Approaching the axiomatic enrichment of the Gene Ontology from a lexical perspective

Manuel Quesada-Martínez[a], Eleni Mikroyannidi[b], Jesualdo Tomás Fernández-Breis[a], and Robert Stevens[b]

[a] Facultad de Informática, Campus de Espinardo, Universidad de Murcia, 30100, Murcia, Spain. {manuel.quesada, jfernand}@um.es

[b] School of Computer Science, The University of Manchester, Oxford Road, Manchester, UK M13 9PL, United Kingdom. {eleni.mikroyannidi, robert.stevens}@manchester.ac.uk

Abstract

Objective: The main goal of this work is to measure how lexical regularities in biomedical ontology labels can be used for the automatic creation of formal relationships between classes, and to evaluate the results of applying our approach to the Gene Ontology (GO).

Methods: In recent years, we have developed a method for the lexical analysis of regularities in biomedical ontology labels, and we showed that the labels can present a high degree of regularity. In this work, we extend our method with a cross-products extension (CPE) metric, which estimates the potential interest of a specific regularity for axiomatic enrichment in the lexical analysis, using information on exact matches in external ontologies. The GO consortium recently enriched the GO by using so-called cross-product extensions. Cross-products are generated by establishing axioms that relate a given GO class with classes from the GO or other biomedical ontologies. We apply our method to the GO and study how its lexical analysis can identify and reconstruct the cross-products that are defined by the GO consortium.

Results: The label of the classes of the GO are highly regular in lexical terms, and the exact matches with labels of external ontologies affect 80% of the GO classes. The CPE metric reveals that 31.48% of the classes that exhibit regularities have fragments that are classes into two external ontologies that are selected for our experiment, namely, the Cell Ontology and the Chemical Entities of Biological Interest ontology, and 18.90% of them are fully decomposable into smaller parts. Our results show that the CPE metric permits our method to detect GO cross-product extensions with a mean recall of 62% and a mean precision of 28%. The study is completed with an

analysis of false positives to explain this precision value.

Conclusions: We think that our results support the claim that our lexical approach can contribute to the axiomatic enrichment of biomedical ontologies and that it can provide new insights into the engineering of biomedical ontologies.

Keywords: ontology engineering; axiomatic enrichment; biomedical ontologies; gene ontology

1. Introduction

Many biomedical ontologies have been developed in recent years, and their development has been stimulated by their increasing importance in the scientific community [1]. An indicator of such an increasing importance is that ontologies are considered to be a key technology for semantic interoperability in healthcare; see the semantic health project [2] and SemanticHealthNet[1] for examples. According to [3], an *ontology* is a set of *logical axioms* that are designed to account for the intended meaning of a vocabulary; in other words, it is a representation that captures the categories of *objects* in a field of interest and the relationships that those objects have to each other in such a way that it is possible to recognise category membership. For example, the Gene Ontology (GO) [4] has the aim of standardising the representation of gene product attributes across species and thus across databases. The *objects* of an ontology encompass different *components,* such as classes, individuals and object properties / relationships [5]. For human readability, ontology authors include strings of characters as labels that describe an ontology component. However, machines need *logical axioms* that are expressed in a *formal language* with which to reason.

The Open Biomedical Ontologies (OBO) Foundry [6] contributes to the development of an orthogonal collection of biomedical ontologies and defines criteria[2] to be followed by biomedical ontology authors who contribute to the OBO Foundry. Ideally, different contributors would model an *ontology* that focuses on a specific sub-domain, but they would re-use *components* from other ontologies, where appropriate. However, the high level of activity in biomedical ontologies [7] makes reaching this goal a complex task. Moreover, the largest repository of biomedical ontologies is the National Center for Biomedical Ontology's BioPortal [1], which has 372 ontologies at the

[1] http://www.semantichealthnet.eu accessed September 2014

[2] http://obofoundry.org/crit.shtml accessed September 2014

time of this writing.

According to [8], the labels in biomedical ontologies can embed a meaning that is not always represented as *logical axioms* in the ontology. Such hidden semantics constitute not only implicit references to *components* within an ontology but also implicit references to other ontologies. For example, the GO class `*oocyte differentiation*´ is a type of `*cell differentiation*´ that implicitly references the class `*oocyte*´ from the Cell Ontology [9]. The goal of axiomatic enrichment is to make explicit such implicit relationships.

The enrichment of ontologies should establish new formal relationships between existing ontologies, increasing the potential and usefulness of the biomedical applications that are supported by such ontologies [10]. In recent years, different approaches have been proposed within this research area:

- Reference [11] defined the "lexically suggested logical closure" metric for medical terminology maturity. This metric was based on the evaluation of relationships that were proposed by lexical processing programs.

- The Gene Ontology Next Generation project aimed to provide a method for the migration of biological ontologies to *formal languages* such as the Web Ontology Language (OWL) and to explore issues that are related to the maintenance of large biological ontologies [12, 13].

- The Open Bio-Ontology Language (OBOL) project [14] generated formal relationships for existing OBO ontologies using reverse engineering. Later, reference [15] described a frame-based integration of the GO and two other ontologies for improving the *logical axioms* between classes of biological concepts.

- Additionally, [16] proposed a method for the enrichment of ontologies by defining ontology design patterns [17] and their corresponding implementation in the Ontology Pre-Processor Language[3].

- Reference [18] addressed the normalisation of GO by explicitly stating the labels of the compositional classes and partitioning them into mutually exclusive cross-product sets; they

[3] http://oppl2.sourceforge.net/ accessed September 2014

used a combination of OBOL and manual curation to generate *logical axioms*, which they called logical definitions, for selected parts of GO.

- Reference [19] detected hidden semantics, which were named underspecification, in classes from the Systematised Nomenclature of Medicine (SNOMED) that were without *logical axioms*; the authors used natural language processing, which associated each class with a set of equivalence classes that grouped lexical variants (based on their labels), synonyms and translations.

- Reference [10] represented the Foundational Model of Anatomy ontology in OWL2, exploiting the naming conventions in its labels to make explicit some hidden semantics. For example, the pattern *A_of_B* was used to enrich the class *`Lobe_of_Lung´*. In most cases, the name *A of B* is a contraction that is formed from *A* and *B* that omits some *logical axiom p* that relates the two entities, *A* and *B*. The missing *p* was recovered from scanning the list of property restrictions that are attached to the *class*. For example, *`regional_part_of´* is the *p* for *`Lobe_of_Lung´*.

Our approach [16] used a manual analysis of lexical regularities; the results were used for detecting linguistic patterns from a GO sub-hierarchy such as the following: (1) *`X binding´*: the selective, non-covalent, often stoichiometric interaction of a molecule with one or more specific sites on another molecule; or (2) *`translation X factor activity´*: any molecular function that is involved in the initiation, activation, perpetuation, repression or termination of polypeptide synthesis at the ribosome. These linguistic patterns inspired the core concept of this work. In the previous examples, the lexical regularities are the fixed part of the patterns (e.g., *binding*, *translation* or *factor activity)*. Another example of a lexical regularity is *`negative regulation´*, which in general stands for the prevention or reduction of a biological process. This linguistic expression appears in several biomedical ontologies, but it is not usually represented with *logical axioms*. The *`negative regulation of transcription´* and the *`negative regulation of translation´* in the Gene Regulation Ontology or the *`negative regulation´* in the Phenotypic Quality Ontology are similar examples.

Our initial hypothesis was that classes that exhibit lexical regularity encode the meaning of a domain object, and there should be a relation between this class and other classes that exhibit that regularity. In previous work, our method demonstrated its ability to retrieve a large set of classes that exhibited regularities, but not all of them are relevant for enriching the ontology. Hence, we

identified the need for methods that select which sets are relevant for such a purpose. Therefore, in this paper, we extend our method with a new metric that analyses the relation between the lexical regularities exhibited by the labels of the classes and the labels of the classes that are defined in the ontologies and used for enrichment, namely, the cross product extension (CPE). This metric can be understood to be an estimation of the enrichment of those classes that exhibit such regularities. Moreover, we propose three different conditions of the CPE metric that define different types of matches. Our hypothesis here is that such conditions provide information about the degree and type of enrichment that can be expected. For example, in GO, if `*translation´* is a lexical regularity that can be generalised as the pattern `*X translation´*, then the usefulness of the pattern can be estimated by the percentage of classes that exhibit the regularity and that are decomposable as cross-products.

Here, we focus on the GO for several reasons. First, the GO provides a controlled vocabulary for the functional annotation of gene products. To date, GO classes have been used to produce millions of annotations, which are available in resources such as the GO annotation database [20]. Its enrichment would have an impact on the exploitation possibilities of the GO. Such enrichment would enable machines to not only exploit the GO labels but also manage and exploit more fine-grained *objects*, such as biochemical substances or links between molecular functions, biological processes and cellular components. Consequently, enrichment provides additional dimensions for analysis, in this case, functional biomedical data. Second, our analysis of BioPortal ontologies revealed the *prima facie* suitability of the GO for its enrichment: 100% of the classes have labels, 92% of the words of the labels are repeated, and 85% of the ontology labels exhibited 67 lexical regularities [21]. Finally, the GO consortium and other scientists have already identified the necessity of increasing the axiomatic richness of GO, and they have recently developed a partially enriched version, the GO cross-product extensions [18]. In this work, we will compare our results with these GO cross-product extensions. Although each applies different techniques, the comparison will help to evaluate our method and suggest improvements.

2. The lexical analysis framework

In this paper, we consider ontologies that are expressed in OWL. We analyse classes (e.g., owl:Class) and their associated labels, which are specified with the owl:AnnotationProperty of the type rdfs:label; both the labels and classes are in the source file of the ontology. Although one class could have more than one label that is associated, we assume a 1:1 relationship between the labels and classes. This assumption is based on our experiments, which revealed that the ratio of the

number of labels / number of classes is lower than 1.2 for 98.36% of the ontologies (over a corpus of 244 ontologies).

