

Sapkota, K., Aldea, A., Younas, M., Duce, D. and Barnes-Alcantara, R. (2016) 'Automating the Semantic Mapping between Regulatory Guidelines and Organizational Processes', *Service Oriented Computing and Applications*, 10 (4), pp. 365-389.

DOI: <https://doi.org/10.1007/s11761-016-0197-2>

This document is the authors' Accepted Manuscript.

License: <https://creativecommons.org/licenses/by-nc-nd/4.0>

Available from RADAR: <https://radar.brookes.ac.uk/radar/items/e81389e1-8495-4558-a103-6e5550482156/1/>

Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners unless otherwise waved in a license stated or linked to above. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

Automating the Semantic Mapping between Regulatory Guidelines and Organizational Processes

Krishna Sapkota¹, Arantza Aldea¹, Muhammad Younas¹,
David a. Duce¹, Rene Banares-Alcantara²

¹Department of Computing and Communication Technologies, Wheatley Campus,
Oxford Brookes University, Oxford
ks_21285@hotmail.com {aaldea,m.younas,daduce}@brookes.ac.uk

²Department of Engineering Science, University of Oxford,
Parks Road, Oxford, OX1 3PJ
{rene.banares-alcantara}@eng.ox.ac.uk

Abstract. Mapping regulatory guidelines with organizational processes helps in Regulatory Compliance Management. Automating the mapping process can help in automating the overall compliance process. The existing approaches compute the mapping between different entities such as ontology mapping, sentence similarity, semantic similarity and regulation-requirement mapping. However, there has not been adequate research on the automation of the mapping process between regulatory guidelines and organizational processes. In this paper, we explain how Natural Language Processing and Semantic Web technologies can be exploited in order to map the regulatory guidelines and organizational processes. In particular, we explain how we can take advantage of the structures of regulation ontology and the process ontology in order to compute the similarity between a regulatory guideline and a process. Our methodology is validated using a case study in the Pharmaceutical industry, which has shown promising results.

Keywords: Semantic similarity, ontology mapping, information extraction, regulatory compliance management, regulation, text analysis.

1. Introduction

Regulatory Compliance Management (RCM) is a management process, which is implemented by an organization to ensure organizational compliance with the relevant requirements and expectations. Examples of requirements are regulatory or legal guidelines, and that of expectations are mandates, policies and guidelines for the organizations. Failure to maintain the RCM in organizations generally results in heavy penalties or organizations being sued or even suspension and closure. Managing compliance is an expensive process. For example, legislations, such as the Sarbanes-Oxley Act (SOX) imposed stringent compliance requirements, and organizations had to make heavy investments to meet the requirements [1].

Our research identified that early approaches to the RCM were largely manual. Managing the compliance manually is an arduous, extensive and error prone task. It requires expertise in the field, which costs heavy capital investments for organizations. As a solution, computer aided RCM systems¹²³⁴ have been developed. However, these systems are still experiencing various challenges to streamline and automate the process. One of the challenges being experienced by the systems is coping with the frequent changes in regulations. With every change in the regulations, the systems should identify the affected processes. Besides, these approaches are proprietary in the sense that the knowledge about the requirements and processes are embedded within the specific codes designed for specific domains and particular purpose. The proprietary knowledge is hard to share and re-use.

The recent approaches are concentrating on using Semantic Web technologies to reduce the manual work [2–11]. Representing regulatory and process knowledge in a standard, homogeneous

¹ <http://technet.microsoft.com/en-us/library/cc677002.aspx>

² <http://www.bwise.com/grc-challenges/regulatory-compliance>

³ <http://www-01.ibm.com/software/ecm/compliance/>

⁴ <http://accelus.thomsonreuters.com/products/accelus-compliance-manager>

and interoperable format can improve the updating processes and reusability. In particular, modeling the organizational processes in a process-ontology and regulatory guidelines in a regulation-ontology helps in the reusability of the knowledge. However, the semantic representation of the processes and regulations needs to be updated in circumstances such as (1) changes in the existing regulatory guidelines or (2) need of the processes to conform to regulations from other regulatory bodies or in other territories. In such cases, mapping of the new regulatory guidelines with the processes constitutes an important step towards updating the affected processes. The automation of the mapping process also contributes to the overall automation of RCM, which has not been explored by these approaches

The process of automatic mapping between regulatory guidelines and organizational processes comes with various research challenges. Firstly, there is a lack of a standard framework for mapping regulation and process ontologies. Secondly, there are ambiguities and complexities in the regulatory text. Thirdly, there is implicit information in the description of organizational processes. This paper tackles the first challenge of designing and developing an appropriate framework for the mapping. For this, we developed a framework, called RegCMantic framework, which automates the mapping between regulatory guidelines and organizational processes. This work is based on our previous works presented in [12, 13]. The contributions of the RegCMantic framework are outlined below.

- 1) **Algorithm to Identify Document Components and Predicting Document Structure:** A document contains various document components constituting the structure of the document. Some examples of the components are the title, paragraph, headers, and footers. In order to extract meaningful regulatory entities from the regulatory text, it is essential to identify the document components that contain regulatory guidelines. The RegCMantic framework has created some algorithms to identify these components and the document structure.
- 2) **Algorithm to Identify the Regulatory Guidelines:** From the document structure, it identifies the regulatory guidelines in the document.
- 3) **Algorithm to Identify Meaningful Entities in the Regulatory Guidelines:** Within the regulatory guidelines, this framework identifies the important regulatory entities such as the subject, object, action and obligation. Identification of the regulatory entities helps in relating the regulatory guidelines with organizational processes automatically.
- 4) **Tools for Constructing Regulatory Ontology and Representing the Regulatory Entities and Regulatory Guidelines in the Ontology:** An ontology to represent the regulatory guidelines and regulatory entities is essential for further processing the information in semantic means. This framework has constructed a regulatory ontology by extending an existing upper-level legal ontology.
- 5) **Computing Similarity between the Entities of Regulatory Guidelines and Organizational Processes:** In order to compute the similarity between a regulatory guideline and an organizational process, it is essential to identify the similarity between their entities. For example, determining the similarity between the subjects and the actions of a regulatory guideline and an organizational process helps in determining the similarity between the guideline and the process. This research computes the similarity between the entities in regulatory guidelines and organizational processes.
- 6) **Computing Similarity between Regulatory Statements and Organizational Processes:** A regulatory guideline contains one or more regulatory statement. Before relating the regulatory guideline to organizational processes, it is essential to relate its statement with the processes. This framework computes the relatedness of a statement with processes.
- 7) **Computing Similarity between Regulatory Guidelines and Organizational Processes:** Finally, this research determines the relation between a regulatory guideline and an organizational process.

Among the above contributions, the authors claim the originality in the followings:

- 1) **Identifying the Regulatory Guidelines and Entities in a Regulation Document:** The RegCMantic framework identifies the regulatory guidelines from various document structures. Furthermore, within the regulatory guidelines, it identifies the regulatory entities.
- 2) **Relating Regulatory Guidelines to Organizational Processes:** In order to facilitate the compliance manager with automation in the update process in RCM, this framework relates to the regulatory guidelines with organizational processes with the help of regulation entities and process entities.

The rest of the paper is organized as follows. The overview of the RegCMantic framework is described in Section 2. Section 3 and Section 4 elaborates the extraction and mapping parts of the framework respectively. Section 5 describes how the RegCMantic framework has evolved. Section 6 analyzes the results obtained from the case study. Section 7 compares the related works and finally Section 8 concludes the paper and identifies the future works.

2. The Framework

The RegCMantic framework comprises two parts: extraction and mapping (see Figure 1). These two parts have been described briefly and separately in our previous papers [12–15]. In the extraction part, the regulatory guidelines in different document formats, such as pdf, rtf and doc, are converted into a uniform XML format by identifying their document structures. The process of identifying the structure is called Document Structure Analysis (DSA). In the XML document, the regulatory guidelines and the regulatory entities are annotated; and this process is described as “Regulatory Entity Annotation”. Finally, the annotated entities are extracted and represented in an ontology, which is described as “Regulation Ontology Population”. In the mapping part, a regulatory statement is compared with an organized process in order to determine the level of relationship or similarity between them.

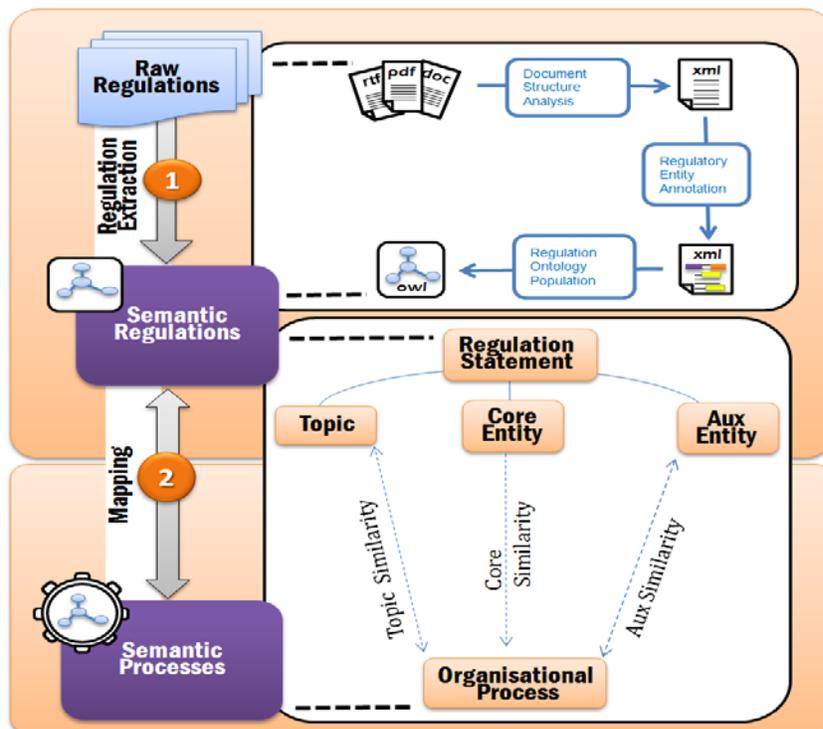


Figure 1. The RegCMantic framework

The comparison depends on three types of similarities: (i) topic similarity, (ii) core similarity and (iii) aux similarity. The three types of similarities are computed from the three types of regulatory entities in a regulation: (i) the topic entities, (ii) core entities and (iii) the aux entities. Each step in these two parts is described in the following sections.

3. Extraction Part

It is the first part of the framework and includes three steps: (i) representing the structure of the regulatory guidelines in XML format or Document Structure Analysis (DSA), (ii) extracting the meaningful entities from the text (see Figure 2) and (iii) representing the regulatory guidelines in ontology.

