and refer the reader to page 23 of Del Vescovo's thesis [10] for a complete definition.

3 Previous Studies on Atomic Decomposition

The most comprehensive study on Atomic Decomposition to date is presented in Del Vescovo 2013 [10]. In this work, Del Vescovo describes a series of experiments on 253 ontologies that were taken from a November 2012 snapshot of the NCBO BioPortal repository [6]. For each ontology Del Vescovo investigated the time to compute the ontology's $\bot$-AD, $\top$-AD, and $\top\bot^*$-AD and she also explored the makeup of the structure of each Atomic Decomposition.

The expressivity of ontologies contained in Del Vescovo's corpus, ranges from lightweight $\mathcal{EL}$ [1] (OWL2EL) and $\mathcal{AL}$ (56 ontologies), through $\mathcal{SHIF}$ (OWL-Lite, 51 ontologies) to $\mathcal{SHOIN}$ [3] (OWL-DL, 36 ontologies) and $\mathcal{SROIQ}$ [2] (OWL2DL, 47 ontologies). While this corpus does not contain ontologies that could not be downloaded or parsed from BioPortal, for obvious reasons, it also excludes BioPortal ontologies that are either (a) inconsistent or, (b) that are greater than 20,000 axioms in size.

Given that Del Vescovo's experiments are limited to a single filtered corpus, which has itself evolved since 2013, a replication study, which uses both the current BioPortal corpus and other ontology corpora, would be useful in order verify her results and help reduce threats to the external validity of her experiments. In what follows we therefore repeat and extend her experiments using three different corpora of ontologies.

4 Ontology Corpora

In our replication experiments which follow we use three different ontology corpora. The first, is the *exact* corpus used by Del Vescovo. We refer to this as the DEL-VESCOVO corpus. The second and third corpora, which contain much larger ontologies than the DEL-VESCOVO corpus, are made up from all parseable OWL (and OWL compatible syntaxes such as OBO) ontologies from the BioPortal ontology repository [6], and ontologies from a Web-crawl. We refer to these as the BIOPORTAL corpus and the WEB-CRAWL corpus respectively. BioPortal is a community-based repository of biomedical ontologies [6][4], which at the time of writing contains more than 360 biomedical ontologies written in various languages.

We now describe the three corpora in more detail. All three corpora, along with summary descriptions for each (sizes, expressivities etc.), may be found on-line.[5]

[4] http://bioportal.bioontology.org

[5] http://www.stanford.edu/ horridge/publications/2014/iswc/atomic-decomposition/data

The DEL-VESCOVO Corpus (242 Ontologies)

The corpus used by Del Vescovo is described in detail in Del Vescovo 2012 [10]. It contains a handful of ontologies that are well known ontologies in the area of modular-ontologies research, namely Galen, Koala, Mereology, MiniTambis, OWL-S, People, TambisFull, and University. It also contains a subset (234 ontologies) of the ontologies from a November 2012 snapshot of the NCBO BioPortal repository. Del Vescovo graciously provided us with the exact set of ontologies used in her experiment. For each ontology in the corpus, its imports closure was provided to us merged into a single OWL/XML ontology document.

The BIOPORTAL Corpus (249 Ontologies)

Since the DEL-VESCOVO corpus contains a subset of the ontologies from BioPortal, and in particular does not contain ontologies whose sizes are greater than 20,000 axioms, we decided to construct a corpus based on all of the downloadable, and parseable, OWL and OBO ontologies contained in BioPortal. The corpus was constructed as follows: We accessed BioPortal on the 5th of May 2014 using the NCBO Web services API. We downloaded all OWL compatible (OWL plus OBO) ontology documents. For each document in the corpus we parsed it using the OWL API version 3.5.0, merged the imports closure and then saved the merged imports closure into a single ontology document. We silently ignored missing imports and discarded any ontologies that would not parse. The total number of (root) ontology documents that could be parsed along with their imports closures was 249.