The method applies tokenisation to labels, using a blank (white space) character as a delimiter. A **token (T_i)** is the smallest fragment of text into which a label can be decomposed using a blank. Thus, each label can be expressed as a *token decomposition*: an ordered list of tokens $T_1, T_2, \dots, T_n$. For example, the GO has the class *GO_2000256* with the label *`positive regulation of male germ cell proliferation´*, and its *token decomposition* is $\{T_1 =$*`positive´*, $T_2 =$*`regulation´*, $T_3 =$*`of´*, $T_4 =$*`male´*, $T_5 =$*`germ´*, $T_6 =$*`cell´*, $T_7 =$*`proliferation´*$\}$.

2.1. Definitions for lexical regularities

Our basic assumption is that groups of tokens that appear in many class labels are likely to encode some domain meaning. We refer to such groups of repeated tokens as lexical regularities.

Definition 1. Lexical regularity (LR**)**: a lexical regularity is a group of consecutive, ordered tokens that appear in more than one class of an ontology θ. Because a lexical regularity can be identified by its sequence of tokens $T_i \dots T_{(i+k)}$ (where $k \in \{0,1,2,3\dots\}$), LR $\subseteq$ the set of labels. Every lexical regularity is related to the set of classes that exhibits it, $CS_{\{LR\}}=\{C_1 \dots C_l\}$. $CS_{\{LR\}} \subseteq$ the set of ontology classes. Each LR has a frequency that is equal to the size of the set $CS_{\{LR\}}$, which allows them to be ordered. The more frequent the LR is, the more general it is.

Definition 2. Lexical analysis (LA**):** given an ontology θ, its lexical analysis comprises the whole set of lexical regularities that are found in the ontology, LA$= \{LR_1, \dots, LR_n\}$.

The previous definition of lexical regularity considers the most general case, which requires only one repetition for a lexical regularity. However, two repetitions in hundreds of classes might be irrelevant; to address this issue, we define an input parameter of a lexical analysis: the *coverage threshold*.

Definition 3. Coverage threshold (CV): the coverage of a lexical regularity is the minimum percentage of classes in which a lexical regularity must appear to be included in the lexical analysis.

The value of the *coverage threshold* has a clear impact on the number of lexical regularities that are retrieved, and this number depends on different factors. For example, an ontology that follows a systematic naming convention in its labels would produce a larger number of regularities. The

reason is that the descendant classes include some of the labels that belong to their ancestors.

The *coverage threshold* is a minimal threshold. Once the lexical regularities have been identified and meet the *coverage threshold*, they are grouped and studied by their frequencies. In this way, we group the lexical regularities whose frequency belongs to concrete intervals [(MinFrequency, MaxFrequency)], which would depend on the specific study, as will be illustrated later in the results section.

What does a lexical regularity reveal? Linguistic patterns such as *X binding* are composed of a fixed part (the lexical regularity `*binding´*) and a variable part (*X*). The lexical regularities encode such fixed parts. In addition, the text of the lexical regularity can be the entire label of a class that already exists. For example, '*binding'* is a lexical regularity that appears to be a self-standing label of the class GO_0005488, but it is also a part of the class labels `*frizzled binding´*, `*transcription factor binding´*, and `*FMN binding´*. In this case, `*binding´* is the most general class, and the others are specialisations, which should be formalised with *logical axioms* such as *is_a*.

Given that our method can identify overlaps between tokens in ontology labels, concepts such as external ontologies and exact matches must be defined.

Definition 4. External ontology: any ontology that is not directly imported by the ontology being analysed. An external ontology can be used to enrich such an ontology.

Definition 5. Exact match: the type of overlap between a label of the ontology being analysed and a label of the same ontology or an external ontology. This arrangement occurs when an ordered group of tokens from the label of a class of the ontology that is being analysed is found in the same order as the whole set of tokens of a different class label in the same ontology or in an external ontology.

For example, the search for the lexical regularity '*binding'* in BioPortal retrieved 20 external matches in ontologies such as the Neural-Immune Gene Ontology (reusing the class from the GO) or the National Institute Thesaurus (without reusing the class from the GO).

2.2. General metrics that characterise a lexical regularity

A lexical regularity has some associated descriptors, such as its content (tokens), length (number of tokens), or frequency in an ontology. Next, we summarise the general metrics that our approach uses to analyse the content and structure of the labels of an ontology. These metrics are classified

into the following three groups:

- **Metrics of an ontology $\boldsymbol{\theta}$**: (1) number of classes in an ontology; (2) number of labels; and (3) the type token ratio (TTR), which measures the lexical diversity of the ontology labels. This ratio is calculated as the number of unique tokens (types) / the total number of tokens.

- **Metrics of the lexical regularities**: (1) percentage of classes in which the lexical regularity appears; and (2) number of tokens (e.g., length) of the lexical regularity.

- **Metrics of an ontology $\boldsymbol{\theta}$ based on its lexical regularities**: (1) number of lexical regularities found in the whole ontology for a given *coverage threshold*; (2) set of classes that exhibit lexical regularities: number and percentage of classes that exhibit lexical regularities in a lexical analysis; (3) number of lexical regularities that have external matches; (4) set of classes that exhibit lexical regularities with matches: number and percentage of classes that exhibit a lexical regularity and have at least an exact match in external ontologies; and (5) mean number of external matches by lexical regularities.

2.3. Representation of labels and extraction of lexical regularities

We represent the set of labels of a given ontology using a graph that is similar to the graph in Fig. 1. Each token is a node in the graph, and the arrows represent the order of the tokens in a given label (see Fig. 1). We also store additional information, such as the index in the label (because the same token could appear several times in the same label) and the uniform resource identifier (URI) of the class. The graph is built as the ontology labels are parsed. For example, Fig. 1 shows the part of the graph that highlights the regularity `*regulation of isoprenoid*' of length 3 (tokens).

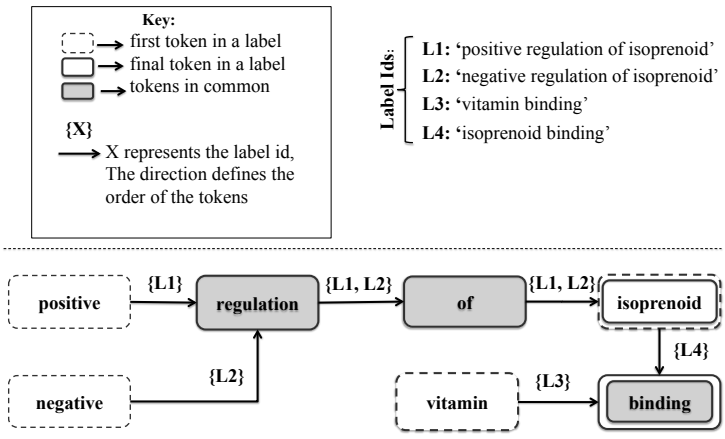

Fig. 1 Graph representation of the content of four labels

The graph shows the analysis of four class labels: (1) *`positive regulation of isoprenoid´*, (2) *`negative regulation of isoprenoid´*, (3) *`vitamin binding´* and (4) *`isoprenoid binding´*. Their lexical analysis yields a graph of 7 nodes (tokens) and highlights 4 shared tokens across the four labels. The token *`regulation´* is common in labels 1 and 2; thus, the corresponding node has two input arrows in the graph. Similarly, token *'of'* is shared across labels 1 and 2; thus, the incoming arrow of the corresponding node in Fig. 1 has the label ids on the top. The direction of the arrow depicts the order of the tokens. For example, the *'regulation of isoprenoid'* regularity consists of three consecutive tokens that are used in labels 1 and 2. Similarly, *'binding'* is shared across labels 3 and 4.

Algorithms 1 and 2 describe how the lexical graph is created (lines 1-19 of Algorithm 1) and how to extract lexical regularities from an ontology (lines 21-25 of Algorithm 1 and Algorithm 2).

```
Function: SearchWholeSetOfLRs
Input:
    (1) ONT: OWL or OBO ontology file
    (2) CV:  Coverage Threshold
Output:
    (1) LRSET:  set with the lexical regularities (LRs) found
```

```
1.  Load ONT in memory using a library for manipulating ontologies
2.  FOR each CLASS in ONT
3.    Extract the LABEL associated with CLASS
4.    FOR each TOKEN of the LABEL
5.      Search in the graph the node TOKEN
6.      IF ( TOKEN not exists in the graph of labels )
7.        Create NODE with id TOKEN
8.        ADD NODE in a global HASHTABLE for query tokens in O(1)
9.      END IF
10.     IF ( TOKEN is not first token in LABEL )
11.       Search in the graph the node TOKEN_PREC that precedes TOKEN
12.       IF( ARROW not exists from TOKEN to TOKEN_PREC )
13.         Create an ARROW from TOKEN to TOKEN_PREC
14.     END IF
15.     Register LABEL in the edge
16.     END IF
17.   END FOR
18.   MNT = update the maximum number of tokens according to LABEL
19. END FOR
20.
21. FOR each NODE in the graph of labels
22.   FOR LR_LENGTH 1 TO MNT-1
23.     LRSET = LRSET Union SearchLRs(NODE, NULL, LR_LENGTH-1, CV)
24.   END FOR
25. END FOR
```

Algorithm 1 Pseudo-code of the algorithm for loading an ontology, extracting the labels, creating the graph and finding lexical regularities. The variables are represented in red.

```
Function: SearchLRs
Input:
    (1) NODE:   node to expand searching lexical regularities
    (2) ACS:    active class (ACS) set of identifiers
    (3) LENGTH: the remainder length of the lexical regularity
    (4) CV:     minimum coverage threshold of the lexical regularities
Output:
    (1) LRSET:  set with the lexical regularities (LRs) found
```

```
1.  IF ( ACS id EMPTY ) RETURN ACS                                  (Lines 1-5: BASE CASES)
2.  IF ( |ACS| < CV ) LRSET = {}, RETURN LRSET
3.  IF ( LENGTH is 0 )
4.    LRSET = ADD LR with ACS as exhibited classes, RETURN LRSET
5.  END IF
6.
7.  FOR each ARROW departing from NODE
8.    NEXT_NODE = node where ARROW arrive
9.    IF ( ACS is NULL )
10.     ADD all the labels id register in ARROW to ACS
11.   END IF
12.
13.   FOR each ARROW_EXP departing from NODE
14.     ADD all the labels ids register in ARROW_EXP to ACS_EXP
15.   END FOR
16.
17.   ACS = ACS intersection ACS_EXP
18.   LRSET = LRSET Union SearchLRs(NEXT_NODE, ACS, LENGTH-1, CV)
19.   ACS is set as the initial value of the parameter
20. ENDFOR
21.
22. RETURN LRSET
```

Algorithm 2 Recursive function in pseudo-code that expands the node-searching regularities. The variables are represented in red.