A regulatory document contains several documents-components, such as headers, footers, page numbers, footnotes, comments, titles and paragraphs. Identifying the documents-components is called DSA. In order to extract meaningful regulatory entities from regulatory text, it is essential to identify the document-components that contain regulatory guidelines. In particular, we need to identify regulatory-paragraphs and topics in order to extract regulatory entities. The regulatory-paragraphs or regulations are the paragraphs that impose some restrictions on organizational processes. The restrictions are usually imposed by using modal verbs, such as **must**, **should** and **may**. Once document-components are identified and regulatory entities are extracted, they need to be represented in a semantic format such as ontology. The following steps describe the process in detail.

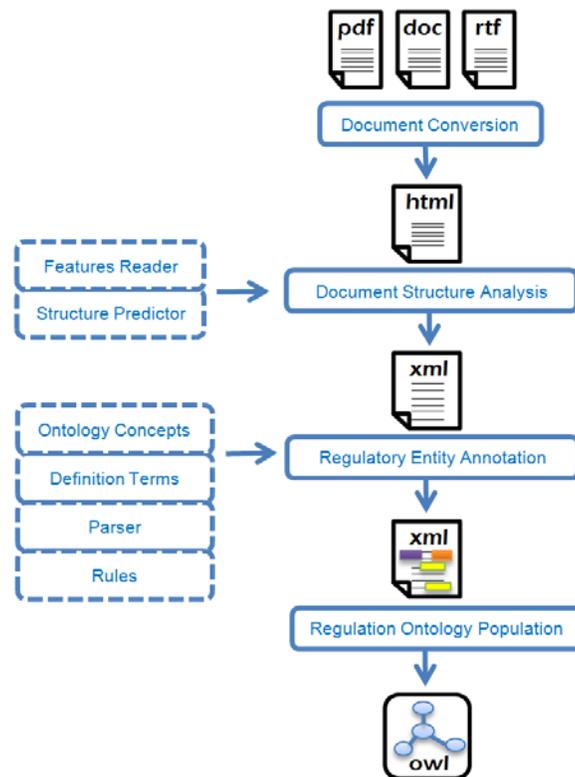


Figure 2. Regulatory entity extraction in the RegCMantic framework

3.1 Document Conversion

HTML is more favorable than the PDF document because at times, there are accessibility problems with Pdf documents. More users can access the file in its HTML version as compared to the accessibility of the PDF version of the important document. The audience must be kept in mind while uploading a pdf version of the document instead of the HTML version. Most of the times users find it difficult to access the PDF file directly as they miss out the search engine optimization.

However, there are a few rules for this conversion that can make this conversion fruitful.

1. Provide an HTML version for online use whenever possible, or;
2. Provide a summary web page to help search engines find the content, and let users decide if it is relevant.
3. You should clearly warn web users when a link will bring them to a PDF file instead of a web page.
4. You should create guidelines that identify when you should and shouldn't use PDF.

The regulatory guidelines are available in various document formats, such as PDF, DOC, HTML and XML (e.g. UK⁵, EU⁶ and USA⁷ regulations for the Pharmaceutical industries). Instead of developing processors for each format, the RegCMantic approach is to convert them into a single uniform processing format: HTML. An example of converting regulatory guidelines from PDF file format to HTML file format is provided in Figure 3 and Figure 4. There is a fair amount of tools, which convert documents into HTML format. In addition, there are tools available that convert documents into XML formats as well. However, in the RegCMantic framework (see Figure 2), the documents are first converted from various file formats to HTML and then to XML. They are not directly converted into XML because the direct conversion only converts the document into the XML file format; it does not identify the document-components. The RegCMantic framework represents the structure of a document explicitly, where each document-component is clearly identified and labeled. Converting the files into HTML format preserves the original information such as font features and the location of the text, which helps in the identification of the documents-components. Once identified, the document-components are represented in an explicit (and meaningful) format such as XML.

Figure 4 represents regulatory guidelines in the HTML format which was created by using an off the shelf HTML converter tool. In this figure, some spaces and tags have been removed to make it clearer to understand in this paper.

3.2 Document Structure Analysis (DSA)

In this step, the structure of the regulatory document is identified.

A document contains different types of text having different font-features such as font-size, font-style, font-strength and font-color. In this framework, the type of the text is called **Text-Type**. A document contains a set of text-type: $T = \{t_1, t_2, \dots, t_n\}$. For example, the font-size of the title of a document is larger than that of the text in the body; therefore, they can be regarded as two different text-types. For each text-type, a score is computed considering all the font-features and is called **Feature-Score**. The main influencing factor for the feature-score is the font-size. This means that the larger the font-size, the higher the feature-score. A document contains a set of feature-scores: $S = \{s_1, s_2, \dots, s_n\}$. A level is defined for each text-type based on its feature-score, and is called **Text-Level**. This means that the higher the feature-score, the higher the text-level. A

⁵ http://www.mhra.gov.uk/home/idcplg?IdcService=SS_GET_PAGE&nodeId=613

⁶ http://ec.europa.eu/health/documents/eudralex/cd/index_en.htm

⁷ <http://www.fda.gov/>

document contains a set of text-level: $L = \{l_1, l_2, \dots, l_n\}$ for a set of text-type. In the set of the text-levels, the order of the levels is: $l_1 > l_2 > \dots > l_n$.

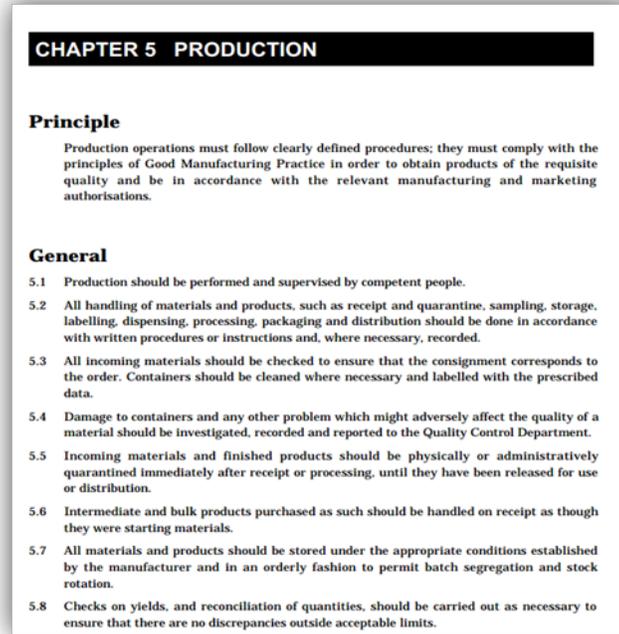


Figure 3. Example regulatory guidelines in the PDF file format

Example: In the text in Figure 3, there are three text-types t_1 , t_2 and t_3 representing chapter, section and paragraph respectively. The first line of text “Chapter 5 Production” has the highest feature-score:

$$\begin{aligned} s_1 &= \text{font-size} \times 10 + \text{font-bold} \\ &= 23 \times 10 + 2 \\ &= 232 \end{aligned}$$

The text in “Principal” and “General” has the second highest feature-score:

$$\begin{aligned} s_2 &= \text{font-size} \times 10 + \text{font-bold} \\ &= 20 \times 10 + 2 \\ &= 202 \end{aligned}$$

The text in the paragraphs starting with some numbers has the feature-score lower than the above two:

$$\begin{aligned} s_3 &= \text{font-size} \times 10 + \text{font-normal} \\ &= 13 \times 10 + 0 \\ &= 130 \end{aligned}$$

There are three feature scores and also three text- types. We have three feature-scores s_1 , s_2 and s_3 for three text-types t_1 , t_2 and t_3 respectively. Now we can assign levels: l_1 , l_2 and l_3 for t_1 , t_2 and t_3 respectively.

Similarly, a document has a set of **Document-Components**: which are denoted by $C = \{c_1, c_2, \dots, c_n\}$ such as chapter, section, sub-section, paragraph and page numbers. The document-components specify the structure of a document. Usually, they follow a hierarchical structure depending on the text-level of each text-type. In summary, each text-type is labeled with a text-level considering its feature-score, and each text-level is labeled with a document-component considering the document-component prediction algorithms.

```

1 <html>
2 <head>
3 <title>pg_0001</title>
4 <style type="text/css">
5 .ft1(font-style:normal;font-weight:bold;font-size:23px;font-family:Arial;color:#ffffff;)
6 .ft2(font-style:normal;font-weight:bold;font-size:20px;font-family:Times New Roman;color:#000000;)
7 .ft3(font-style:normal;font-weight:normal;font-size:13px;font-family:Times New Roman;color:#000000;)
8 </style>
9 </head>
10 <body>
11 <span class="ft1">CHAPTER 5 PRODUCTION</span>
12 <span class="ft2">Principle</span>
13 <span class="ft3">Production operations must follow clearly defined procedures; they must comply with the</span>
14 <span class="ft3">principles of Good Manufacturing Practice in order to obtain products of the requisite</span>
15 <span class="ft3">quality and be in accordance with the relevant manufacturing and marketing</span>
16 <span class="ft3">authorisations.</span>
17 <span class="ft2"> General</span>
18 <span class="ft3">5.1 Production should be performed and supervised by competent people.</span>
19 <span class="ft3">5.2 All handling of materials and products, such as receipt and quarantine, sampling, storage,</span>
20 <span class="ft3">labelling, dispensing, processing, packaging and distribution should be done in accordance</span>
21 <span class="ft3">with written procedures or instructions and, where necessary, recorded.</span>
22 <span class="ft3">5.3 All incoming materials should be checked to ensure that the consignment corresponds to</span>
23 <span class="ft3">the order. Containers should be cleaned where necessary and labelled with the prescribed</span>
24 <span class="ft3">data.</span>
25 <span class="ft3">5.4 Damage to containers and any other problem which might adversely affect the quality of a</span>
26 <span class="ft3">material should be investigated, recorded and reported to the Quality Control Department.</span>
27 <span class="ft3">5.5 Incoming materials and finished products should be physically or administratively</span>
28 <span class="ft3">quarantined immediately after receipt or processing, until they have been released for use</span>
29 <span class="ft3">or distribution.</span>
30 <span class="ft3">5.6 Intermediate and bulk products purchased as such should be handled on receipt as though</span>
31 <span class="ft3">they were starting materials.</span>
32 <span class="ft3">5.7 All materials and products should be stored under the appropriate conditions established</span>
33 <span class="ft3">by the manufacturer and in an orderly fashion to permit batch segregation and stock</span>
34 <span class="ft3">rotation.</span>
35 <span class="ft3">5.8 Checks on yields, and reconciliation of quantities, should be carried out as necessary to</span>
36 <span class="ft3">ensure that there are no discrepancies outside acceptable limits.</span>
37 <span class="ft3">5.9 Operations on different products should not be carried out simultaneously or consecutively</span>
38 <span class="ft3">in the same room unless there is no risk of mix-up or cross-contamination.</span>
39 <span class="ft3">5.10 At every stage of processing, products and materials should be protected from microbial and</span>
40 <span class="ft3">other contamination.</span>
41 <span class="ft3">5.11 When working with dry materials and products, special precautions should be taken to</span>

```

Figure 4. Regulatory guidelines converted into an HTML file format.

When the document-components are identified, they are represented in an XML file. In order to create the XML file, two processors are implemented: **Feature Reader** and **Structure Predictor** as shown in Figure 2.