The WEB-CRAWL Corpus (4327 Ontologies)

The WEB-CRAWL corpus is based on a corpus obtained by crawling the Web for ontologies and is described by Matentzoglu in the ISWC 2013 article, "A Snapshot of the OWL Web" [4]. This is a large and diverse corpus containing ontologies from many different domains (including biomedicine). A "raw" version of the corpus was supplied to us by Matentzoglu as a zip file containing the exact collection of RDF ontology documents that were obtained by the Web-crawl. For each document in this collection, we parsed it using the OWL API version 3.5.0, merged its imports closure and then saved the merged imports closure into a single ontology document. We silently ignored missing imports and discarded any ontologies that would not parse.[6] The total number of (root) ontology documents that could be parsed along with their imports closures was 4327.

Corpora Summary

Table 1 shows ontology sizes (number of logical axioms) and lengths for the three corpora. Looking at the 90th and 99th percentiles, and also the max values of the BIO-PORTAL corpus, and comparing these to those of the DEL-VESCOVO corpus, it is clear

[6] In the time between the Web-crawl and present day several imported ontologies have become unavailable.

Table 1. A summary of the three ontology corpora. For each corpus the 50th, 75th, 90th, 99th and 100th (Max) percentiles are shown for ontology size (number of logical axioms) and ontology length. For any given percentile Pn, the value represents the largest size (or length) of the smallest n percent of ontologies.

Corpus	P50	P75	P90	P99	Max
DEL-VESCOVO #Ax	691	2,284	4,898	12,821	16,066
Length	1,601	5,812	14,226	35,327	38,706
BIOPORTAL #Ax	1,230	4,384	25,942	324,070	433,896
Length	3,113	12,303	62,950	835,834	1,209,554
WEB-CRAWL #Ax	105	576	3,983	68,593	740,559
Length	255	1,427	11,374	184,646	2,720,146

to see that the BIOPORTAL corpus includes much larger ontologies, both in terms of size (an order of magnitude larger) and length (two orders of magnitude larger). Similarly, the WEB-CRAWL corpus is distinctively different in terms of size. It contains a lot of small and mid-sized ontologies (75% being 576 axioms or less), and a also some extremely large ontologies. For example, the largest ontology in the WEB-CRAWL corpus contains 740,559 axioms (it has a length of 2,720,146), which is two orders of magnitude larger than the largest ontology in the DEL-VESCOVO corpus.

5 Materials and Methods

Apparatus

All experiments were performed using Ubuntu GNU/Linux machines running 24-core 2.1 GHz AMD Opteron (6172) processors. The machines were running Java version 1.7.0_25 OpenJDK Runtime Environment (IcedTea 2.3.10).

Algorithm Implementation

For computing Atomic Decompositions we used the off-the-shelf implementation provided by Del Vescovo and Palmisano. The implementation is available via Maven Central (`maven.org`) with an artifactId of `owlapitools-atomicdecomposition`. We used version 1.1.1 dated 23-Jan-2014. For parsing and loading ontologies we used the OWL API version 3.5.0—also available via Maven Central.

Procedure

The algorithm implementation described above was used to compute the $\bot$-AD, $\top$-AD and $\top\bot^*$-AD of each ontology in each of the three corpora (DEL-VESCOVO, BIOPORTAL , WEB-CRAWL). Each Atomic Decomposition was run as a separate process with 8

Gigabytes of RAM set as the maximum available memory for the Java Virtual Machine (-Xmx8G).[7] A timeout of 12 hours was imposed for each kind of Atomic Decomposition on each ontology. The CPU-time required for each Atomic Decomposition was measured using the JavaThreadMX framework. Finally, for each Atomic Decomposition, the number atoms and the sizes of each atom were recorded.

6 Results

In what follows we present the main results that we obtained in this replication study. An analysis and interpretation of the results takes place in Section 7.