2.4. Contextualising cross-products in the lexical analysis: the cross-product extension metric

The GO cross-product extensions [18] provided logical definitions for GO classes using genus-differentia constructs of the form "*an X is a G that D*". Here, *X* is the class that we are defining, *G* is the genus (more general class), and *D* is the differentia, a collection of characteristics that serves to discriminate instances of *X* from other instances of *G*. For example, the class *`mitochondrial translation´* can be seen as the genus *`translation´,* and the differentia occurs inside a *`mitochondrion´*. Such logical definitions can be partitioned into mutually exclusive sets that are called cross-products. Each XP is a subset of the cross-product between one genus and one differentia between two ontologies, and such a cross-product is identified by genus_X_differentia (i.e., GeneOntology_X_CellOntology).

If the labels of the classes of the source ontology have exact matches in an enriching ontology, then part of the domain that is defined in the source ontology refers to concepts that are defined in an enriching ontology. For example, the class *`oocyte differentiation´* is formally defined using the

parent class `*cell differentiation´* from the GO biological process ontology and using discriminating characteristics that reference `*oocyte´* in the Cell Ontology. This approach can be represented in OWL Manchester syntax as

'oocyte differentiation' *SubClassOf* 'cell differentiation' ***and***   'results_in_acquisition_of_features_of' ***some*** 'oocyte'

This style of definition is a direct counterpart to that of our approach.

- **Source ontology (θS):** the ontology whose lexical analysis is performed and for which the lexical regularities are obtained. This ontology is the ontology that we want to enrich, and it plays the role of genus.
- **Enriching ontology (θE):** the ontology used for finding exact matches from tokens of those classes where a lexical regularity appears (from the source ontology θS). This ontology is not used to find any lexical regularity but plays the role of filler for the differentia that extend the description in the source ontology.

The selection of θE depends on the domain described in θS. An ontology must play the role of θS if the user has the intuition that its concepts can be defined by reusing concepts from θE. For this reason, given the ontologies *A* and *B*, the meaning of the process *A*=θS and B=θE (AxB) would be very different from the inverse process *A*=θE and B=θS. For example, one of the GO sub-domains is the cellular component ontology, which describes the locations for the gene products at the levels of subcellular structures and macromolecular complexes. Examples of cellular components include `*nuclear inner membrane´*; thus, parts of these classes can be enriched using an ontology that is focused on cells, for example, the Cell Ontology. The inverse approach, which would involve defining types of cells by reusing cellular components, might not make sense.

Moreover, a lexical regularity could have several exact matches in the enriching ontology. For example, `*motor neuron apoptotic process´* defines a specific type of biological process (θS). Its groups of tokens `*motor neuron´* and `*motor´* have exact matches in the Cell Ontology (θE). This arrangement is a consequence of the hierarchy of concepts in ontologies, where specialisations are expressed as compound nouns in natural language. For example, `*motor neuron´* is a specific type of `*neuron´*. Multiple exact matches basically differ in the tokens that are involved and represent different alternatives for enriching the class. This arrangement has led us to propose three different

conditions that impose different criteria on the exact matches.

Definition 6. Cross product extension condition of a class (CPE-class): given the *token decomposition* (TD) of a label, the CPE-class is a Boolean condition, and it has three variations/versions that are controlled by the user. For each version, CPE is true when

- **CPE - Condition 1 (CPE-c1)**: at least one single token of TD has an exact match in the enriching ontology θE.
- **CPE - Condition 2 (CPE-c2):** at least one sub-list of TD has an exact match in the enriching ontology θE. The sub-list is created using combinations of consecutive tokens. The length of the sub-list ranges from 1 to the maximum number of tokens, and thus, it includes condition 1. There could be sub-lists that have the cardinality 2 (or more), which would correspond to all of the tokens of a label, but their corresponding sub-lists of size 1 would not.
- **CPE - Condition 3 (CPE-c3):** the sub-lists of TD that have exact matches in the enriching ontology θE or the source ontology θS include all of the elements in TD. If all of the elements are found in θS, then we call it an *intra-decomposition*.

The three versions of the CPE provide complementary information, not only about our ontology but also about the enrichment ontology that is used. Given an ontology and two different θEs, the CPE values show some properties of the relation of our ontology with the enrichment ontologies. The three conditions of the CPE provide information about how partial and specific the enrichment can be for the different classes. Let us suppose that we perform an analysis on our class with the label *`positive regulation of biological process´* and two classes with the labels *`regulation´* and *`positive regulation´* in two different enrichment ontologies, A and B.

- CPE-c1 would be true for A and not for B, which means that partial enrichment could be achieved using A.
- CPE-c2 would be true when using both A and B, which means that partial enrichment could be achieved with both ontologies.
- CPE-c3 would be false for both A and B, which means that the class cannot be completely enriched with both ontologies. The complete enrichment of a class can be ensured only

when CPE-c3 is true.

The CPE-class condition allows for filtering classes that are based on exact matches; in other words, they are based on an estimation of the enrichment of the lexical regularities that are associated with the classes, as follows:

Definition 7. Degree of CPE of a lexical regularity (CPE-metric): given a lexical regularity and the set of labels in which it appears, we estimate the enrichment of the regularity by measuring the percentage of the members of this set that have the CPE-class condition true. This percentage depends on the CPE-class condition that is chosen as well as the selected θE.

For example, the lexical regularity `*binding´* is found in 1592 labels of the molecular function sub-ontology of the GO, which plays the role of $\theta S$. Using the Chemical Entities of Biological Interest (ChEBI) as $\theta E$, the degree of CPE of `*binding´* using CPE-c3 is 1087 labels; thus, 68.27% of the labels that exhibit regularity are fully decomposable using the class `*binding´* from θS and the other classes from θE. We could not automatically enrich the remaining classes that have CPE-c3 as false, although the results that are provided by less restrictive versions of the CPE and that search for partial decomposition (CPE-c1 or CPE-c2) could help the ontology authors to understand the regularity.

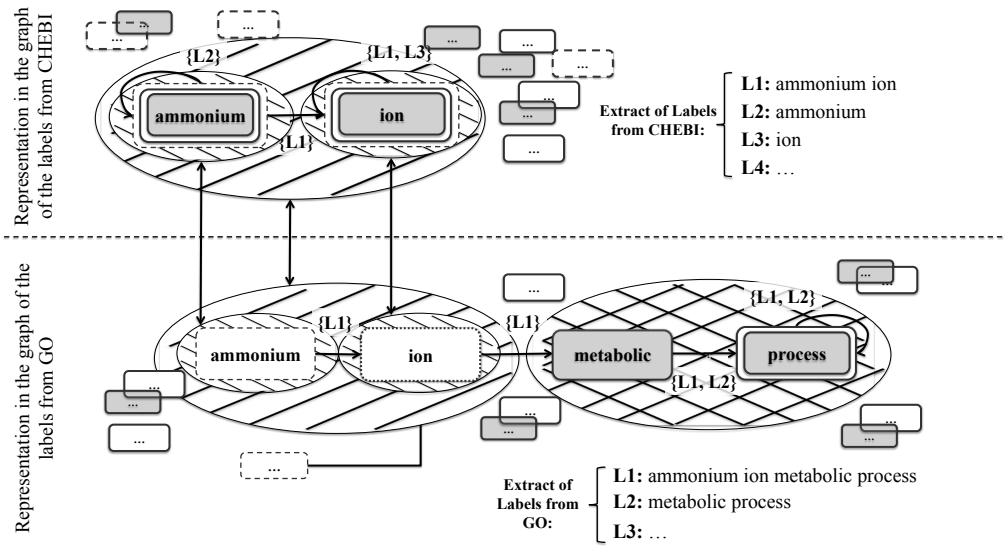

Fig. 2 Graphical representation (which represents the labels as graphs of tokens) of the decomposition of *ammonium ion metabolic process* using classes from GO as θS and ChEBI as θE. The graph would be formed by the whole set of labels from each ontology, but we show only some labels that participate in the decomposition of the class *ammonium ion metabolic process*.

Fig. 2 shows the decomposition of the label `*ammonium ion metabolic process´* (GO_009714), using GO as θS and ChEBI as θE. This metabolic process stands for the chemical reactions and pathways (metabolic processes) that involve ammonium ions (as chemical entities). Fig. 2 shows the graph representation of the 5 labels that illustrate the decomposition in the example. The token decomposition of `*ammonium ion metabolic process´* has 4 tokens: `*ammonium´*, `*ion´*, `*metabolic´* and `*process´*. These tokens are, respectively, in positions 1 to 4. The CPE algorithm finds `*metabolic process´* as the entire class label in θS (see the cross-hatched circle in Fig. 2). The algorithm inspects the edges and detects that these two tokens are the class label that has the identifier L2; thus, the tokens in positions 3 and 4 are marked. Moreover, `*ammonium´*, `*ion´* and `*ammonium ion´* are the entire labels of three classes in θE (see the hatched circles in Fig. 2); then, the tokens in positions 1 and 2 are marked as well. As a result, the value of the metric CPE-c3 for this annotation is true because all of the tokens of the label are found in θS or θE. CPE-c1 and CPE-c2 are also true because single (`*ion´* and `*ammonium´*) and multiple (`*ion ammonium´*) matches are found in the θE. We think that using the three conditions permits us to obtain complementary information that could help the ontology author to make the best decisions to enrich the ontology.