The **Features Reader** identifies the document features such as font-style, font-weight, font-family, font-color and text-content. Reading the sufficient amount of document features helps in processing the index for each document-component.

Based on the documentary features, the **Structure Predictor** infers the components of the document. The paragraph is the main document structure, which helps to determine the regulation. Therefore, among the document components, at first, the paragraph is identified. Then, the other components are investigated based on their preceding text or label. A series of algorithms is implemented in order to predict the structure of the document; and the structure is presented in a user interface, where the user verifies the suggested structure.

3.2.1 Paragraph Prediction

In the set of text-levels L , each text-level l determines (i) how much text it contains (ii) how many sentences it has (iii) how many obligatory words, such as **must** and **should**, has, and (iv) how far its font-size is from the standard font - size of a paragraph text.

The prediction of a text as a paragraph requires computing the paragraph index of the text. Moreover, it needs to compute the indices of a sentence, text, obligation and deviation. A sentence index is the percentage of a sentence in a text-level. The text index of a text-level is the percentage of its text content. The obligation index of a text-level is the percentage of the obligatory words in the text. The deviation index of a text-level is the percentage of the distance of the text-level from the text-level of a standard paragraph. In general, the font-size of a paragraph is 12px; and it is not bold and italic. A paragraph index prediction is the average value of the weighted values of these four indices. The text in the text-level that has the highest paragraph index is regarded as the paragraph (see Algorithm 1).

Example: Following from the previous example, there are three text-types in Figure 3: $t1$, $t2$ and $t3$. The feature-score of a typical paragraph is computed as

$$\begin{aligned} s_p &= \text{font-size} \times 10 + \text{font-weight} \\ &= 12 \times 10 + 0 \\ &= 120. \end{aligned}$$

In this case, the closest feature-score to the paragraph is that of $t3$ (i.e. 130). This suggests that $t3$ is most likely to be a paragraph. Similarly, three other factors also suggest that $t3$ is a paragraph: the amount of text in $t3$ is the highest; $t3$ has the highest number of sentences; and there are more modal verbs in $t3$.

ALGORITHM 1. Paragraph Prediction

Input: L is a set of text-level in the document.
Output: L' is a new set of text- levels with the predicted text-level for the paragraph
Function: PREDICT-PARAGRAPH(L) returns L'

```

i = 0, lk = null
L = {l1, l2, ..., ln}
for each li ∈ L
    j = COMPUTE-PARA-PREDICTION-INDEX(li)
    if (j > i) then
        lk = li
        i = j
    end if
end for
lk.SET-COMPONENT(paragraph)
L' = L
return L'

```

3.2.2 Indicators Based Prediction

When the paragraph prediction is completed, the next process will predict the rest of the text-levels based on its preceding label or text also referred to as indicators. In many cases, the document-components with higher text-level, such as **part**, **chapter** and **section** are preceded with the relevant text such as “Chapter 5 Production” and “Section 5.3 Starting Materials”. When a text-level with an indicator is found, the document-component of the text-level is determined by the indicators. For example, if the text in the text-level l_1 starts with “Chapter”, then the document-component of the text-level l_1 will be set to `chapter` (see Algorithm 2).

Example: Following from the previous example, the $t3$ has been suggested as the paragraph in Figure 3. Now, we need to identify the document-component of $t1$ and $t2$. The text-type $t1$ is preceded with an indicator term “Chapter”, which suggests that $t1$ is a chapter.

ALGORITHM 2. Paragraph Based on the Indicator Text

Input: C is a set of document-components (document-structure). L is a set of text- level in the document.
Output: L' is a new set of text- levels in the document with document structure values computed from the preceding text
Function: PREDICT-COMPONENT-WITH-INDICATOR(C, L) returns L'

```

C = {c1, c2, .., cn}
L = {l1, l2, .., ln}
for each li ∈ L
  ci = GET-COMPONENT(li)
  text = GET-INDICATOR-TEXT(li)
  if (ci = null) then
    for each cj ∈ C
      if (text = cj) then
        ci = cj
      end if
    end for
  end if
end for
L' = L
return L'

```

ALGORITHM 3. Predicting the Remaining Structure of the Document

Input: C is a set of possible document-components (document-structure). L is a set of text-levels in the document.

Output: L' is a new set of text-levels in the document with document structure values computed from the preceding text

Function: PREDICT-REMAINING-COMPONENT(C, L) returns L'

```

C = {c1, c2, .., cn}
L = {l1, l2, .., ln}
for each li ∈ L
  ci = GET-DOCUMENT-COMPONENT(li)
  ci+1 = GET-DOCUMENT-COMPONENT(li+1)
  if (ci = null) then
    ci ∈ C
    ci = ci
  end if
  if (ci ≠ null or ci+1 = null) then
    for each cj ∈ C
      if (ci = cj) then
        ci+1 = cj+1
      end if
    end for
  end if
end for
L' = L
return L'

```

3.2.3 Prediction Based on Empirical Values

The predictions of the text-levels that have not been completed yet are computed based on the proximity of empirical values (see Algorithm 3). Based on the proximity, the algorithm predicts the closest document-component with respect to an empirically created hierarchical component set $C = \{c_1, c_2, .., c_n\}$. When there are many possible documents-components for a text-level, the document-component of the text-level is determined as the closest one to the highest predicted document-component.

Example: Following from the previous example, in Figure 3, t_1 and t_3 has been suggested as chapter and paragraph respectively. Now, we need to identify the document-component of t_2 . The empirical value suggests that the document-components between chapter and paragraph are section and sub-section. In this case, the document-component closest to chapter is a section. Therefore, it suggests that t_2 is a section.

The predicted document-structures are presented to users via a GUI. Users, then, are able to select, analyze and modify the suggested document-structures.

```

1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <document>
3  <meta>
4      <name>Eudralex</name>
5      <description>EU regulation for the pharmaceutical industry</description>
6      <body>EMEA</body>
7      <version>1.0</version>
8      <published_on>2007</published_on>
9  </meta>
10 <content>
11 <chapter title="CHAPTER 5 PRODUCTION">CHAPTER 5 PRODUCT
12 <section title="Principle">Principle
13 <paragraph paraNum=""> operations must follow clearly defined pro
14 </section>
15 <section title=" General"> General
16 <paragraph paraNum="5.1"> Production should be performed and s
17 <paragraph paraNum="5.2"> All handling of materials and products,
18 <paragraph paraNum="5.3"> All incoming materials should be check
19 <paragraph paraNum="5.4"> Damage to containers and any other p
20 <paragraph paraNum="5.5"> Incoming materials and finished produ
21 <paragraph paraNum="5.6"> Intermediate and bulk products purcha
22 <paragraph paraNum="5.7"> All materials and products should be st
23 <paragraph paraNum="5.8"> Checks on yields, and reconciliation of
24 <paragraph paraNum="5.9"> Operations on different products shoul
25 <paragraph paraNum="5.10"> At every stage of processing, product
26 <paragraph paraNum="5.11"> When working with dry materials and
27 <paragraph paraNum="5.12"> At all times during processing, all mat
28 <paragraph paraNum="5.13"> Labels applied to containers, equipm

```

Figure 5. An example of the regulatory guidelines represented in the XML representation format

3.2.4 XML Regulation

Following the earlier steps, the HTML document format is converted into XML (see Figure 5). The conversion is an important step since it identifies a different document-components in a document, and represents the document-components in an explicit format. When the document-components are explicitly labeled or represented, it helps in the extraction of specific entities from specific document-components. Note that, in rare situations, if regulators publish the regulation documents in a standard and explicit format, the previous two steps may not be necessary. However, this is not a common practice; and those stages constitute an important part of the process.

The most important document-component is paragraph because the regulatory guidelines are represented in paragraphs. A regulation-document contains several paragraphs; however, not all the paragraphs are regulatory guidelines. In this framework, a paragraph containing regulatory guidelines is called **regulation** or **regulation-paragraph**; and a sentence within in a regulation-paragraph is called **regulation-statement**.

3.3 Regulatory Entity Annotation

A regulation-statement contains regulation entities, such as subject, obligation and action, which helps to express regulatory requirements. A **subject** is a regulation-entity, upon which the requirements are imposed. For example, in a regulation-statement “*Equipment should be cleaned*”

after processing”, the word **Equipment** is the subject. In a regulation-statement, a subject can be equipment, substance, person, document or a process. The text in a regulation document contains some modal verbs such as **should, must** and **shall**. These modal verbs are the means of expressing the requirements of a regulatory guideline and are called **obligations**. The strength of the obligations may also vary from soft and medium to strong; for example, **shall, should** and **must** are the soft, medium and strong obligations respectively. An **action** is a regulation-entity that has to be performed in order to comply with some requirements and expectations. Usually, the action is the main verb in a sentence; however, sometimes the verb may be modified to different grammatical forms such as nouns and adjectives. In the example described above, **cleaned** is the action. The three entities subject, obligation and action are called core-entities. Beside the core-entities, there are other entities that express time, place, reason and quality, and they are called **auxiliary-entities** or **aux-entities**.

In the process of regulatory entity annotation, the RegCMantic framework identifies the regulatory constraints in organizational processes. The first task in this process is to identify the regulation-statements. In each regulation-statement, it annotates the regulation-entities. For the annotation, it uses four main components: natural language parser, ontology concepts, definition terms, and IE rules.

3.3.1 Natural Language Parser

Natural language parsers interpret a sentence in terms of its grammatical structure. In particular, it identifies grammatical units and their relationship in the sentence such as subject, verb, object, preposition and determiners (see Table 1). Breaking down a regulation-statement into subject-containing chunk, object-containing chunk, action-containing chunk and complementary chunk helps in identifying the regulation-entities in a sentence accurately. For example, if a concept or a term is identified in a regulation-statement, and the position of the concept or the term is located within a subject-containing chunk, it verifies that it is a subject. In this process, a parser is used with some rules to identify the special chunks such as condition-chunk, subject-chunk, obligation-chuck, action-chunk, complement-chunk, where-chuck, when-chunk, why-chunk and how-chunk.

3.3.2 Ontological Concepts

The ontological concepts defined in a domain are useful for IE. For example, in the Pharmaceutical industry, some concepts in the process ontology are *Equipment*, *Substance* and *Filtering*. Using these concepts, and their synonyms and hyponyms, the RegCMantic framework can identify meaningful entities in the regulatory guidelines. In order to achieve this, a list of concepts is created from the process ontology. Misleading concepts or the parts of the concepts should be removed. In this framework, these concepts are referred to as “Domain Specific Stop-Words”. Some examples of the domain specific stop-words in the Pharmaceutical industry, as in the OntoReg ontology, are *Action*, *Module*, *Entity* and *Domain* in *Equipment_Module*, *Physical_Entity*, *Abstract_Entity* and *Process_Domain* respectively. The stop-words are removed from the list of ontological concepts before using them for the annotation.