The times for computing each type of Atomic Decomposition are shown in Figures 2–7. To make comparison with Del Vescovo's work easier the results for the DEL-VESCOVO corpus have been repeated throughout the figures. Figures 2, 3 and 4 show CPU-times for the Atomic Decompositions of the DEL-VESCOVO corpus versus the BIOPORTAL corpus for $\bot$-AD, $\top$-AD and $\top\bot^*$-AD respectively. Figures 5, 6 and 7 show CPU-times for the Atomic Decompositions of the DEL-VESCOVO corpus versus the WEB-CRAWL corpus for $\bot$-AD, $\top$-AD and $\top\bot^*$-AD respectively. For each Figure, the x-axis plots the *length* of the ontology and the y-axis plots the time in milliseconds (ms) for the associated computation. It should be noted that the axes in all plots are logarithmic.

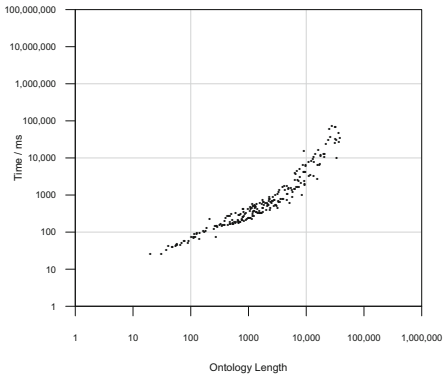
Summaries of CPU-times for each corpus are described below and presented in Tables 2, 3 and 5. Due to the large spread of times, some of the summaries that we present include percentile times (for the 90th, 95th and 99th percentiles). The time for the nth percentile represents the maximum time taken for n percent of ontologies in the relevant corpus. For example, the 95th percentile time for $\bot$-AD in the DEL-VESCOVO corpus (shown in Table 2) is 23,366ms. This means that 95 percent of ontologies in this corpus can be decomposed in 23,366ms or less.

The DEL-VESCOVO corpus All computations finished within the 12 hour time-out window. A summary of the CPU-time required to compute Atomic Decomposition over the corpus is shown in Table 2. All times are shown in milliseconds.

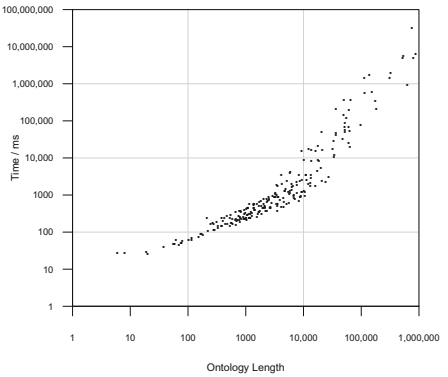
The BIOPORTAL Corpus Within this corpus 240 ontologies completed within the 12 hour timeout period. A summary of the CPU-time required to compute Atomic Decomposition over the corpus is shown in Table 3. All times are shown in milliseconds. There were 9 timeouts, with the ontologies that timed out being very large in size. Table 4 shows these ontologies, along with their sizes and lengths. Although these ontologies timed out, we note that there are other very large ontologies that do not time out. For example, three such ontologies are: one with 433,896 axioms and a length of 1,209,554; one with 356,657 axioms and a length of 891,619; and one with 227,101 axioms and a length of 726,421.

The WEB-CRAWL Corpus Within the WEB-CRAWL corpus Atomic Decompositions for 4,321 ontologies were completed within the timeout period. The Mean, Standard Deviation (StdDev), Median, 90th percentile, 95th percentile, 99th percentile and

[7] We chose 8 Gigabytes of RAM as this was the limit used in Del Vescovo's original work. We acknowledge that that there are other differences in the hardware used, but where possible we used the same parameters, for example, max available RAM.