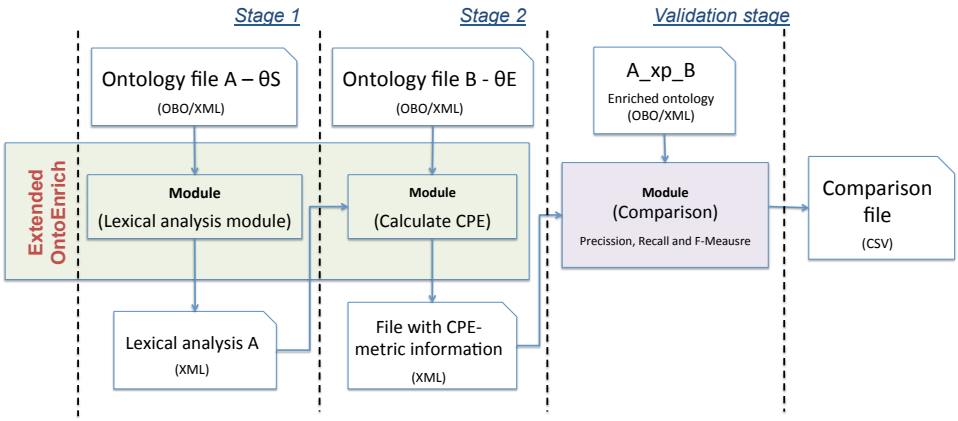

Fig. 3 The lexical analysis workflow

2.5. Implementation of the approach

Fig. 3 shows the workflow of the lexical analysis. The workflow consists of 2 main stages, which are implemented in the new version of our OntoEnrich software [22]. Fig. 3 also includes the validation stage that is used in this paper but that is not a part of the general method.

- (Stage 1): the user specifies an ontology (ontology file A) in OBO or OWL that plays the

role of θS. The lexical analysis is saved into an eXtensible Markup Language (XML) file (lexical analysis A) that is the input to the OntoEnrich module that calculates the metrics. An extract of this file is shown in Fig. 4 and contains two main sections. Section 1 includes information about the ontology and other input parameters of the lexical analysis, such as the value of the *coverage threshold.* Section 2 includes the lexical regularities that are detected in the lexical analysis.

- (Stage 2): the module for calculating metrics takes as inputs the files that have the lexical analysis of θS and θE. The calculation of all of the CPE conditions is performed and saved into a new XML file. This XML file contains only those classes for which the CPE metric is true, and they are grouped by lexical regularities. We call this process the ***reduction of a lexical analysis***, which includes additional information about the decomposition of the CPE (see node <decomposition> in Fig. 4), similar to other classes in θS or θE for which each label is matched.

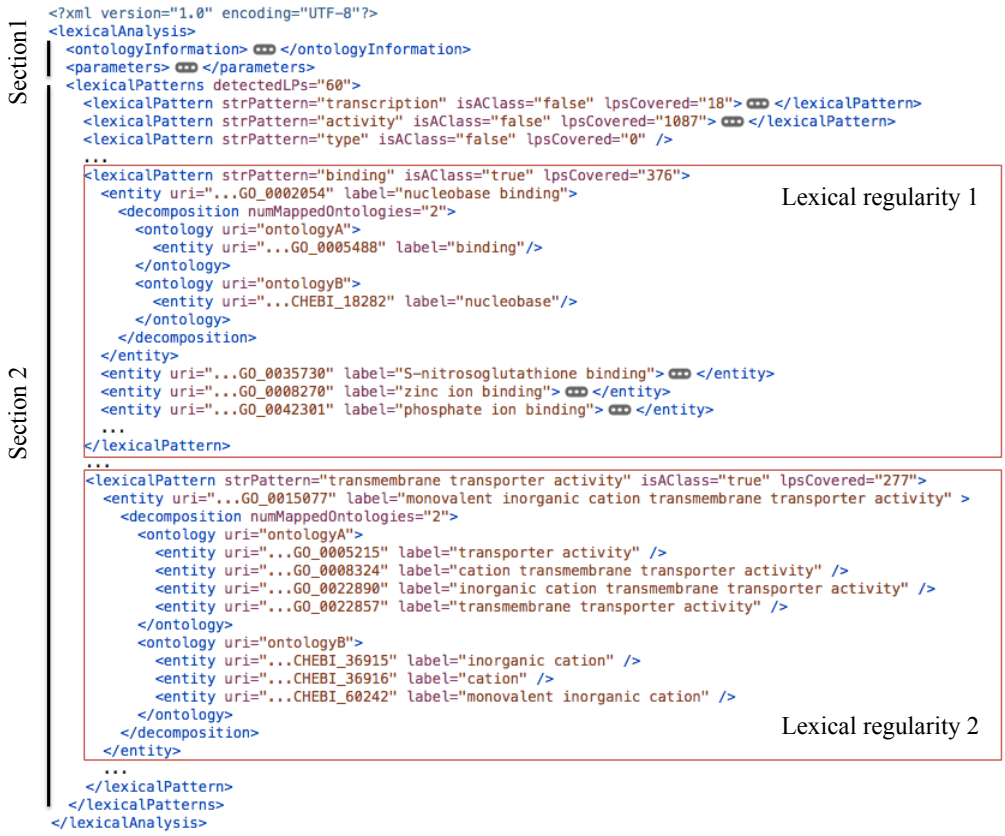

Fig. 4 Fragment of an eXtensible Markup Language (XML) file of a reduced lexical analysis of a part of the Gene Ontology.

- (Validation stage): this stage focuses on the evaluation and comparison of our method against a gold standard. To accomplish this goal, we create two independent files that have URIs of classes in our lexical analysis and the enriched file from Mungall et al. (2011). We explain our strategy in section 3.3.

3. Results

Our experiments were run on a machine that has Mac OS X (10.6.8) and the processor 2.4 GHz Intel Core 2 Duo, with 4 GB 1067 MHz DDR3. The Java virtual machine 1.6 (1.5.0_51-b11-457) was launched with 3 GB of RAM. The current version of OntoEnrich uses the OWL API[4] (v.3.4.3) for manipulating the ontologies. The complete data for our lexical analysis of GO are available at http://miuras.inf.um.es/cpe (accessed September 2014). We discuss the results in section 4.

3.1. Lexical analysis of the Gene Ontology: general metrics

Table 1 shows the general descriptors of the lexical regularities that are extracted from GO as a result of the execution of stage 1 (see Fig. 3). The OWL ontology file was processed in 243.814 seconds (lines 1-19 of Algorithm 1). GO is rich in labels because all of its classes are labelled. The type token ratio is 7.05%, which means that there are many repeated tokens and, therefore, it exhibits regularities. This property is also a sign of the application of a systematic naming of the classes. Finally, the number of tokens in the labels ranges from 1 to 28. After the generation of the lexical graph, we select a value for the *coverage threshold* for finding the regularities. In this experiment, we use 1% (380 classes), for which 67 regularities are retrieved in 2.093 msec (lines 21-25 in Algorithm 1). The mean number of tokens of these lexical regularities is 1.35. Finally, the external matches were searched using all of the ontologies in BioPortal, which were downloaded with the BioPortal web services [1]. A total of 44 out of 67 lexical regularities were found to be full labels in other ontologies, and each match was found, on average, 141 times.

3.2. The lexical analysis and cross-products: using the CPE metric

The analysis of the CPE metric needs the specification of the source ontology θS and the enriching ontology θE. In this case, θS is the GO, which comprises three ontologies: molecular function (MF), biological process (BP) and cellular component (CC). These three aspects are defined as

[4] http://owlcs.github.io/owlapi/ accessed September 2014

three independent ontologies. Thus, when managing them independently, we can focus on a specific subdomain, and using the same *coverage threshold*, we can study the lexical regularities in the different sub-hierarchies.

The following ontologies are used in this experiment: GO, BP, CC and MF are used as θS; and Cell Ontology (CL) and ChEBI are used as θE. Following the terminology from [18], every file is named following the pattern *aXb*, where *a* is the source ontology (θS) and *b* is the enriching ontology (θE). In particular, the following cross-products will be analysed: (1) *bpXcl*, (2) *ccXcl*, (3) *goXchebi* and (4) *mfXchebi*.

CL (ontology for the representation of cell types) or ChEBI are not defined in terms of the GO; as a result, we discard the pairs *clXbp*, *clXcc*, *chebiXgo*, and *chebiXmf*. In addition to applying the criterion of selecting appropriate θEs for each θS, we checked the availability of the cross-product extension files that were developed by the GO consortium and are currently available as GO cross-products[5], which we use to validate our method. The GO consortium has a wide range of experience in the domain of the GO; therefore, if a pair is not available, the absence of a biological intuition behind the pair is likely to be the reason for such unavailability.

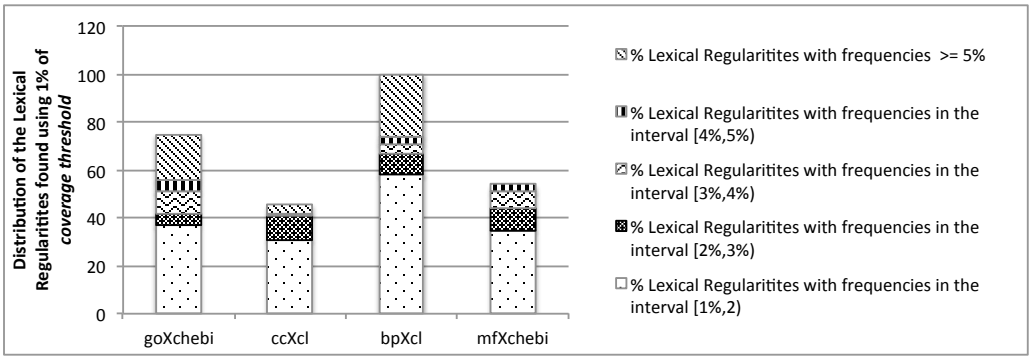

Fig. 5 Representation of the number of lexical regularities obtained in the θS (bp, cc, go, mf) using a *coverage threshold* of 1% and grouping by frequency intervals.