Table 1. An example of a parsed text

Natural Text	<i>Starting materials should only be purchased from approved suppliers named in the relevant specification and, where possible, directly from the producer.</i>
Parsed Text (Typed Dependencies)	<pre> amod(materials-2, Starting-1) nsubjpass(purchased-6, materials-2) aux(purchased-6, should-3) advmod(purchased-6, only-4) auxpass(purchased-6, be-5) root(ROOT-0, purchased-6) prep(purchased-6, from-7) amod(suppliers-9, approved-8) pobj(from-7, suppliers-9) partmod(suppliers-9, named-10) prep(named-10, in-11) det(specification-14, the-12) amod(specification-14, relevant-13) pobj(in-11, specification-14) cc(specification-14, and-15) dep(possible-18, where-17) dep(specification-14, possible-18) conj(specification-14, directly-20) prep(named-10, from-21) det(producer-23, the-22) pobj(from-21, producer-23) </pre>

3.3.3 Definition Terms

Regulatory guidelines are usually provided with definition terms. The definition terms in regulatory documents are also known as introductory terms or glossary, and they are provided at the beginning of the documents. The terms are provided with their definition and the context in which they are being used (see Figure 6). These terms help to understand the semantic of the regulatory guidelines, and help in the annotation of the regulatory entities in the text. Similar to the list of ontological concepts, a list of definition terms is created for the annotation.

3.3.4 Information Extraction Rules

Application of pattern matching rules is regarded as an established IE technique [16]. As an advancement on the regular expression technology, some rule specification languages are being used as state-of-art tools such as Common Pattern Specification Language (CPSL) [17]. Java Annotation Pattern Engine (JAPE) [18] is an example of implementation of the CPSL (see Figure 7). These rules typically have patterns on the left-hand-side (LHS) as their conditions, and actions to be performed on the right-hand-side (RHS). A typical example of the actions on the RHS is the annotation.

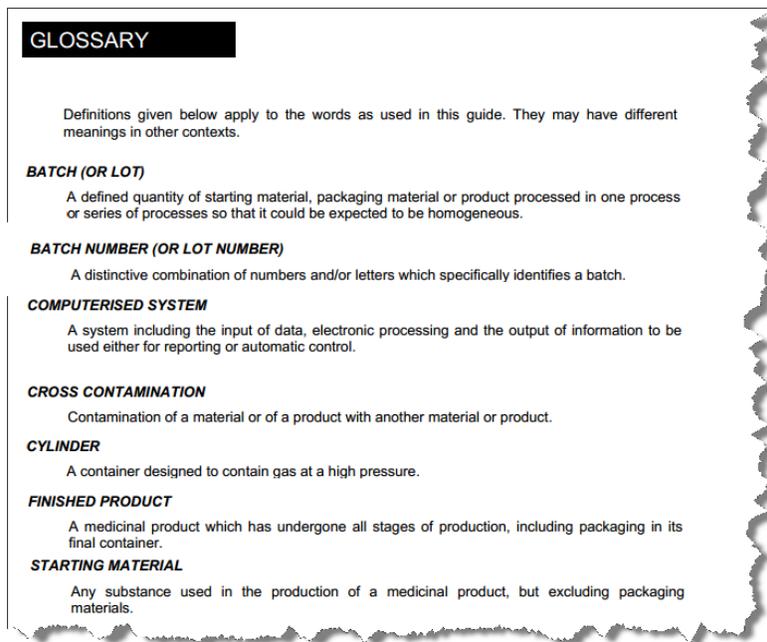


Figure 6. An example of definition terms

```

1  /*
2  * converts original markups ( annotation from the xml file) to standard gate annotations.
3  */
4  Phase: action_final
5  Input: action_container
6  Options: control = appelt
7
8  /* rule */
9  Rule: ActionRefiner
10 Priority:90
11 ((action_container)):ann
12 -->
13 {
14 // obtains the annotation
15 gate.AnnotationSet containerSet = (gate.AnnotationSet)bindings.get("ann");
16 gate.AnnotationSet containedSet = inputAS.getContained(containerSet.firstNode()
17 |.getOffset(), containerSet.lastNode().getOffset());
18 Set selectedSet = new HashSet();
19 selectedSet.add("rule_action");
20 selectedSet.add("definition_term");
21 selectedSet.add("extracted_term");
22 selectedSet.add("concept_ontology");
23 Iterator annIter = containedSet.get(selectedSet).iterator();
24
25 /* */
26 while (annIter.hasNext()){
27     gate.Annotation ann = (Annotation) annIter.next();
28
29     // get features from the annotation
30     String startNode = ann.getFeatures().get("startNode").toString();
31     String endNode = ann.getFeatures().get("endNode").toString();
32     String rule = ann.getFeatures().get("rule").toString();
33     String text = ann.getFeatures().get("text").toString();
34
35     // creating new annotation
36     gate.FeatureMap features = Factory.newFeatureMap();
37     features.put("rule","ActionRefiner");
38     int sNode = Integer.valueOf(startNode);
39     int eNode = Integer.valueOf(endNode);
40     features.put("startNode",sNode);
41     features.put("endNode", eNode);
42     features.put("text", text);
43     outputAS.add(ann.getStartNode(), ann.getEndNode(), "_ACTION", features);
44 }
45 }

```

Figure 7. An example of a JAPE rule

Therefore, the application of these rules helps to annotate the text if a specified pattern is met. In this step, the rules incorporate all the above annotations and create a new set of annotations and/or confirm the existing annotations.

In Figure 7, line 5 indicates that it takes input the annotation called “action_container”. Line 6 determines what type of option is applied to the rule. Line 9 defines the rule name, and line 10 defines the priority of the rule. In this example, it takes “action_container” as the annotations to process from the LHS. In the RHS, the annotations are processes using Java programming language. Lines 15-16 accept the annotations passed from the LHS. Similarly, lines 18-22 define the names of the annotations that need to be processed. Finally, lines 26 – 43 process the annotations and output the results.

In summary, ontological concepts help to identify the synonyms and hyponyms of the concepts in regulatory guidelines. Rules such as JAPE [18] help in specifying the grammar for pattern matching and incorporating the entities identified by ontological concepts. Similar to ontological concepts, the definition terms, provided by the regulatory document creators, can help in the identification of the regulatory terms, their synonyms and hyponyms. A lexical parser can be used to separate different grammatical units in a sentence; this helps in the identification of the important chunks in a sentence such as subject containing chunk and action containing chunk.

3.4 Semantic Representation of Regulatory Guidelines

The semantic representation is the population of regulatory ontology with the extracted regulatory entities such as subject, action, obligation and modifiers. Representing regulatory guideline in semantic models such as ontology helps in the automation of RCM. For the population, ontology with appropriate concepts is required. The ontology creation and population processes are described below.

3.4.1 Regulation Ontology Creation

In order to represent the regulatory guidelines semantically, a regulatory ontology called SemReg is created. It is recommended [19] that the ontology engineering should utilize the concepts of the existing ontologies in a similar domain and that of the upper ontologies. Therefore, the LKIF-Core ontology [20, 21] is considered for the SemReg engineering. The LKIF ontology is the recent development in the legal domain, and it has defined the appropriate level of concepts. These concepts are extended to the application level concepts and populated with the extracted entities. Although it is a core ontology, in order to adapt the concepts in the pharmaceutical domain, further concepts are created. Among the concepts created are *Subject*, *Obligation*, *Action*, *Regulation*, *Statement*, *Time*, *Place*, *Intention* and *Evaluative Expression*. Figure 8 shows the extension of the LKIF-Core concepts in the SemReg ontology. In this figure, big boxes with dark borders are the extended concepts and the other boxes are the concepts in LKIF-Core ontology (Please refer to ([22]) for detailed information about this ontology).

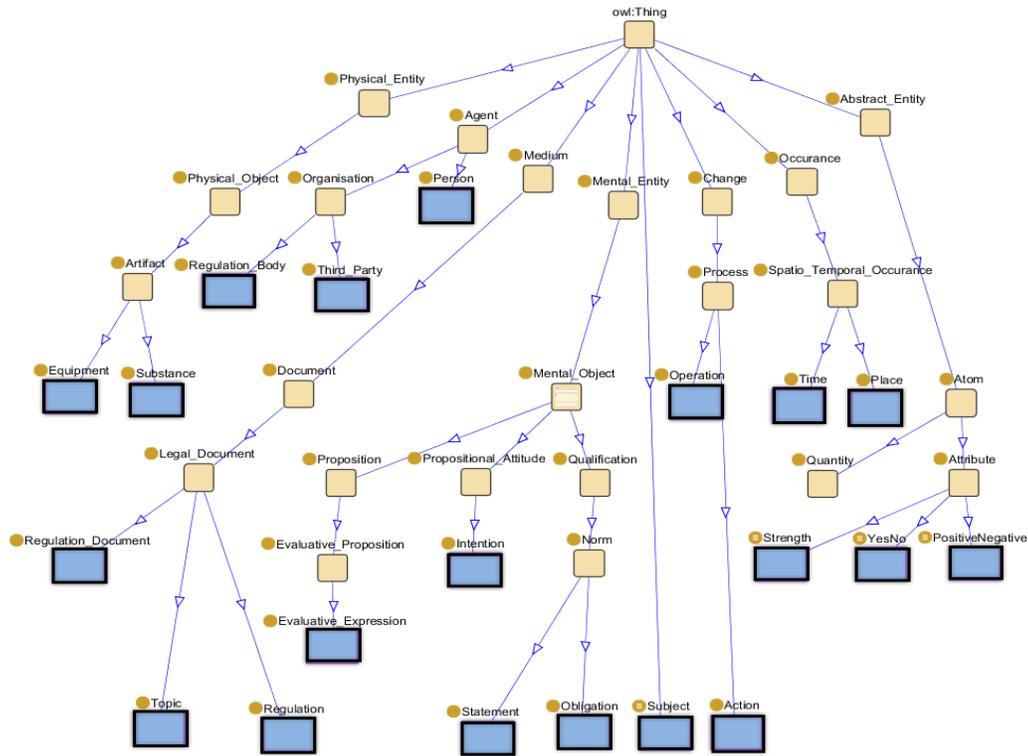


Figure 8. Concepts in the SemReg ontology

3.4.2 The SemReg ontology Population

Ontology population is a process where ontological classes are populated with their instances. After the identification and annotation of the regulatory entities in the regulatory guidelines, they are converted into the instances of the SemReg ontological classes (see Figure 9); and the regulatory guidelines are called semantic regulations. In other words, the semantic regulations are the regulations represented in an ontology. Semantic representation helps to process the regulations efficiently. The process of converting regulatory guidelines from text to semantic format has also been described briefly in [14]. In this framework, the purpose of making the regulations semantic is to identify the mapping between the regulations and the organizational processes.

Figure 9 displays the Protégé ontology engineering environment loaded with the SemReg ontology. On the left panel or class browser, it is showing hierarchies of classes preceded with circles. The classes also indicate the number of individuals they contain. For example, in the selected class *Statement*, there are 91 individuals. On the middle panel or instance browser, it is enlisting the individuals of the class *Statement*, which are indicated by purple diamonds. On the right panel or individual editor, it is displaying the properties of the individual *Eudralex_5.26_1* such as *id*, *description*, *isStatementOf*, *hasSubject*, *hasObligation* and *hasAction*.