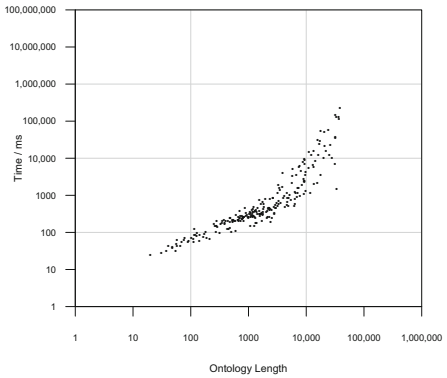


(a) DEL-VESCOVO

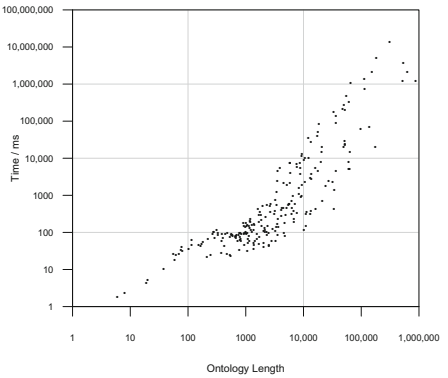


(b) BIOPORTAL

Fig. 2. The time (ms) to compute $\bot$-AD versus ontology length

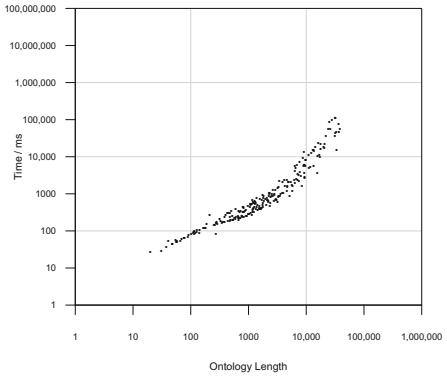


(c) DEL-VESCOVO

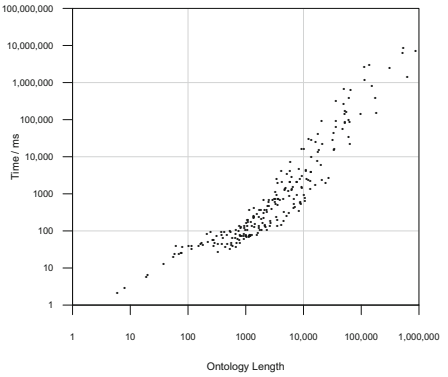


(d) BIOPORTAL

Fig. 3. The time (ms) to compute $\top$-AD versus ontology length