3.2.1. Distribution of lexical regularities by frequency intervals

The number of lexical regularities decreases with an increase in the *coverage threshold* (see Fig. 5). The highest *coverage threshold* retrieves the most general regularities because regularities appear in

[5] http://wiki.geneontology.org/index.php/Category:Cross_Products accessed September 2014

more classes of the ontology. For example, some regularities that were found using a *coverage threshold* of 5% are as follows: *`process´*, *`biosynthetic process´*, *`to´*, *`regulation of´*, *`negative regulation of´*, *`positive regulation of´*, *`involved in´* or *`cell´*. In this set, we can see that the regularity *`process´* is also included in other regularities, such as *`biosynthetic process´*. If we reduce the coverage, new sub-types of processes such as *`metabolic process´* (at 4%) and *`catabolic process´* (at 3%) appear. The frequency analysis of the lexical regularities identified with a coverage threshold of 1% show that 60% are in the frequency interval [1%, 2%]. Fig. 6 shows all of the lexical regularities that were obtained from the GO using a 1% coverage and grouping by the intervals that were previously shown in Fig. 5.

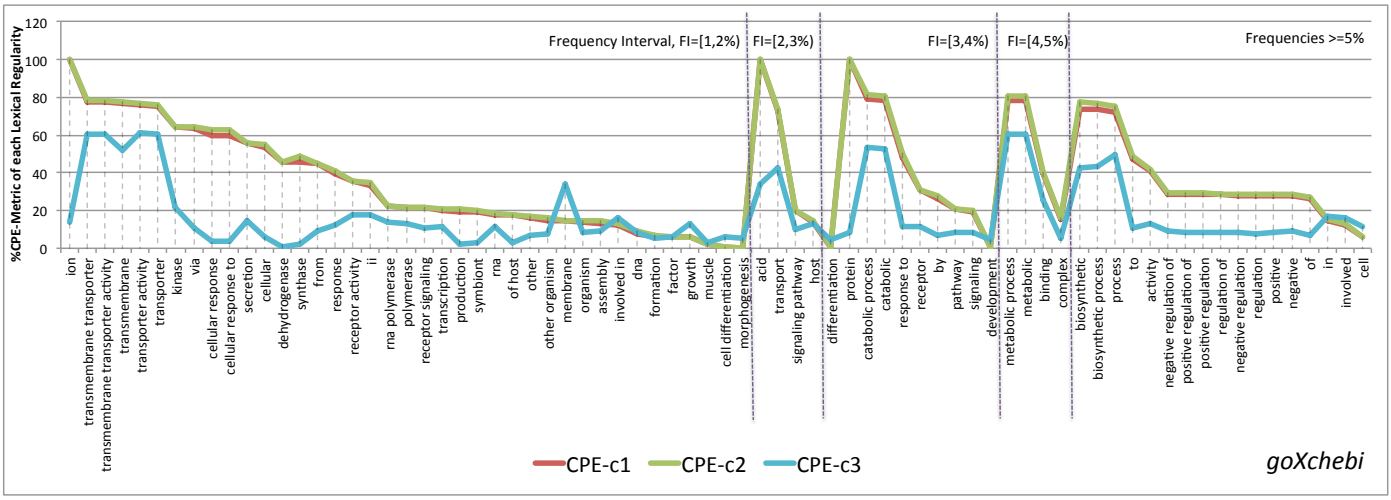

Fig. 6. Graphical representation of the percentage of CPE obtained for the lexical regularities found in GO. The three series represent the results for each version of the CPE. The lexical regularities are ordered first by the frequency intervals and second by the CPE-c1 value.

3.2.2. Lexical regularities and the CPE metric

We have calculated the values of the three versions of the CPE metric and grouped the results by different *frequency intervals*. The results shown in Table 2 are also grouped by combinations of $\theta S \text{x} \theta E$. The rows "CPE-c1", "CPE-c2" and "CPE-c3" show the mean values of the degree of CPE for those lexical regularities that are found in the frequency interval indicated in each column. The "% Regularities" metric stands for the distribution of regularities based on the frequency interval.

Comparing versions of CPE: The mean value of the degree of CPE is 31.48%; this finding means that the fragments of the classes that exhibit lexical regularities are classes in the enriched ontology. The mean value of CPE is similar for the versions CPE-$c1$ and CPE-c2 (except for the two cases

<$bpXcl, CV1$> and <$bpXcl, CV2$>). This finding suggests that the matches that are found to be entire labels of a class in θE are single tokens. Otherwise, the CPE percentage should be greater for CPE-c2, which accounts for sub-lists of more than one token. When CPE-c3 is applied, the value decreases from 37.77% (mean values of CPE-c1 and CPE-c2) to 18.90% (mean value of CPE-c3). This finding means that 18.90% of the classes that are captured by the lexical regularities (CPE-c3) are fully decomposable into smaller parts that are classes of θS or θE. This metric also reveals that GO labels contain tokens that are found in the CL and ChEBI.

Tokens not matched in θE: Here, 18.87% of the classes that were captured by lexical regularities had no matches that covered the whole label. The ideal value is to have 0 tokens not matched. For example, 50% of the tokens of the GO label *`cytosolic creatine kinase complex´* have matches in ChEBI (*`cytosolic´* and *`creatine´*). For the cases with CPE-c1 equal to true and CPE-c3 equal to false, we have calculated the average percentage of the tokens that were not detected with respect to the number of tokens of the label. When the average percentage of non-matched tokens (see Table 3, column 4) is lower than 50%, more than half of the labels correspond to full labels of classes. Thus, they can be used for enrichment although their labels cannot be completely decomposed.

Distribution of the three versions of the CPE by lexical patterns: Fig. 6 shows the values of the three versions of the percentage of CPE for each lexical regularity in *goXchebi*; the corresponding figures using the GO sub-hierarchy are available at the web page. The three series represent the results for each version of the CPE. The lexical regularities are sorted first by the frequency interval and second by the value of CPE-c1. Fig. 6 shows more clearly the influence of the three CPE strategies.

3.3. Comparing the CPE metric with a reference method

We use [18] as a reference method. Despite not being a gold standard, the fact that both methods share the same objective of the axiomatic enrichment of the GO, their expertise in the biological domain and their process, including manual curation, makes it relevant for the evaluation of our results. We calculated the standard metrics of precision, recall and F1-measure using files that have sets of URIs that were generated during stage 3 (see Fig. 3). For a comparison, a template is required to define the equivalences between our method and the reference method. Fig. 7 (1) shows the CPE decomposition for the class *`nucleobase binding´*; Fig. 7 (2) shows the enrichment template of the GO cross-product that was explained in section 2.4 and the equivalent elements in

both methods. In OWL, the classes can be either defined or primitive. A defined class has necessary and sufficient conditions, whereas a primitive class has only necessary conditions. The enrichment template uses primitive classes to create the axioms for the defined class. Fig. 7 (3) shows the GO cross-product axioms for the class `*nucleobase binding*´. This axiom uses the primitive classes `*binding*´ and `*nucleobase*´ and the relation `*results_in_joining_of*"; the lines that connect the different parts of the figure show the equivalences between the classes and the enrichment template.

Our goal is to compare whether the defined classes in GO CPE (the reference method) are captured by decomposable classes in our CPE lexical analysis. To accomplish that goal, we define and extract the following two sets:

- Base Method Set (BMS): URIs of all of the decomposed classes of the lexical reduction. This set does not contain those classes in the decomposition of the CPE that would play the role of genus and differentia.
- Reference Method Set (RMS): URIs of the defined classes in the reference method's enriched ontology. This set does not contain primitive classes that play the role of genus and differentia.

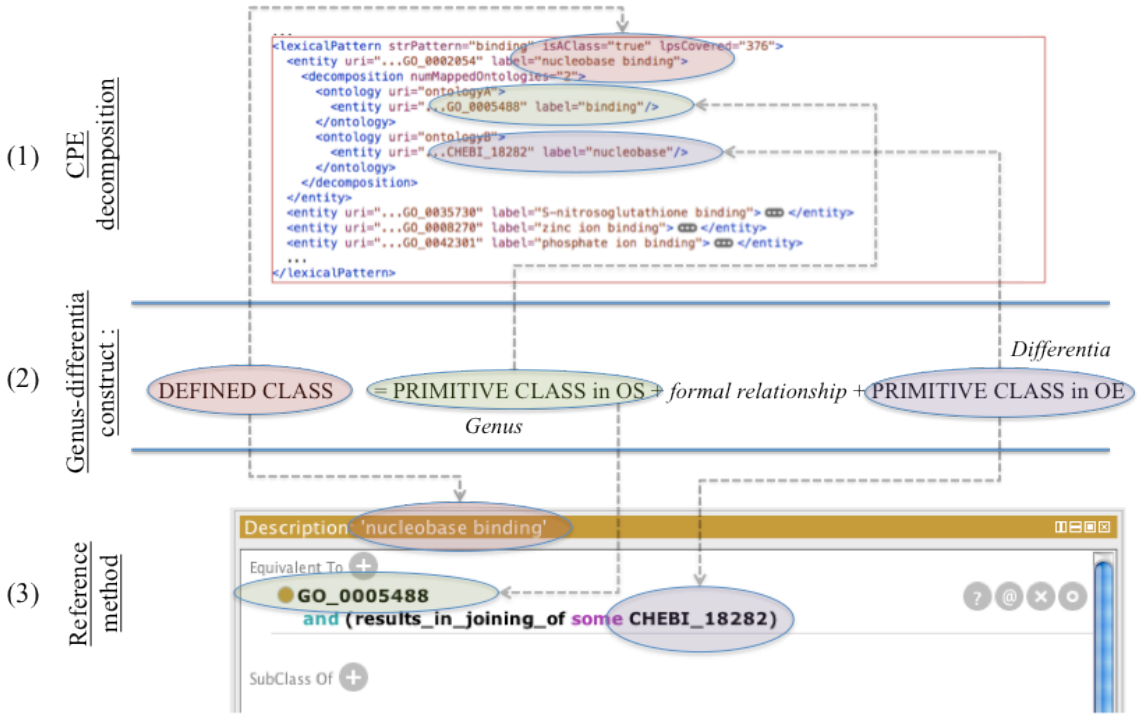

Fig. 7 Enrichment template and equivalent elements between a CPE lexical analysis and the reference method.