4. Mapping Part

It is the second part of the framework, which identifies the relationship between the regulatory guidelines and the organizational processes by using the regulatory entities extracted from the first part of the framework. In particular, it needs two ontologies: a regulation ontology representing regulatory guidelines and a process ontology representing organizational processes. Creating process ontology is not the scope of this paper. Therefore, in order to explain the framework, an

off the self process-ontology, OntoReg, has been used, which was created by the Engineering Science Department in the University of Oxford [23]. In the OntoReg ontology, a validation-task (Task) is the smallest unit of an organizational process that is used for compliance checking. The two most important concepts associated with a validation-task are subject (Sub) and action (Act). Figure 11 displays a validation task S101_PurchasingTask, which is associated with a subject, SalicyclicAcid and an action, Purchasing101 respectively.

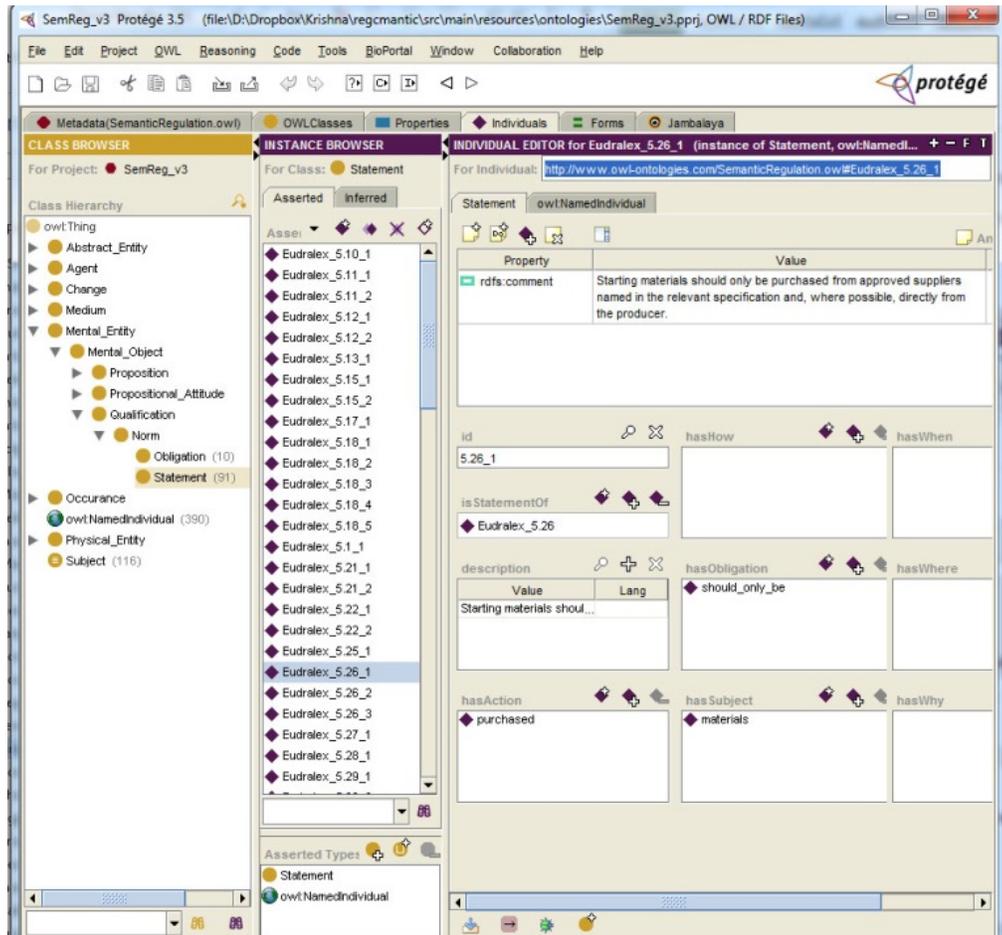


Figure 9. An example of the population of a regulatory ontology in Protégé

In the mapping part, three kinds of similarity scores are computed: (1) topic similarity, (2) core-entity similarity and (3) auxiliary-entity similarity. Figure 10 shows the computation of the three types of similarities. Figure 11 depicts a mapping between a regulation and a validation task in the regulation-ontology SemReg and the process-ontology OntoReg. The steps involved in the similarity computation are described separately in the following sub-sections.

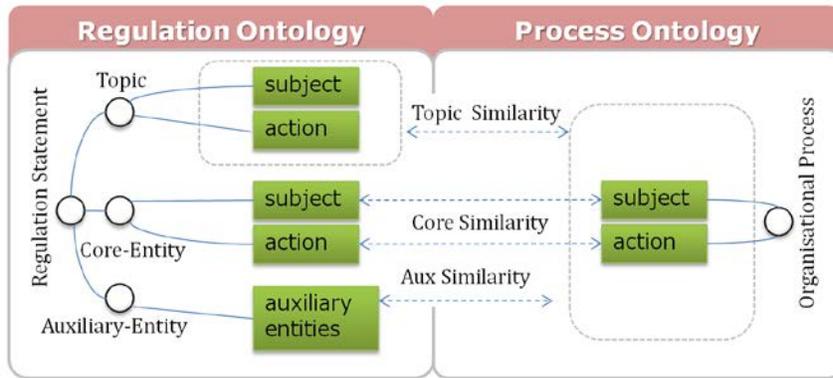


Figure 10. Three different types of similarity computations in the RegCMantic framework

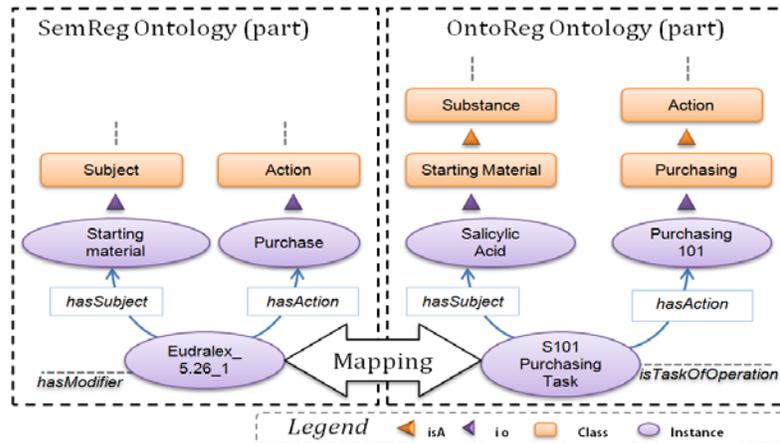


Figure 11. Mapping between a regulation and a validation task (process) using regulation and process ontologies

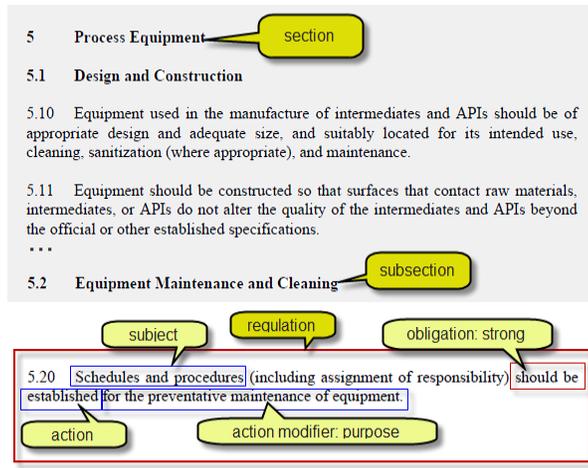


Figure 12. An excerpt from the Eudralex regulation showing regulatory entities

4.1 Conceptual Distance Computation

In the similarity computation, the similarity between an individual in the regulatory ontology and an individual in the process ontology is identified. Although some concepts look like very similar to each other in a general context, they can be different from each other in terms of their intentions in a specific context. For example, the concepts *substance* and *equipment* are closely related in the WordNet ontology; whereas in the OntoReg ontology, they are defined as different from each other. In the RegCMantic framework, the distance between two concepts in the OntoReg ontology is computed considering the axiom `disjointWith`. Currently, the value becomes 1 or 0 considering their disjointness; but in the future, we aim to consider the semantic distance computation algorithm [24] to determine the value. After the conceptual difference computation, a table is created; and each row in the table is represented by $\langle c_1, c_2, \delta \rangle$, where c_1 and c_2 are two concepts in the ontology and δ is the difference value between the concepts.

4.2 Three Types of Similarity Score Computation

In a regulation-ontology, regulations (Reg) are placed under a hierarchy of topics (Topic) such as part, chapter, section and subsection. A regulation contains one or more regulation statement (Stmt). A regulation-statement comprises core-entities (Core) and auxiliary-entities (Aux). The core-entities represent subject (Sub) and action (Act); and the auxiliary-entities represent extra information such as time, place and purpose. An example of the regulatory text depicting topics, core-entities and auxiliary entities, such as action modifier, is presented in Figure 12.

In this framework, three types of similarities are computed: (1) Topic similarity (Topic vs. Task), (2) Core-entity similarity (Core vs. Task) and (3) Auxiliary-entity similarity (Aux vs. Task).

In the core-entity similarity, each individual in a regulation-statement is compared with that of a validation-task. Since the individuals are associated with their subjects and actions, the similarity scores for the subjects and the actions are computed separately. The similarity score between two words is computed using the Lin-similarity [25] algorithm since it is widely used. The Lin-similarity considers the hierarchical structure of the terms in a lexical ontology, WordNet [26], and information content value (IC) of the terms in large corpora. It identifies the lowest common subsumer (LCS) between two compared words, computes the depth of the LCS from the root, measures the distance between the two compared terms via the LCS, and applies the IC values obtained from large corpora to compute the similarity measure. The subject-score computation results into a set of similarity-scores. The highest similarity-score among them is selected as the similarity-score of the subjects.

Algorithm 4 shows the similarity computation between a regulation-subject and a process-subject. Initially, the score is set to zero, which will be updated with the computed value. Consider there are two sets of subjects: S_r from the regulation-statement and S_t from the validation-task. Now, we compare each word in these sets. The difference-value δ is obtained from the difference-table, which is created from the process-ontology. If the two words are not defined as different in the process-ontology, only then, the similarity-score between them is computed.

Similarly, the action similarity is computed by comparing the action words associated with a regulatory-statement and a validation-task. After these two similarity scores are computed, the core-entity similarity is determined as the average of the subject score and the action score. The topic similarity is computed by comparing each word in the topic of a regulatory guideline with the subject and the action of a validation-task. Similar to the topic similarity computation, the auxiliary similarity-score is computed by comparing each word in the auxiliary-entities of a regulation-statement with the subjects and the actions of a validation-task.