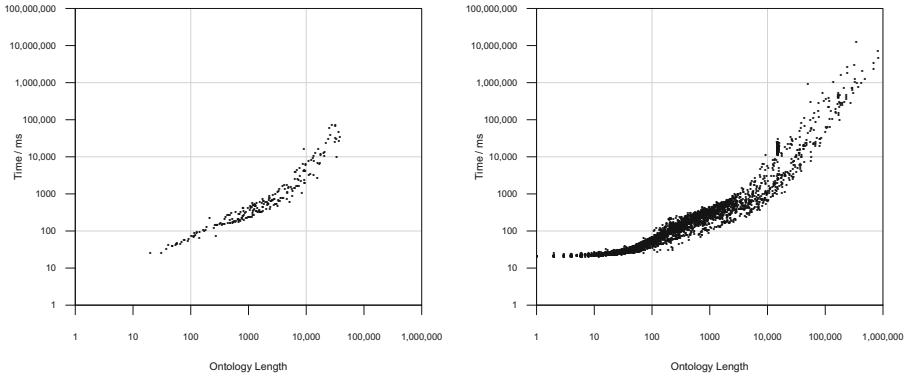


(a) DEL-VESCOVO



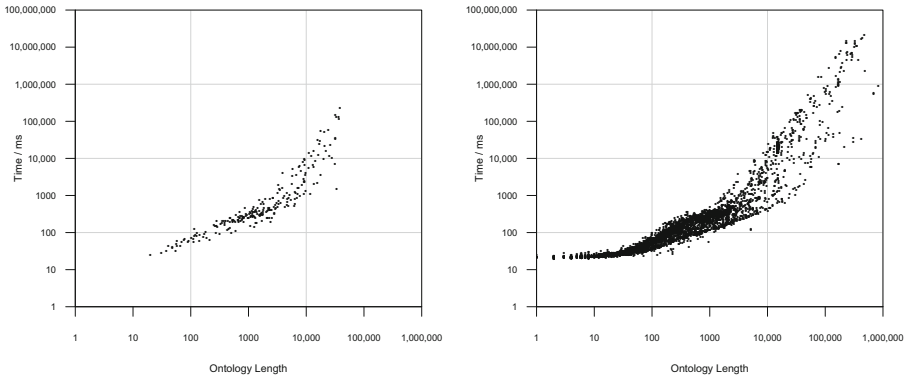
(b) BIOPORTAL

Fig. 4. The time (ms) to compute $\top\bot^*$-AD versus ontology length



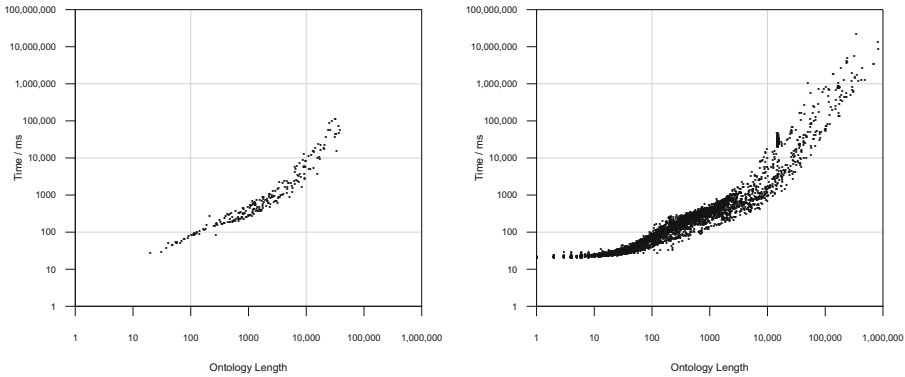
(a) DEL-VESCOVO (b) WEB-CRAWL

Fig. 5. The time (ms) to compute $\bot$-AD versus ontology length.