We run the experiment of calculating a lexical analysis using a 1% *coverage threshold* for our three θS. Afterward, we reduced the lexical analysis using the CPE information with the corresponding θE. As a result of this reduction, we obtained three independent files using the three versions of the CPE; these files were used for calculating the different BMS sets. The results are shown in Table 4. Columns 2-5 show the number of defined and primitive classes (in OWL terminology) for each method and the CPE condition. The inspection of these four columns helps us to understand the variations in the precision, recall, and false positives. Columns 7-12 and 13-18 show the results of comparing our results with the cross-products that were identified in the reference method. The difference between the two sets of results is that the second set (columns 13-18) has been calculated without considering *intra-decompositions*, which are not considered in the reference method. The last column shows the percentage of reduction in false positives when the *intra-decompositions* are discarded.

According to the results of Table 4, the correspondence between the two methods is not complete, given the mean precision (0.25) and the mean recall (0.62), and the results depend on the pair θSxθE that was taken and the CPE condition that was applied. CPE-c3 is the most restrictive and requires the labels to be fully decomposable; this aspect penalises the recall and benefits the precision.

4. Discussion

4.1. The method

We introduced three variations of the CPE-metric according to the CPE-class condition used in its calculation. The CPE-metric estimates how the classes that exhibit lexical regularities can be decomposed into tokens and how useful such a decomposition could be for the enrichment of the ontology. The *coverage threshold* is the main input parameter of our algorithm; it indicates the minimal percentage of a regularity coverage on the labels. According to Fig. 5, the higher the threshold is, the fewer and more general the lexical regularities that are detected. Lower thresholds lead to more lexical regularities. For example, our results in section 3.2.1 show that more specific types of processes are found when a lower *coverage threshold* is selected. Concerning the CPE values (Fig. 6), we cannot draw any conclusion about the relation between the frequency of the lexical regularities and the values of the CPE.

In [21], we obtained regularities in 216 BioPortal ontologies using *coverage thresholds* from 1% to 5%. *Coverage thresholds* that were lower than 1% cause a considerable growth in the retrieved regularities. For example, the minimum *coverage threshold* (2 labels) produced 64550 regularities in the GO. However, not all of these regularities are relevant. In addition, highly modularised ontologies [23] that are structured in independent sub-hierarchies can focus on certain sub-domains. For example, the GO describes separately in its sub-hierarchies the molecular functions, cellular components and biological processes. If the classes, and consequently their labels, are not equally distributed in the sub-hierarchies, then the use of one *coverage threshold* for the whole ontology will leave out regularities that are specific to the subdomains that are covered in the small sub-hierarchies. Our method could be improved by developing auto-tuning methods for optimising the *coverage threshold*. For example, the algorithm should automatically detect modules and adapt the *coverage threshold* to them.

4.1.1. Why do we use a graph for representing the labels?

We decided to use a graph because it provides benefits for extracting lexical regularities and querying the textual content of the labels. For example, Algorithm 2 searches for exact matches without covering the whole set of classes. This query is frequent when calculating the CPE conditions; thus, we are reducing the time that is needed for calculating the CPE conditions.

We studied N-grams [6] and De Bruijn graphs[7] as alternative representations. N-grams are contiguous sequences of n "items" (phonemes, syllables, letters, words or base pairs) from a given sequence of text or speech; N-grams are commonly used in the field of computational linguistics for detecting regularities. For example, in DNA sequencing, the sequence "AGAGC" has the 2-grams "AG", "GA", "AG", and "GC" and the 3-grams "AGA", "GAG" and "AGC". The frequencies depict regularities such as "AG", which is repeated twice. N-grams would permit the full set of lexical regularities to be obtained but would not allow advantage to be taken of the token redundancy or the creation of links between them. In the running example, if we process the sequence "AGC" after processing "AGAGC", then the frequency table would be: <AG, 3>, <GC, 2>, <AGC, 2>, <GA, 1>, <AGA, 1> and <GAG, 1>. In this case, three frequencies increase, but both "AGC" and "AGAGC"

[6] http://nlpwp.org/book/chap-ngrams.xhtml accessed September 2014

[7] http://www.homolog.us/Tutorials/index.php?p=1.1&s=1 accessed September 2014

are missing. De Bruijn graphs are n-dimensional directed graphs that represent overlaps between sequences of symbols, and they are widely used in bioinformatics to reconstruct genomes through next generation sequencing libraries. They have m^n vertices, which consist of all possible length-n sequences of the given symbols. In our approach, each label could be represented as a De Bruijn graph, but detecting regularities from many De Bruijn graphs would be complex.

Next, we discuss the complexity of our algorithms. The number of classes in the ontology ("c") and the number of unique tokens ("t") influence the size of the graph. Theoretically, in terms of the edges, the worst scenario is a label that contains all of the tokens "t" and in which all of the pairs appear consecutively (e.g., "A A B B C C A C B A"). Then, the number of nodes would be O(t), and the number of edges would be $O(t^t)$. Moreover, each edge contains a set that has the classes that exhibit its tokens consecutively. Then, the worst case would be that all of the pairs are repeated in all of the classes, with the load of each edge being O(c). Hence, in terms of big O, the graph is formed by $O(t^t)$ nodes, and the load of the edges is O(c). Fortunately, in practice, the worst scenario is unlikely, and as a result, we can exploit the sparseness of the data.

We assume the next upper limits of the graph, as follows: "tl" (the maximum number of tokens in one label) and "l" (the maximum number of labels with two consecutive repeated tokens). The use of data structures such as sets or hash maps means that operations such as read, search and add are performed in O(1). Then, the time complexity for building the graph (Algorithm 1, lines 1-19) is O(c*tl), which means that the algorithm is scalable. The graph is created once. Algorithm 2 depends on the length of the regularity ("n") and the coverage threshold. Its complexity is $O(n*a^2*l)+O(n*a*l^2)+O(n*a)$, and it does not depend on the number of classes ("c") but instead on "n", "l" and "a" (the maximum number of arrows that depart from a node). Assuming the sparseness of the tokens, "l" and "a" are smaller than "c". Finally, we prune the search by starting the exploration from the labels in which the first token appears (lines 9-11 of Algorithm 2). The coverage threshold and other conditions are also used for pruning (lines 1-5 of Algorithm 2). For example, the most computationally expensive scenario is when searching all of the regularities that appear in at least 2 classes; using the GO (38533 classes) as θS, our method obtained 64550 results in 774,653 msec. Using new heuristics or caches would speed up the process.

4.2. The analysis of the Gene Ontology cross-products

According to Table 1, 80% of the GO classes that exhibit regularities had exact matches in external ontologies, which is a promising result. However, we cannot affirm that all such classes can be

effectively enriched or that all of those matches are useful for the enrichment. The CPE conditions could contribute to removing potentially noisy matches.

We have used the GO cross-product extensions [18] as a gold standard for evaluating our results. Although both approaches are not directly comparable, they constitute a good reference. In terms of the precision and recall, there is not a baseline performance. However, we discuss next the values that are lower than 40% and the influence of the different CPE conditions. The absence of a true gold standard requires a qualitative analysis of both the false negatives and false positives. The false negatives help us to see to what extent our automatic lexical method was able to identify the GO cross-product extensions and identify potential improvements to our approach. False positives must be validated to reveal where they come from and if they cover new useful decompositions.

4.2.1. Influence of the different conditions of CPE in the results

The highest recall for bpXcl with CPE-c1 and CPE-c2 is 0.85, while for CPE-c3, it is 0.49. The analysis of the defined and primitive classes in Table 4 (columns 4-5) explains this variation. The ratio of defined/primitive between different conditions of the CPE is greater for CPE-c1 and CPE-c2. We interpret these values as CPE-c1 and CPE-c2 and identify decompositions that are based on general classes, which are exhibited in a large number of classes. For example, the token *cell* contributes to CPE-c1 and to CPE-c2 but not to CPE-c3 in many classes. This fact distorts both the precision and recall measures due to the large number of decompositions. For this reason, CPE-c1 and CPE-c2 can be used as a complement to CPE-c3, although CPE-c3 is the most appropriate of all for the automatic processes. On the one hand, the CPE-c3 false positives can be used for creating new axioms that are not detected by the reference method. On the other hand, we cannot conclude that the CPE-c1 and CPE-c2 false positives could be used to create new axioms; however, the inspection of the set of primitive classes can help the users to identify classes that are too general, which could introduce noise into the results. In the next sections, we will focus on CPE-c3.

4.2.2. Precision and false positives in CPE-c3

CPE-c3 could be true only for classes that are decomposed by matches in θS (*intra-decompositions*). The reference method excluded *intra-decompositions* from its files $\theta S \times \theta E$ because they are included in $\theta S x \theta S$. In our work, a false positive means that a GO class was not enriched in the GO cross-product extensions. Therefore, *intra-decompositions* are false positives and decrease the precision. However, this arrangement should not be interpreted as not being useful

for the enrichment of the ontology. For this reason, we have studied the percentage of false positives that are *intra-decompositions* (see column 19 in Table 4) and how their exclusion affects the precision. *Intra-decompositions* are more frequent in BP, CC and GO than in MF, although *intra-decompositions* in GO are a subset of those found in BP and CC.

In cases such as *bpXcl* and *ccXcl*, the precision is doubled by the removal of *intra-decompositions*, but for *ccXcl,* it is still very low, specifically, 0.08. Although the low number of defined classes influences the low recall, this relationship does not justify the low precision. Our study of each false positive revealed that 8 out of the 87 false positives could be directly used to create axioms between the θS and θE classes. This low ratio is mainly due to the token cell, which is found as a class in both θE and θS. This token is a unique token that is found in θE; thus, these false positives can be considered to be *intra-decompositions*. This circumstance would increase the precision to 0.5, which is in line with that obtained for the remaining pairs. We have checked that this situation does not occur in the false positives for the remaining pairs.