ALGORITHM 4. Computing the Subject Similarity

Input: r is a regulation and t is a validation task.
Output: $score$ is the similarity score
Function: GET-SUBJECT-SCORE(r, t) **returns** $score$

```

 $score = 0$ 
 $S_1 = \{s_1 \mid s_1 \text{ is\_a\_subject\_in } stmt\}$ 
 $S_2 = \{s_2 \mid s_2 \text{ is\_a\_subject\_in } task\}$ 
for each  $s_i \in S_1$ 
  for each  $s_j \in S_2$ 
     $\delta = \text{GET-DIFFERENCE-VALUE}(s_i, s_j)$ 
    if ( $\delta < \theta$ ) then
       $score' = \text{SIMILARITY-SCORE}(s_i, s_j)$ 
      if ( $score' > score$ ) then
         $score = score'$ 
      end if
    end if
  end for
end for
return  $score$ 

```

ALGORITHM 5. Computing Aggregate Similarity Score

Input: S_{topic} , S_{core} and S_{aux} are topic-score, core-score and aux-score respectively
Output: S_{agg} is the aggregate similarity score of the three scores.
Function: GET-AGGREGATE-SCORE($S_{topic}, S_{core}, S_{aux}$) **returns** S_{agg}

```

 $S_{agg} = 0$ 
 $S_{tc} = \text{MAX}(S_{topic}, S_{core})$ 
if ( $S_{tc} \geq S_{aux}$ ) then
   $S_{agg} = S_{tc}$ 
else
   $S_{agg} = (S_{tc} + S_{aux})/2$ 
end if
return  $S_{agg}$ 

```

4.3 Aggregation of Similarity-Scores

Once the three similarity scores have been computed, the overall similarity between the regulation and the validation-task is determined by computing the aggregate similarity score from the three similarity scores.

The similarity aggregation algorithm (see Algorithm 5) emphasizes the importance of the topic-similarity and the core-similarity, as these similarities are more meaningful as compared to the aux-similarity. The aux-similarity considers every annotated word in the regulatory text, such as the annotations within exceptions, which can be sometimes misleading.

In the aggregation algorithm (see Algorithm 5), the maximum score between topic-score and core-score is chosen as the aggregate score. However, if the aux-score is the highest of all, the highest of the topic-score and the core-score is computed. Then, the average between the highest score and the aux-score is regarded as the aggregate score. The aggregation of the similarity scores has been simplified from its previous implementation [13]; and it has shown improved results.

4.4 Statement Similarity to Regulation Similarity Computation

The three types of similarity scores computed above are between a regulation-statement and a validation-task, not between a regulation-paragraph and a validation-task. As mentioned earlier, a regulation is composed of one or more statements. The overall similarity computed above is the similarity of a statement with a validation-task in the process ontology. Now, if a regulation contains more than one statement, it also contains a set of similarity scores; and the maximum

score in the set, i.e. $\text{SimReg} = \text{MAX}(\text{Sim}_{s1}, \text{Sim}_{s2}, \dots, \text{Sim}_{sn})$, is regarded as the similarity score between the regulation and the validation-task.

5. Baseline Framework vs Extended Framework

The framework has evolved during its implementation. In this paper, the initial framework is called Baseline Framework (BF) and the evolved framework is called Extended Framework (EF).

The extraction phase of the BF used only two components: ontological concepts and rules. Whereas, that of EF uses two additional components: lexical parser and definition terms. Use of lexical parser helps to separate the different chunks of the text in a sentence. These chunks help to identify the entities more accurately. The definition terms have also been found helpful to identify the entities more accurately. The mapping phase of the BF used only the core-similarity; whereas, the EF uses two additional similarities: topic-similarity and aux-similarity. It has been observed that the results of the EF outperformed that of the BF.

6. Result and Evaluation

6.1 Experimental Setup

In order to test the framework, we have used a case study in the Pharmaceutical industry in the EU, which is one of the most heavily regulated domains. The regulation governing this domain in the EU is the Eudralex⁸ regulation. As described earlier, the framework requires two ontologies: one for regulatory domain called SemReg and the other for process domain called OntoReg. A group of chemical engineers in the University of Oxford have developed the OntoReg ontology and been using it for checking compliance in real scenarios. The group has been regularly consulted for the requirements and validation of the framework.

In order to explain the results in this paper, a regulation, `Eudralex_5.22` in the SemReg ontology and a validation task, `FilterCleaningTask` in the OntoReg ontology have been selected.

Among the tools and technologies used for the framework are NLP and Semantic Web technologies. The interactions to the ontologies with JAVA have been carried out with the help of Jena API [27]. Jena with Pellet reasoner helps to trace the property values and infer new knowledge from the implicit knowledge in the ontologies. General Architecture for the Text Engineering (GATE) has been used as a platform to create and test various NLP related tools.

6.2 Extraction

This section presents the results and analysis of the extraction part of the framework. In particular, it analyses how the regulatory entities displayed in Figure 14 have been extracted from the regulatory guidelines in a pdf file in Figure 13.

The regulation, `Eudralex_5.22` (see Figure 13) [28] comprises only one regulation-statement and is preceded by an indicator number, 5.22. Each regulation is associated with some topics, which indicates the context of the regulatory guidelines expressed in the regulatory text. The topics, in this regulation, are “*Process Equipment*” and “*Equipment Maintenance and Cleaning*”.

⁸ <http://ec.europa.eu/health/files/eudralex/vol-4/pdfs-en/cap5en.pdf>

⁸ <http://www.europeanlawmonitor.org/what-is-guide-to-key-eu-terms/eu-legislation-what-is-an-eu-directive.html>

⁸ http://findlaw.co.uk/law/government/european_law/basics_european_law/500358.html

⁸ <http://www.innertemplelibrary.org.uk/news/FAQeu/DifferencesDirectives.htm>

EudraLex
The Rules Governing Medicinal Products in the European Union
Volume 4
EU Guidelines to
Good Manufacturing Practice
Medicinal Products for Human and Veterinary Use

Part II

Basic Requirements for Active Substances used as Starting Materials

..... (details hidden...)

5 Process Equipment

..... (details hidden...)

5.2 Equipment Maintenance and Cleaning

..... (details hidden...)

5.22 Equipment and utensils should be cleaned, stored, and, where appropriate, sanitized or sterilized to prevent contamination or carry-over of a material that would alter the quality of the intermediate or API beyond the official or other established specifications.

Figure 13. Regulation text in the Eudralex 5.22 regulation

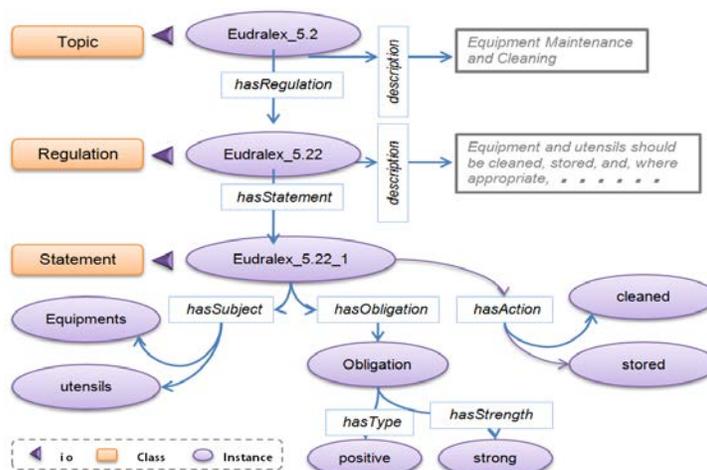


Figure 14. Eudralex 5.22 regulation represented in the SemReg ontology

The regulation paragraphs have been annotated using the process described in the framework and the extracted entities have been populated in the SemReg ontology. A graphical representation of the part of the ontology is shown in Figure 14. In this figure, the classes are Topic, Regulation and Statement, and their individuals are Eudralex_5.2, Eudralex_5.22 and Eudralex_5.22_1 respectively. The descriptions of the topic and the regulatory individuals are represented by a data-type property called description. A statement is a part of a regulation, which comprises the core and auxiliary entities. Among the core entities, Equipments and utensils are presented as the subjects; and cleaned and stored are actions. The subjects and actions relate to the statement via object properties: hasSubject and hasAction respectively. The obligation, along with its type and strength, has very little impact in the similarity computation; however, it acts as an indicator phrase in order to identify the subjects and the actions.

Analysis of the results of the baseline and extended frameworks is presented in Table 2. The precisions of the baseline framework and extended framework were determined as 0.89 and 0.96 respectively. The recall of the baseline framework and extended framework were found 0.78 and

0.86 respectively. The f-measures of the baseline and extended framework were computed as 0.83 and 0.91 respectively. This means that the extended framework performed better than the baseline framework did. The comparison between the BF and the EF presented that the current version outperformed the initial version. Although there is no change on identification of obligations, there is improvement in the identification of other core-entities: subject and action. On the extraction of auxiliary entities such as object, modifier and condition, it showed better improvement in the extended framework.

The first three rows in these tables present information about subject, obligation and action, which are described as the core-entities in this framework. The core-entities play a more important role in the regulation process mapping as compared to the auxiliary-entities. The both frameworks have identified all 52 obligations. This is because the framework has created an exhaustive list of obligatory words such as “should be”, “must” and “can be”. Regarding the actions, the extended framework showed a good f-measure, 0.97. Identification of an object, a modifier and a condition did not perform as well as that of the core-entities because the framework focuses on identification of the core-entities. A comprehensive algorithm to identify the auxiliary-entities remains recommended for the future-work of this research.

Table 2. Evaluation of the different types of annotations

Evaluation Measures	Precision		Recall		F-Measure	
	BF	EF	BF	EF	BF	EF
Annotation Types	BF	EF	BF	EF	BF	EF
Subject	0.89	0.96	0.78	0.86	0.83	0.91
Obligation	1.00	1.00	1.00	1.00	1.00	1.00
Action	0.88	0.96	0.90	0.99	0.89	0.97
Object	1.00	1.00	0.29	0.86	0.44	0.92
Modifier	0.58	0.88	0.27	0.54	0.37	0.67
Condition	0.50	1.00	0.22	0.67	0.31	0.80

6.3 Mapping

This section analyses the results of the three types of similarity scores and their aggregation. In particular, it describes a walk thorough example of mapping between the regulatory guideline, “Eudralex_5.22” and an organizational process, “FilterCleaningTask”.

6.3.1 A Regulatory Guideline in SemReg Ontology

In order to compute the three scores, the framework compares three types of entities: (i) topic, (ii) core-entities and (iii) aux-entities. An XML snippet representing these three types of entities, prior to the computation of the aggregate similarity-score, is presented in

Figure 15.

```

<regulation id="Eudrallex_5.22">
  <statement id="Eudrallex_5.22_1">
    <topic>
      <text>Chapter 5 Production, Process Equipment, Equipment
      <annotation>Process,Equipment,Cleaning</annotation>
      <bow>Equipment,Maintainance,Process,Equipment,Cleaning
    </topic>
    <core>
      <subject>equipment, utensils</subject>
      <action>cleaned,stored, sanitized, sterilized</action>
    </core>
    <aux>
      <text>5.22 Equipment and utensils should be cleaned, stored,
      <annotation>API,quality,material,Equipment,other established
      <bow>utensils,sanitized,sterilized,prevent,alter,intermediate,o
    </aux>
    </statement>
  </regulation>

```

Figure 15. Three types of entities in Eudrallex 5.22 regulation

The text in the topic comprises a combination of higher and lower topics related to the statement. Annotations are the most important entities in the text in terms of their meanings and their relation to the regulation and process. All the words except the stop-words are included in the bag of words (bow). The difference between the annotations and the bow is that the earlier ones are the concepts annotated from the domain ontology and the later ones are all the words remaining after removing the stop words. The core entities are collected directly from the subject and action properties of the statement in the SemReg ontology. The auxiliary entity collection is similar to the topic entity collection, where the annotations and the bag-of-words collection follow the same process. The text in the auxiliary entity is the text of the statement.