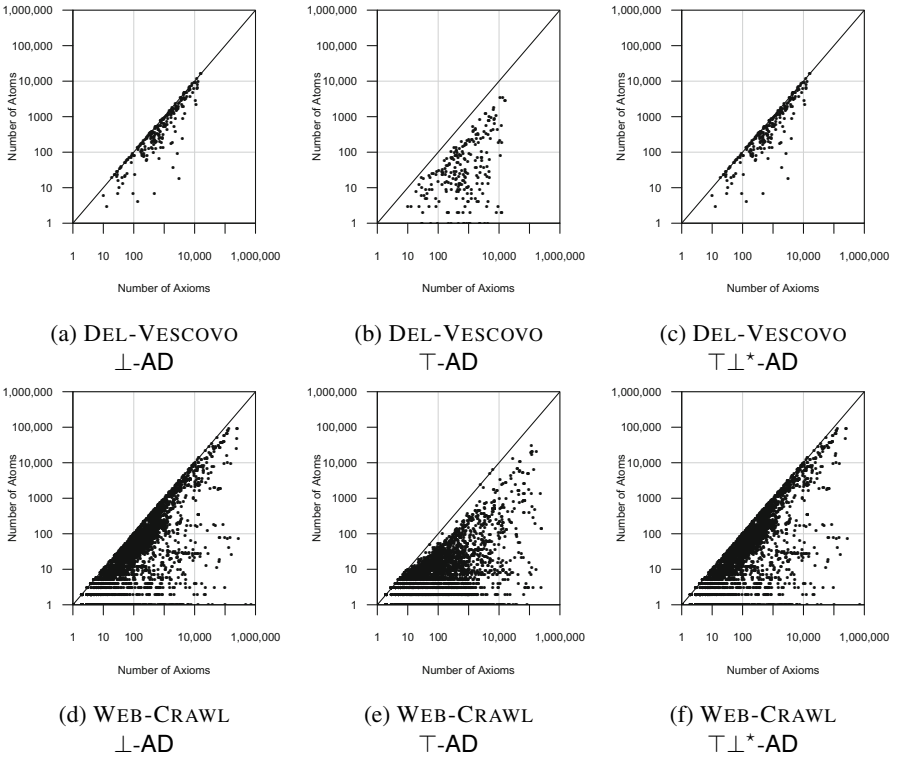
(c) DEL-VESCOVO (d) WEB-CRAWL

Fig. 6. The time (ms) to compute $\top$-AD versus ontology length



(a) DEL-VESCOVO (b) WEB-CRAWL

Fig. 7. The time (ms) to compute $\top\bot^\star$-AD versus ontology length



(a) DEL-VESCOVO ⊥-AD (b) DEL-VESCOVO ⊤-AD (c) DEL-VESCOVO ⊤⊥*-AD

(d) WEB-CRAWL ⊥-AD (e) WEB-CRAWL ⊤-AD (f) WEB-CRAWL ⊤⊥*-AD

Fig. 8. Number of axioms vs number of atoms for the WEB-CRAWL corpus. Each point on the plot represents one ontology. The diagonal line represents a one-to-one correspondence between axioms and atoms, where each dot on this line is an atom containing exactly one axiom.

the Maximum (Max) CPU-time required to compute Atomic Decompositions over the corpus is shown in Table 5. All times are shown in milliseconds. There were 12 ontologies for which timeouts occurred in one form or another. Table 6 shows the failures and where they occurred.

7 Analysis

In what follows we analyse the repeatability of Del Vescovo's work and also make some observations on the verifiability of the results in relation to the fresh ontology corpora that we used.

Are the Experiments Published in Del Vescovo's Work Repeatable? We were able to obtain Del Vescovo's input dataset, the software that she used and we were able to replicate the experiments. Further more, when we replicated the experiments on the DEL-VESCOVO corpus, all of the algorithms terminated on all inputs and, while we do not include an exact comparison of times due to hardware setup differences, our

Table 2. A Summary of the CPU-Time required for computing Atomic Decompositions on the DEL-VESCOVO corpus. All times are shown in milliseconds. Pn represents the maximum time for the nth percentile.

	CPU-Time / (milliseconds)						
Type	**Mean**	**StdDev**	**Median**	**P90**	**P95**	**P99**	**Max**
$\bot$-AD	3,756	10,597	461	8,424	23,366	64,586	72,499
$\top$-AD	5,379	21,857	353	8,559	23,327	131,541	222,760
$\top\bot^*$-AD	5,633	16,275	564	13,051	35,783	93,090	113,581

Table 3. A Summary of the CPU-Time required for computing Atomic Decompositions on the BIOPORTAL corpus. All times are shown in milliseconds.

	CPU-Time / (milliseconds)						
Type	**Mean**	**StdDev**	**Median**	**P90**	**P95**	**P99**	**Max**
$\bot$-AD	31,592	164,585	575	27,988	102,048	587,664	1,778,371
$\top$-AD	56,499	387,190	171	20,573	113,216	1,274,069	5,168,475
$\top\bot^*$-AD	52,687	288,363	306	44,074	155,289	1,053,092	3,046,087

times were in the same order of magnitude as the times computed by Del Vescovo. Figures 2(a) - 7(a) exhibit the same data spread as Figures 4.7, 4.8 and 4.9 in Del Vescovo's presentation of the results [10]. Del Vescovo observed that, over her complete corpus, times for computing $\top$-AD's are generally larger than those for computing $\bot$-AD's. We also observed this aspect (Table 2), mainly for larger ontologies in the corpus. Overall, we therefore consider Del Vescovo's results to be repeatable. Moreover, we consider our results on the DEL-VESCOVO corpus to be a reliable proxy for her results.