4.2.3. Recall

Recall is invariant with respect to *intra-decompositions*. The mean CPE-c3 recall (0.48) is due to three factors: (1) our approach is purely lexical, (2) the lexical comparison is not sensitive to lexical or linguistic variations, and (3) the process is completely automatic. Some classes that were enriched in [18] used lexical coincidences as well as a combination of OBOL and manual curation. The use of OBOL considers the taxonomic relations between ontology classes and other lexical variations, which lexical regularities do not capture. Manual methods include defined classes that are not based on a lexical equivalence and instead are based on further knowledge of the domain. An example is *`oocyte differentiation´*. The reference method enriches this class by combining *`cell differentiation´* and *`oocyte´*. Our lexical method does not assume that *`cell differentiation´* is equivalent, in this context, to *`differentiation´*; thus, in this case, our method does not exploit the taxonomic information *`oocyte differentiation is-a cell differentiation´*, which is available in the ontology. Other defined classes in the reference method were not detected by our method because they did not exhibit any lexical regularity using the coverage threshold of 1%.

4.2.4. Influence of the pairs on the number of defined classes

Columns 2 and 4 of Table 4 show that the nature of θS and θE is determinant in the level of enrichment. The number of defined classes in the enriched file of *bpXcl* and *ccXcl* is 654 and 25,

respectively, and *goXchebi* and *mfXchebi* is 2627 and 3132. However, we cannot conclude that the number of defined classes in the enrichment file influences the precision or recall.

Next, we discuss the impact of the variation of θS for the same θE. We focus our attention on the differences between *goXchebi* and *mfXchebi*. Despite *mfXchebi* being a sub-set of *goXchebi*, the number of defined classes by the reference method for *mfXchebi* is larger than for *goXchebi*. Moreover, the classes that were enriched in *mfXchebi* (the reference method) were not included in *goXchebi* (the reference method) but were included in our lexical analysis, which contributes to decreased precision, for example, from 0.76 to 0.42 in CPE-c3 (column 13, Table 4).

In summary, each cross-product contains its own definitions that depend on the pair θSxθE and on how the classes re-use the content. This arrangement hinders us in our efforts to compare the cross-products from the GO and its three sub-hierarchies.

4.2.5. Number of primitive classes

Our complete decomposition of the labels of the classes causes the number of primitive classes to be larger in comparison with the number used by the reference method (columns 3 and 5 in Table 4). For example, the class '*monovalent inorganic cation transmembrane transporter activity'* (see GO_0015077 in Fig. 4, lexical regularity 2) is not found in the enriched ontology, but all of the tokens in the label can be declared as new classes whose intersection can completely define the GO_0015077 class. The GO_0015077 decomposition has seven different classes that are associated: three from θE (*`inorganic cation´, `cation´,* and *`monovalent inorganic cation´*) and four from $\theta S$ (*`transporter activity´, `transmembrane transporter activity´, `cation transmembrane transporter activity´,* and *`inorganic cation transmembrane transporter activity´*). This overlap occurs in both the true positives and false positives.

4.2.6. Limitations and further work

Our method is strictly lexical when the matches are searched. The variation of one letter between two tokens makes such tokens different. More permissive conditions, which are appropriate for natural language content, could increase the performance of the method. This possibility also opens up a discussion on how to address synonyms (which are common in natural language) or other relationships that are embedded in the labels and structure of the ontology. Finally, given that we pursue an automatic method, the number of primitive classes should be reduced to those that maximise the semantic expressivity of the class. In contrast, CPE-c1 and CPE-c2 provide

information about more general decompositions than those provided by the reference method. This opportunity is useful for the domain experts but not for the automation process.

The experiments that were performed in this work have identified such limitations and have motivated further work, which we describe next. The analysis of the recall for CPE-c3 reveals that, except for the third factor (manual curation), we can complete the knowledge that is captured by lexical regularities as follows:

- **Lexical and linguistic flexibility:** performing lexical analysis at the lexeme level or using approximate string matching would have permitted us to address classes such as *`mitochondrial translation´* because the enriching ontology includes the class *`mitochondrion´* but not *`mitochondrial´*. Other natural language processing (NLP) techniques (derivation, stemming, lemmatisation) that are supported by tools such as the Unified Medical Language System specialist lexicon could improve the method, also. Although this step is not performed by the reference method and also not performed by our method, the linguistic analysis of the defined classes and decompositions is worthwhile (e.g., the types of the words, such as noun, verb).

- **Semantic knowledge**: the analysis of the recall reveals that the inclusion of semantic knowledge offers significant benefits. The inclusion of semantic information into the graph is not trivial. For example, a new type of edge that includes the semantic relations instead of the labels could be created. However, this solution would not work for classes that have more than one token.

Concerning the precision, the analysis of the false positives for CPE-c1 and CPE-c2 has been useful for determining that patterns that are too general cannot be meaningful, and the method must provide users with options for analysing and removing them.

Finally, while our method for decomposition proposes seven classes as the decomposition of GO_0015077, the reference method includes only some of these in the enrichment. This result could occur because they used concrete ontology design patterns [17], which would filter out some of the seven classes. First, our method could be extended with techniques that reduce the overlaps in the decompositions. For example, in the case of CPE-c3, applying heuristics such as selecting the minimum set of classes that cover the whole label can control the decomposition. Then, the decomposition of GO_0015077 (see Fig. 4, lexical regularity 2) could be reduced to '*monovalent*

inorganic cation' and '*transmembrane transporter activity'*. This arrangement would help to automate the enrichment and would also help to increase the precision of the decompositions. Second, we think that our method can be helpful for identifying the ontology design patterns to apply in the enrichment of a given ontology.

4.3. From lexical analysis to the creation of enrichment axioms

Our current approach does not create axioms that enrich ontologies. The use of the CPE decompositions to enrich an ontology with axioms and how to choose the relations is a challenging task, which is left as future work. Reference [10] selected the relations (the class properties) from scanning the list of property restrictions that are attached to the classes, and [18] defined a function that mapped an OBO type-level relation to a corresponding instance-level relation. In addition, NLP techniques will be used to elucidate properties from the text. Lexical regularities such as *`regulation´*, *`positive regulation´, `negative regulation´* or *`involved in´ (*see Fig. 6*)* will be used as properties to enrich classes, where they are exhibited. The definition of the mapping function that is proposed in [18] is a potential option in combination with the aforementioned NLP techniques or with information that is extracted from the regularities. Known issues such as the choice of the correct quantifier, the expression of dispositional meanings, or the problem of negation will be addressed by the mapping function in future work. Moreover, such a function will suggest an applicable ontology design pattern for the regularities. For example, the logical representation of the regularities *`negative regulation´* and *`positive regulation´* requires specifying the type of regulation, either positive or negative, but not both. A structural "value partition" pattern from the Manchester ontology design patterns catalogue[8] can be suggested to create the value partition for regulation.

5. Conclusions

Biomedical ontologies are typically rich in text content (labels and annotations), but such information is often missing in the logical axioms. Our previous work revealed that many of the ontologies from BioPortal present lexical regularities in the structure and content of the labels [21].

Our method has retrieved 44 lexical regularities from the GO that were found in other BioPortal

[8] http://odps.sourceforge.net/odp/html/Value_Partition.html accessed September 2014

ontologies, which cover 80% of the classes. The application of the CPE conditions allows us to reduce the classes that exhibit regularity, which reduces the number of results to those that meet the criteria that are defined under such conditions in terms of only one enriching ontology. We have compared our method for detecting useful regularities with the GO cross-product extension effort. Our method, after removing *intra-decompositions*, showed a mean recall of 62% and a mean precision of 28%. This study has been completed with an analysis of both false negatives and false positives. Concretely, CPE-c3 false positives could be useful for generating new axioms that are based on the decompositions.

Our automatic, lexical approach (CPE-c3) covers 48% of the decompositions that are found in the reference method. This finding confirms the hypothesis that classes that exhibit regularities together with information on token matches (decompositions) are prone to be enriched.

To conclude, we think that our results support the claim that our approach can contribute to the axiomatic enrichment of biomedical ontologies and can provide new insights into the engineering of biomedical ontologies.

Acknowledgements

This project has been made possible because of the funding of the Spanish Ministry of Science and Innovation through grant TIN2010-21388-C02-02, and it was co-funded by the FEDER Programme and the Fundación Séneca through grant 15295/PI/10. The Spanish Ministry of Science and Innovation funds Manuel Quesada-Martínez through fellowship BES-2011-046192 and EEBB-I-13-06298.

Bibliography

[1] P.L. Whetzel, N.F. Noy, N.H. Shah, P.R. Alexander, C. Nyulas, T. Tudorache, et al., BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications, Nucl. Acids Res. 39 (2011) W541–W545. doi:10.1093/nar/gkr469.

[2] V. Stroetman, D. Kalra, P. Lewalle, A. Rector, J. Rodrigues, K. Stroetman, et al., Semantic interoperability for better health and safer healthcare [34 pages], (2009). http://www.semantichealth.org/DELIVERABLES/SemanticHEALTH_D1_1_finalC.pdf (Accessed: September 1, 2014).

[3] N. Guarino, Formal Ontology in Information Systems, in: N. Guarino (Ed.), Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy,

1st ed., IOS Press, Amsterdam, The Netherlands, 1998: pp. 3–15.

[4] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, et al., Gene ontology: tool for the unification of biology, Nat. Genet. 25 (2000) 25–29. doi:10.1038/75556.

[5]P. Lord, Components of an Ontology, Ontogenesis. (2010). http://ontogenesis.knowledgeblog.org/514 (Accessed: 1 September 2014).

[6] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, et al., The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration, Nat Biotech. 25 (2007) 1251–1255. doi:10.1038/nbt1346.

[7] J. Malone, R. Stevens, Measuring the level of activity in community built bio-ontologies, Journal of Biomedical Informatics. 46 (2013) 5–14. doi:10.1016/j.jbi.2012.04.002.

[8] A. Third, Hidden semantics: what can we learn from the names in an ontology?, in: INLG 2012 Proceedings of the Seventh International Natural Language Generation Conference, Association for Computational Linguistics, Utica, IL, USA, 2012: pp. 67–75.

[9] J. Bard, S.Y. Rhee, M. Ashburner, An ontology for cell types, Genome Biol. 6 (2005) R21. doi:10.1186/gb-2005-6-2-r21.

[10] C. Golbreich, J. Grosjean, S.J. Darmoni, The Foundational Model of Anatomy in OWL 2 and its use, Artificial Intelligence In Medicine. 57 (2013) 119–132. doi:10.1016/j.artmed.2012.11.002.