6.3.2 An Organisational Process in OntoReg ontology

In the process ontology, OntoReg, a validation-task is associated with a subject via an object property `hasPatient`, for which we have created an equivalent property called `hasSubject` for clarity. Similarly, an action is indirectly associated with a task, which can be determined by traversing through some object properties and individuals. In the `FilterCleaningTask`, the subject is `Filter101`, which is an individual of a class `Filter`. The class `Filter` is subsumed by the classes `ProcessingEquipment` and `Equipment`. The action for the `FilterCleaningTask` is defined implicitly. Having traversed through the property `isResponsibilityOf` and `performs`, it was inferred that `CleaningIndividual` is an individual of a class `Cleaning`. The class `Cleaning` is subsumed by its super-class `Action`.

In the mapping process, the regulatory entities such as topic, core and auxiliary entities are compared with the process entities such as subject, action and annotations. Figure 16 depicts the collection of subjects, actions and annotations of `FilterCleaningTask` just before the similarity score computation. The subjects are identified by the names and labels of the subject individual, classes and super-classes. Similarly, the action is determined by the names and labels of the action individual, their classes and super-classes. The annotation is the combination of these two types of entities.

```

<task id="FilterCleaningTask">
  <subject>filter,processing equipment, equipment</sub
  <action>cleaning</action>
  <annotation>filter, processing equipment, equipment,
</task>

```

Figure 16. Subject, action and annotations in Filter Cleaning Task.

6.3.3 Three Scores Computation

The comparison of the regulatory entities (topic, core and auxiliary) and the process entities (subject and action) produces three types of scores namely topic-score, core-score and aux-score.

For the core-score computation, the subject and action in the regulation-statement Eudrallex_5.22_1 were compared with the subject and action of the validation-task FilterCleaningTask respectively. In particular, the terms in regulatory subject “*equipment and utensils*” were compared with the terms in the process subject “*filter, processing equipment, equipment*”. This comparison produced a set of similarity between these two subjects. After the two separate comparisons, it produced two sets of scores: subject-score set (see Table 3) and action-score set (see Table 4).

Table 3. Similarity scores between regulatory and process subjects

Regulatory Subject	Process Subject	Similarity Score
Equipment	Filter	0.42
Equipment	Processing Equipment	0.54
Equipment	Equipment	1.00
Utensils	Filter	0.32
Utensils	Processing Equipment	0.27
Utensils	Equipment	0.48
Highest Similarity Score		1.00

Table 4. Similarity scores between regulatory and process actions

Regulatory Action	Process Action	Similarity Score
Cleaned	Cleaning	1.00
Stored	Cleaning	0.00
Sanitized	Cleaning	0.00
Sterilized	Cleaning	0.84
Highest Similarity Score		1.00

In the subject-score set {0.42, 0.54, 1.00, 0.32, 0.27, 0.48} the highest score is determined as 1.00. Therefore, 1.00 was set as the similarity score between the sets of subjects in the regulation-statement, Eudrallex_5.22_1 and the process, FilterCleaningTask. Similarly, in the action-score set {1.00, 0.00, 0.00, 0.84, 1.00} the highest score was found as 1.00. Therefore, the similarity score between the sets of actions in the regulation-statement, Eudrallex_5.22_1 and the process, FilterCleaningTask was set as 1.00. Then, the average score between the subject-score and action-score, 1.00 was determined as the core-score.

In the topic-score computation, the terms, “*Equipment, Maintenance, Process, Equipment, Cleaning*” in the bow of topic in the regulation-statement, Eudrallex_5.22_1 were compared with the terms, “*filter, processing equipment, equipment, cleaning*” in the annotation of FilterCleaningTask (see Table 5). The highest similarity score between the term “*Equipment*” in regulation and the terms “*filter, processing equipment, equipment, cleaning*” in the process was found as 1.00. Similarly, the highest similarity scores of “*Maintenance*”, “*Process*”, “*Equipment*” and “*Cleaning*” with respect to their comparison with the terms in process annotations were found as 0.73, 0.56, 1.00 and 1.00 respectively. Then, the average of these scores, 0.86 was determined as the topic-score between the regulation-statement, Eudrallex_5.22_1 and the process, FilterCleaningTask.

Table 5. Similarity scores between a regulatory topic and a process

Regulatory Topic	Process Annotation	Similarity Score
Equipment	Filter	0.42
Equipment	Processing Equipment	0.54
Equipment	Equipment	1.00
Equipment	Cleaning	0.06
Highest Similarity Score		1.00
Maintenance	Filter	0.00
Maintenance	Processing Equipment	0.12
Maintenance	Equipment	0.00
Maintenance	Cleaning	0.73
Highest Similarity Score		0.73
Process	Filter	0.08
Process	Processing Equipment	0.56
Process	Equipment	0.12
Process	Cleaning	0.40
Highest Similarity Score		0.56
Equipment	Filter	0.42
Equipment	Processing Equipment	0.54
Equipment	Equipment	1.00
Equipment	Cleaning	0.06
Highest Similarity Score		1.00
Cleaning	Filter	0.00
Cleaning	Processing Equipment	0.00
Cleaning	Equipment	0.00
Cleaning	Cleaning	1.00
Highest Similarity Score		1.00
Average of the Highest Similarity Scores		0.86

The computations of `aux-score` is similar to that of `topic-score`. In the `aux-score` computation, the terms, “*utensils, sanitized, sterilized, prevent, alter, intermediate, official, API, quality, material, equipment...*” in the bow of `aux` in the regulation-statement, `Eudrallex_5.22_1` were compared with the terms, “*filter, processing equipment, equipment, cleaning*” in the annotation of `FilterCleaningTask`. We also carried out the highest similarity score computation and the average of the highest similarity score computation. Then, the `aux-score` between the regulation-statement, `Eudrallex_5.22_1` and the process, `FilterCleaningTask` was computed as 0.42. A part of an XML file representing the three scores computed between the regulation `Eudrallex_5.22` and the process `FilterCleaningTask` is provided in Figure 17.

```

2 <mapping mapping_id = "mid_133">
3   <reg_id>Eudrallex_5.22</reg_id>
4   <stmt_id>Eudrallex_5.22_1</stmt_id>
5   <task_id>FilterCleaningTask</task_id>
6   <topic_score>0.86</topic_score>
7   <core_score>1.00</core_score>
8   <aux_score>0.42</aux_score>
9   <final_score>1.00</final_score>

```

Figure 17. The three types of similarity scores between `Eudrallex_5.22` and `FilterCleaningTask`

6.3.4 Aggregating the similarity scores

Having computed the three types of similarity scores between the regulation and validation task, the next step was to compute the aggregate similarity between the pairs. In the earlier section, the `topic-score`, `core-score` and `aux-score` were computed as 0.86, 1.00 and 0.42 respectively. In the aggregation algorithm, the maximum score between `topic-score` and `core-score` was computed as:

$$S_{tc} = \text{MAX}(S_{\text{topic}}, S_{\text{core}}) = \text{MAX}(0.86, 1.00) = 1.00$$

Where, S_{tc} is the maximum score between `topic-score`, S_{topic} and `core-score`, S_{core} . In this case, the S_{tc} is greater than the `aux-score`, S_{aux} . Hence, the final similarity score between the regulation-statement, `Eudraxlex_5.22_1` and the validation-task, `FilterCleaningTask` was determined as 1.00, which was represented as the `final-score`. Then, an XML file, containing all the three scores and the aggregate score between regulation-statements and processes, was generated. A part of the XML file is shown in Figure 17.

6.3.5 Evaluation of the Mapping Result

The OntoReg ontology contains a set of mapping between Eudraxlex regulations and validation-tasks. In particular, each validation task is associated with one or more regulations, and each regulation is related with one or more validation tasks, called existing mapping. The existing mappings were created by the experts manually. A subset of existing mapping collected from the OntoReg is depicted in Figure 18, where line number 2 indicates that there is a mapping between the regulation `Eudraxlex_5.22` and the validation-task `FilterCleaningTask`. The list in Figure 18 was created by using the values of the object-property `isRegulationOf` of individuals under the concept `Regulation`.

The mappings between a regulation and a validation task generated by the RegCMantic framework is referred to as computed mapping. A subset of computed mappings is shown in Figure 19. The line number 8 indicates that there is a mapping between the regulation, `Eudraxlex_5.22` and the validation-task, `FilterCleaningTask`.

As stated above, a regulation comprises one or more regulation-statements; the final score computed above is the similarity score between a statement and a validation-task. Therefore, the similarity score computation created a set of final similarity scores between the regulation and the validation-task; and the highest score was regarded as the similarity score between the regulation and the validation task.

```

1 mapping_id, reg_id, task_id, score, accuracy
2 em_1, Eudralex_5.22, FilterCleaningTask, ,
3 em_2, Eudralex_5.22, T102CleaningTask, ,
4 em_3, Eudralex_5.22, T101CleaningTask, ,
5 em_4, Eudralex_5.22, T101CleanlinessTestTask,
6 em_5, Eudralex_5.22, T102CleanlinessTestTask,
7 em_6, Eudralex_5.22, FilterCleanlinessTestTask
8 em_7, Eudralex_8.14, ReactionYieldTestTask_1,
9 em_8, Eudralex_8.14, InvestigationTask_1, ,
10 em_9, Eudralex_5.21, FilterCleaningTask, ,
11 em_10, Eudralex_5.21, FilterCleanlinessTestTask,
12 em_11, Eudralex_5.21, T101CleanlinessTestTask,
13 em_12, Eudralex_5.21, T102CleanlinessTestTask,
14 em_13, Eudralex_5.21, T101CleaningTask, ,
15 em_14, Eudralex_5.21, T102CleaningTask, ,
16 em_15, Eudralex_5.31, StartingMaterialTestTask
17 em_16, Eudralex_5.26, PharmaSupplierAssess_1,
18 em_17, Eudralex_5.26, StartingMaterialPurchase

```

Figure 18. An excerpt of the existing mappings between regulations and validation tasks

```

1 mapping_id, reg_id, task_id, score, accuracy
2 mid_1, Eudralex_5.21, FilterCleaningTask, 1.0,
3 mid_2, Eudralex_5.21, T101CleaningTask, 1.0,
4 mid_3, Eudralex_5.21, T102CleaningTask, 1.0,
5 mid_4, Eudralex_5.21, FilterCleanlinessTestTask, 1.0,
6 mid_5, Eudralex_5.21, T101CleanlinessTestTask, 1.0,
7 mid_6, Eudralex_5.21, T102CleanlinessTestTask, 1.0,
8 mid_133, Eudralex_5.22, FilterCleaningTask, 1.0,
9 mid_134, Eudralex_5.22, T101CleaningTask, 0.97562,
10 mid_135, Eudralex_5.22, T102CleaningTask, 0.97562,
11 mid_153, Eudralex_5.26, StartingMaterialPurchase_1, 0.9
12 mid_136, Eudralex_5.22, FilterCleanlinessTestTask, 0.92
13 mid_137, Eudralex_5.22, T101CleanlinessTestTask, 0.8991
14 mid_138, Eudralex_5.22, T102CleanlinessTestTask, 0.8991
15 mid_165, Eudralex_5.31, StartingMaterialTestTask_7, 0.8

```

Figure 19. An excerpt of computed mapping between regulations and validation tasks

In order to evaluate the result of the algorithm, the set of manual mappings was considered as the standard mappings, which were compared with the set of computed mappings; and the comparison generated three types of mappings: the correct mappings, incorrect mappings and missing mappings. These three types of mapping are used to compute the standard evaluation techniques called precision, recall and f-measure. Precision, recall and f-measure are popular in Information Retrieval (IR) and have been borrowed in several other domains, as well. Since the authors have not come across the frameworks that map regulatory guidelines with organizational processes, the evaluation of the framework was carried out by observing the precision, recall and f-measure only.