What Are the Main Similarities and Differences That Can Be Observed between the DEL-VESCOVO Corpus the Other Two Corpora? The first thing to note are significant differences in the makeup of the DEL-VESCOVO corpus and our corpora. Both the BIOPORTAL corpus and the WEB-CRAWL corpus contain ontologies that are smaller and also ontologies that are (one or two orders of magnitude) larger than the ontologies found in the DEL-VESCOVO corpus (see Table 1). For some of the largest ontologies, certain types of Atomic Decompositions could not be computed within 12 hours (Table 4 and Table 6). Having said this, there are equally large ontologies for which it is possible to compute the Atomic Decompositions. Looking at these Figures 2(a) - 7(a) and comparing these with the corresponding 2(b) - 7(b) the distributions of points on the plots over the same length scales are obviously similar. For smaller ontology lengths and larger ontology lengths, the plots highlight the polynomial trend in computation time. For the largest ontologies, which have lengths in excess of 500,000 and up to 1,000,000, it is noticeable that the computation time strays above one hour (3,600,000ms). Ontologies of this size were not present in Del Vescovo's sample and these results begin to give some idea of what is possible with, and the boundaries of,

Table 4. Ontologies in the BIOPORTAL corpus that had timeouts

Ontology	Axioms	Length
OMIM	112,794	302,298
NPO	160,002	389,385
CVRGRID	172,647	431,713
SNMI	218,231	545,611
NCI	227,101	726,421
RXNORM	253377	759,955
PIERO	288,767	794,163
ICD	356,657	891,619
RADLEX	433,896	1,209,554

Table 5. A Summary of the CPU-Time required for computing Atomic Decompositions on the WEB-CRAWL corpus. All times are shown in milliseconds.

	CPU-Time / (milliseconds)						
Type	**Mean**	**StdDev**	**Median**	**P90**	**P95**	**P99**	**Max**
$\bot$-AD	35,617	732,890	105	968	11,018	21,915	54,340
$\top$-AD	72,124	832,698	100	2,732	21,291	982,399	21,793,805
$\top\bot^*$-AD	37,940	643,327	138	2,246	26,418	688,046	28,993,456

the current implementation. Obviously, whether or not these times are practical depends entirely upon the application in question.

Why Do Several Ontologies in the BIOPORTAL Corpus and the WEB-CRAWL Corpus Have Timeouts? The primary cause is the size of the ontology and the size of modules in these ontologies. On closer inspection we found that nearly all of these ontologies have extremely large ABoxes. Browsing through them in Protégé also revealed that these ABoxes are largely used for annotation purposes as their individual-signatures were puns of class names which participated in labelling property assertions (such as skos:notation, or name, where these properties are data properties rather than annotation properties). Ignoring these ABox assertions, which are essentially annotations, would bring many of the ontology lengths into the bounds whereby the Atomic Decompositons could be computed.

How Does the Number of Atoms Vary between the Different Corpora? Figure 8 shows how the number of atoms per ontology vary over the DEL-VESCOVO corpus and the WEB-CRAWL corpus.[8] The main thing to note is that variation over each corpus is similar for the different notions of Atomic Decomposition. For example, it is easy to see that the number of atoms in a $\top$-AD tend to be fewer and larger when compared to the

[8] For the sake of brevity we only compare these two corpora. The results are similar for the DEL-VESCOVO corpus and the BIOPORTAL corpus.

Table 6. Ontologies in the WEB-CRAWL corpus that had timeouts. Ontologies are sorted by length.

OntologyId	Axioms	Length
3631	117,135	234,268
3069	139,358	288,755
4093	119,560	327,946
3886	168,826	423,119
3147	230,477	474,265
4301	334,546	693,230
2245	277,039	816,406
1577	539,885	1,128,610
1123	238,310	1,495,684
496	714,789	1,892,611
47	740,559	2,122,416
2658	476,620	2,720,146

$\bot$-AD and $\top\bot^*$-AD.[9] The other thing to note is that the majority of $\top$-AD and $\bot$-AD atoms are fine-grained. This phenomena is manifested as the points clustering around the diagonals in the plots for these types of decompositions. In this sense, ontologies in the WEB-CRAWL corpus exhibit similar modular structures to the ontologies in the DEL-VESCOVO corpus.