[11] K.E. Campbell, M.S. Tuttle, Kent A. Spackman, A “lexically-suggested logical closure” metric for medical terminology maturity, in: Proceedings of the 1998 AMIA Annual Symposium, AMIA, Lake Buena Vista, FL, USA, 1998: pp. 785–789.

[12] C.J. Wroe, R. Stevens, C.A. Goble, M. Ashburner, A methodology to migrate the gene ontology to a description logic environment using DAML+OIL, in: R.B. Altman, A.K. Dunker, L. Hunter, T.A. Jung, T.E. Klein (Eds.), Pacific Symposium on Biocomputing 2003, World Scientific, Kauai, Hawaii, USA, 2003: pp. 624–635.

[13] M. Egaña Aranguren, C. Wroe, C. Goble, R. Stevens, In situ migration of handcrafted ontologies to reason-able forms, Data & Knowledge Engineering. 66 (2008) 147–162. doi:10.1016/j.datak.2008.02.002.

[14] C.J. Mungall, Obol: integrating language and meaning in bio-ontologies, Comparative and Functional Genomics. 5 (2004) 509–520. doi:10.1002/cfg.v5:6/7.

[15] M. Bada, L. Hunter, Enrichment of OBO ontologies, Journal of Biomedical Informatics. 40 (2007) 300–315. doi:10.1016/j.jbi.2006.07.003.

[16] J.T. Fernandez-Breis, L. Iannone, I. Palmisano, A.L. Rector, R. Stevens, Enriching the Gene Ontology via the Dissection of Labels Using the Ontology Pre-processor Language, in: P. Cimiano, H.S. Pinto (Eds.), Knowledge Engineering and Management by the Masses, Springer Berlin

Heidelberg, 2010: pp. 59–73. doi: 10.1007/978-3-642-16438-5_5.

[17] A. Gangemi, V. Presutti, Ontology Design Patterns, in: S. Staab, D. Rudi Studer (Eds.), Handbook on Ontologies, Springer Berlin Heidelberg, 2009: pp. 221–243. doi: 10.1007/978-3-540-92673-3_10.

[18] C.J. Mungall, M. Bada, T.Z. Berardini, J. Deegan, A. Ireland, M.A. Harris, et al., Cross-product extensions of the Gene Ontology, Journal of Biomedical Informatics. 44 (2011) 80–86.

[19] E. Pacheco, H. Stenzhorn, P. Nohama, J. Paetzold, S. Schulz, Detecting Underspecification in SNOMED CT Concept Definitions Through Natural Language Processing, in: Proceedings of the 2009 AMIA Annual Symposium, AIMA, San Francisco, CA, USA, 2009: pp. 492–496.

[20] D. Barrell, E. Dimmer, R.P. Huntley, D. Binns, C. O'Donovan, R. Apweiler, The GOA database in 2009 an integrated Gene Ontology Annotation resource, Nucleic Acids Research. 37 (2009) D396–D403. doi:10.1093/nar/gkn803.

[21] M. Quesada-Martínez, J.T. Fernández-Breis, R. Stevens, Lexical Characterization and Analysis of the BioPortal Ontologies, in: N. Peek, R.M. Morales, M. Peleg (Eds.), Artificial Intelligence in Medicine, Springer Berlin Heidelberg, 2013: pp. 206–215. doid: 10.1007/978-3-642-38326-7_31.

[22] M. Quesada-Martínez, J.T. Fernandez-Breis, R. Stevens, Enrichment of OWL Ontologies: a method for defining axioms from labels, in: L. Moss, D. Sleeman (Eds.), Proceedings of the International Workshop on Capturing and Refining Knowledge in the Medical Domain (KMED'2012), Galway, Ireland, 2012: pp. 5–10.

[23] A. Rector, S. Brandt, N. Drummond, M. Horridge, C. Pulestin, R. Stevens, Engineering use cases for modular development of ontologies in OWL, Applied Ontology. 7 (2012) 113–132. doi:10.3233/AO-2012-0107.

Metrics of an ontology θ	Number of classes	38571
	Number of labels	38533
	Type token ratio	7.05%
Metrics of an ontology θ based on its LRs	Number of lexical regularities	67
	Percentage of classes exhibiting *LRs*	85%
	Percentage of classes exhibiting *LRs* with matches	80%
	Mean number of external matches by regularity	141
	Lexical regularities with external matches	44

Table 1 Metrics used in the lexical analysis of the GO, using a 1% coverage

		FI[1%,2%)	FI[2%,3%)	FI[3%,4%)	FI[4%,5%)	FI[5%,100%]	MEAN
bpXcl	**CPE-c1**	39,06	24,76	52,57	13,40	23,09	**30,58**
	CPE-c2	52,39	27,70	54,94	15,48	26,96	**35,49**
	CPE-c3	10,49	5,66	6,08	6,47	7,61	**7,26**
	% Regularities	58,59	9,09	4,04	3,03	25,25	
		FI[1%,2%)	**FI[2%,3%)**	**FI[3%,4%)**	**FI[4%,5%)**	**FI[5%,100%]**	**MEAN**
ccXcl	**CPE-c1**	7,99	25,65	11,34	---	38,35	**20,83**
	CPE-c2	7,99	25,65	11,34	---	38,35	**20,83**
	CPE-c3	21,66	27,74	2,06	---	34,70	**21,54**
	% Regularities	68,89	22,22	2,22	0,00	6,67	
		FI[1%,2%)	FI[2%,3%)	FI[3%,4%)	FI[4%,5%)	FI[5%,100%]	**MEAN**
goXchebi	**CPE-c1**	35,28	41,64	48,21	52,97	33,69	**42,36**
	CPE-c2	36,09	41,89	49,36	54,49	34,74	**43,41**
	CPE-c3	16,19	20,76	22,62	37,81	16,44	**22,76**
	% Regularities	50,00	6,76	13,51	5,41	24,32	
		FI[1%,2%)	FI[2%,3%)	FI[3%,4%)	FI[4%,5%)	FI[5%,100%]	**MEAN**
mfXchebi	**CPE-c1**	44,95	49,43	45,73	76,62	53,38	**54,02**
	CPE-c2	45,07	50,10	46,93	77,72	53,86	**54,73**
	CPE-c3	6,02	15,02	8,57	60,95	29,62	**24,04**
	% Regularities	60,32	14,29	11,11	4,76	9,52	
Average % CPE	---	26,93	29,67	29,98	43,99	32,57	**31,48**

Table 2. Mean values of the three versions of the CPE metric grouped by θSxθE and by five frequency intervals.

	Average number of tokens matched in labels that exhibit lexical regularities	Average number of tokens NOT matched in labels that exhibit lexical regularities	Average % of NON-matched tokens
bpXcl	4.38	2.09	32.30
ccXcl	1.10	2.47	69.18
goXchebi	4.35	2.19	33.48
mfXchebi	1.70	2.42	58.73

Table 3 Comparison of the average number of tokens matched for those labels that exhibit regularities in the interval of coverage [1%, 100%]. Only those labels that exhibit lexical regularities and for which at least one token has been matched are considered.

	Classes in the enriched file		Classes in the source file		CPE Conditions	All decompositions						Without intra-decompositions						%Reduc. of fp
	Def	Prim	Def	Prim		P1	R1	F1-1	tp	fp	fn	P2	R2	F1-2	tp	fp	fn	
bpXcl	654	329	3954	62	CPE-c1	0,14	0,84	0,24	552	3402	102	0,14	0,84	0,24	552	3402	102	0%
			3961	279	CPE-c2	0,14	0,85	0,24	554	3407	100	0,14	0,85	0,24	554	3407	100	0%
			1339	1306	CPE-c3	0,24	0,49	0,32	322	1017	332	**0,45**	**0,49**	**0,47**	**322**	**392**	**332**	**61%**
						P1	R1	F1-1	tp	fp	fn	P2	R2	F1-2	tp	fp	fn	
ccXcl	25	36	195	9	CPE-c1	0,06	0,48	0,11	12	183	13	0,06	0,48	0,11	12	183	13	0%
			195	23	CPE-c2	0,06	0,48	0,11	12	183	13	0,06	0,48	0,11	12	183	13	0%
			318	251	CPE-c3	0,03	0,32	0,05	8	310	17	**0,08**	**0,32**	**0,13**	**8**	**87**	**17**	**72%**
						P1	R1	F1-1	tp	fp	fn	P2	R2	F1-2	tp	fp	fn	
goXchebi	2647	993	12408	1886	CPE-c1	0,17	0,80	0,28	2116	10292	531	0,17	0,80	0,28	2116	10292	531	0%
			12684	2455	CPE-c2	0,17	0,81	0,28	2148	10536	499	0,17	0,81	0,28	2148	10536	499	0%
			6005	3303	CPE-c3	0,34	0,77	0,47	2035	3970	612	**0,42**	**0,77**	**0,54**	**2034**	**2802**	**613**	**29%**
						P1	R1	F1-1	tp	fp	fn	P2	R2	F1-2	tp	fp	fn	
mfXchebi	3132	2746	4107	1388	CPE-c1	0,45	0,60	0,52	1867	2240	1265	0,45	0,60	0,52	1867	2240	1265	0%
			4154	1650	CPE-c2	0,45	0,60	0,52	1883	2271	1249	0,45	0,60	0,52	1883	2271	1249	0%
			1446	1058	CPE-c3	0,75	0,35	0,47	1083	363	2049	**0,76**	**0,35**	**0,47**	**1083**	**347**	**2049**	**4%**
Mean					CPE-c1	0,21	0,68	0,29	1137	4029	478	0,21	0,68	0,29	1137	4029	478	
					CPE-c2	0,21	0,68	0,29	1149	4099	465	0,21	0,68	0,29	1149	4099	465	
					CPE-c3	0,34	0,48	0,33	862	1415	753	**0,43**	0,48	0,41	862	907	753	
					TOTAL	0,25	0,62	0,30	1049	3181	565	0,28	0,62	0,33	1049	3012	565	

Table 4 Precision, recall and F1-Measure obtained in the comparison of our method and a reference method [18].