What Is the Practical Implication of These Results? The algorithm for computing Atomic Decompositions has a theoretical worst-case complexity of polynomial runtime behaviour. The polynomial runtime over all corpora is evident from looking at the plots of CPU-time vs ontology length. For the vast majority of ontologies, Del Vescovo's observation, that computing the Atomic Decompositions for naturally occurring ontologies is practical holds—over all corpora the Atomic Decompositions for 90% of ontologies could be computes in less than 30 seconds. For the handful of extremely large ontologies, the polynomial runtime behaviour of the algorithm begins to bite and there a small number of these ontologies for which it is not possible to compute the Atomic Decomposition within what one might regard as a reasonable time frame. For balance, we note that there are huge ontologies for which it is possible to compute the Atomic Decompositions including ontologies of sizes 433,896 axioms, 356,657 axioms and 227,101 axioms.

8 Conclusions

In this article we performed a replication study using an off-the-shelf Atomic Decomposition algorithm on three large test corpora of OWL ontologies. The main aim of this work was to replicate and verify previously published results. Our findings indicate that

[9] Recall that atoms are disjoint with each other and the complete set of atoms for an ontology covers that ontology.

(a) the previously published empirical studies in this area are repeatable; (b) computing Atomic Decompositions in the vast majority of cases is practical, in that they can be computed in less than 30 seconds in 90% of cases, even for ontologies containing hundreds of thousands of axioms; (c) there are occurrences of extremely large ontologies ($< 1\%$ in our test corpora) where the polynomial runtime behaviour of the Atomic Decomposition algorithm begins to bite, and computations cannot be completed within 12-hours of CPU time; (d) the distribution of number of atoms in the Atomic Decomposition for an ontology appears to be similar for distinct corpora. Finally, the ontology corpora, summary metrics for the corpora, experiment results and software used to run the experiments are available online at `http://www.stanford.edu/~horridge/publications/2014/iswc/atomic-decomposition/data`.

Acknowledgements. This work was funded by Grant GM103316 from the National Institute of General Medical Sciences at the United States National Institute of Health.

References

1. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proceedings of IJCAI (2005)
2. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: Proceedings of KR 2006 (2006)
3. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. J. of Web Semantics 1(1), 7–26 (2003)
4. Matentzoglu, N., Bail, S., Parsia, B.: A snapshot of the OWL Web. In: Alani, H., et al. (eds.) ISWC 2013, Part I. LNCS, vol. 8218, pp. 331–346. Springer, Heidelberg (2013)
5. Motik, B., Patel-Schneider, P.F., Parsia, B.: OWL 2 Web Ontology Language structural specification and functional style syntax. Technical report, W3C – World Wide Web Consortium (October 2009)
6. Noy, N.F., Shah, N.H., Whetzel, P.L., Dai, B., Dorf, M.V., Griffith, N., Jonquet, C., Rubin, D.L., Storey, M.-A., Chute, C.G., Musen, M.A.: BioPortal: Ontologies and integrated data resources at the click of a mouse. Nucleic Acids Research 37 (May 2009)
7. Suntisrivaraporn, B.: Polynomial-Time Reasoning Support for Design and Maintenance of Large-Scale Biomedical Ontologies. PhD thesis, T.U. Dresden (2009)
8. Tsarkov, D.: Improved algorithms for module extraction and atomic decomposition. In: Proceedings of DL 2012 (2012)
9. Tsarkov, D., Palmisano, I.: Chainsaw: a metareasoner for large ontologies. In: Proceedings of ORE 2012 (2012)
10. Del Vescovo, C.: The Modular Structure of an Ontology: Atomic Decomposition and its applications. PhD thesis, The University of Manchester (2013)
11. Del Vescovo, C., Gessler, D.D.G., Klinov, P., Parsia, B., Sattler, U., Schneider, T., Winget, A.: Decomposition and modular structure of BioPortal ontologies. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 130–145. Springer, Heidelberg (2011)