The selection of the mappings also needs to define the minimum threshold, τ . The value of τ was set as 0.85; only the mappings with the score 0.85 or above were selected as the accepted mappings; and the rest of the mappings were discarded. Figure 20, Figure 21, and Figure 22 show the precision, recall and f-measure of the mapping results respectively. The value of τ was set as 0.85 because it was found the optimum threshold after repeated observation, which can be seen in Figure 22.

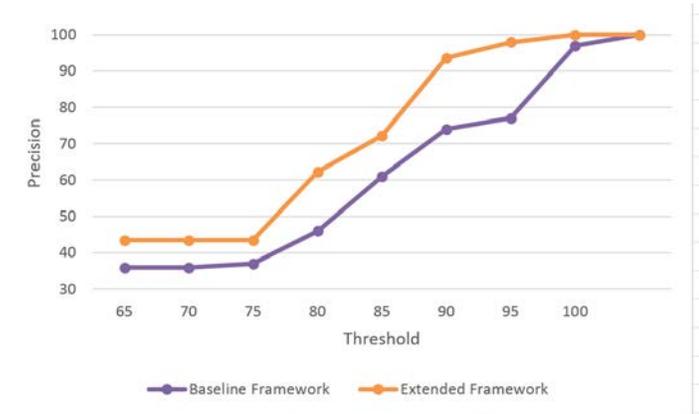


Figure 20. Precisions of the mappings in different thresholds

The base line framework refers to the similarity score computed by using only the core scores and extended framework refers to the score generated by using the topic, core and auxillary scores.

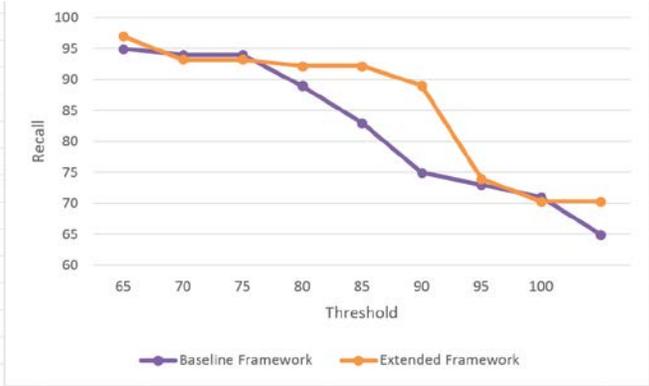


Figure 21. Recalls of the mappings in different thresholds

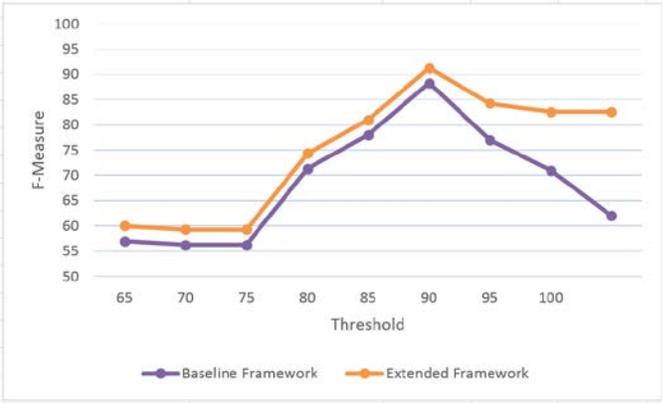


Figure 22. F-Measure of the mappings in different thresholds

7. Related Work

The RegCMantic framework is related to the approaches that 1) extract regulatory entities and 2) map regulations with organizational processes in relation to RCM. Some part of these approaches are presented below.

7.1 Related Extraction Approaches

Extraction of rights and obligation by the extension of the *Cerno* framework [29] is related to the extraction part of this work, since it identifies the requirements by detecting the presence of normative phrases. However, in contrast to the application of shallow parser in the *Cerno* framework, the authors believe that deep parsers are more useful in the more grammatically correct text such as regulation. Furthermore, the *Cerno* framework is applicable to more structured text such as legalese and needs engineers to annotate the regulatory text. In contrast, the extraction part of the RegCMantic framework can be applied to the text with no explicitly defined document-structure and the annotation process is automatic. The exception extraction by Gao *et al.*, [30] and regulation-entities extraction in Mu *et al.* [31] are also related to the extraction part. However, the former is only confined to the extraction of exception with limited indicator terms. The latter is more related to this work, since it extracts a variety of regulation-entities such as subject, subject-modifier, object, object-modifier, action, location, time, manner and constraints. Furthermore, it also uses a deep parser and a list of terms. However, it has not been mentioned how to deal with the text with implicit document-structure. Moreover, the terms are defined by the experts manually, which, in contrast, is extracted automatically in the RegCMantic framework.

7.2 Related Mapping Approaches

This section reviews the existing work related to the RegCMantic mapping approach. Examples of the related work include the similarity techniques in Business Process Modeling (BPM), sentence similarity, word similarity, ontology mappings and conceptual distance.

7.2.1 BPM Similarities

BPM represents the processes of an enterprise so that they can be easily analyzed and improved. There are similarity approaches that relate a process to another process [32–34] or a controlled objective [4, 10, 35]. The controlled objectives are the objectives created by considering the standards and the regulatory guidelines related to the business processes. The similarity techniques used to relate these components could be considered as related to this work.

The similarity in the elements of two processes was determined in [32] with two kinds of matching: graph matching and pure lexical matching. The redundant or duplicate elements in processes were identified in [33] by using ontology matching technology. The similarity between two processes were identified in [34] by extracting annotations from the data schema and templates associated with the processes. However, these approaches do not relate regulatory guidelines with organizational processes.

Creating controlled objectives from the regulations and the processes, and relating the objectives were explored in [4, 35]. Similarly, the regulations were represented in a rule-based logic, FCL and the processes were represented in BPMN, and annotated to align the processes with the regulations [10]. However, it has not been explained how they were related, since their focus was to determine the non-compliance in the processes.

7.2.2 Sentence Similarity

In [36], sentence similarity is computed using align-heuristics where noun, verb, adjective, adverb and numbers are aligned; and the approach was inspired by the popular sentence alignment algorithm in [37]. The decomposition of sentences into different entities for the similarity measure is similar to the RegCMantic framework; however, this can be only applied to compute the sentence similarity. The sentence matching based on the Bag of Words (BoW) algorithm was applied in [38] in order to determine the answer similarity. A BoW is an unordered collection of

words, which does not consider the grammar and the order of the words. It has been predominantly used in Information Retrieval (IR) in order to classify the pages. In the similarity computation, each word in a BoW is compared with the words in the other BoW. The computation of similarity of words in two sentences is related to this work; however, it is only applicable to compare sentences. Similarly, a pilot for similarity in SemEval competition has described the similar algorithms for the sentence similarity which also requires training and testing sentences[39].

7.2.3 Ontological Concept and Relation Similarity

Conceptual distance and similarity computation in ontologies are also related to this work. The use of weight allocation and node routing table in order to compute semantic distance between two concepts in an ontology [24] is related to the RegCMantic framework. In [40], a graph based similarity is computed considering various types of ontological properties and the depth of the concepts. In [41], two ontologies have been defined in order to determine similarity of a new event with an existing event. The similarity computed using WordNet similarity is related to this work; however, it requires that both ontological concepts and individuals designed and populated by the domain expert manually. In this framework, regulatory ontology is populated automatically from the text in the regulatory guidelines.

7.2.4 Combined Similarities

The work presented in [42] applies a combination of similarity approaches in order to determine similarity between contents of two television programs. The most related part in this framework is the computation of the similarity of topics and the text in the television program synopsis. However, it is only applicable if both compared entities contain hierarchy and text description in the sentences. The RegCMantic framework can be applied to determine the similarity where the processes are represented in ontological concepts, and the regulatory guidelines are represented in an unstructured text format; and the regulatory entities are populated in a regulatory ontology automatically [14].

8. Conclusion and Future Work

Mapping regulatory guidelines with organizational processes becomes crucial when there are changes in the guidelines, or the organizational processes need to follow the guidelines from different policy makers. Various extraction and similarity algorithms are closely related to the RegCMantic framework. However, they are not directly related to the mapping between the guidelines and processes. Therefore, there is a greater need for efficient algorithms that can map regulations with processes. This paper has presented RegCMantic framework, which identifies the regulatory entities automatically in order to map the regulatory guidelines with organizational processes. It has computed three typed of similarity scores: (1) topic similarity, (2) core-entity similarity and (3) auxiliary-entity similarity. The framework considers the ontological structures in order to compute the similarity scores. The case study carried out in the Pharmaceutical industry has demonstrated some promising results.

For the future works, experimenting with some iteration in the extraction part of the framework or combining the rule-based approaches with machine learning techniques are recommended in order to improve the results of the regulatory entity extraction. Currently the RegCMantic framework only extracts the regulatory entities; the mapping process may be improved by extracting the process entities from the annotations in the organizational processes as well.

References

1. Zhang, I.X.: Economic Consequences of the Sarbanes-Oxley Act of 2002. *J. Account. Econ.* 44, 74–115 (2007).
2. Goedertier, S., Vanthienen, J.: Designing Compliant Business Processes from Obligations and Permissions. *Proceedings of 2nd Workshop on Business Processes Design (BPD'06)*. pp. 5–14. Springer-Verlag, Vienna (2006).
3. Breaux, T.D., Vail, M.W., Antón, A.I.: Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations. *Proceedings of 14th IEEE International Requirements Engineering Conference (RE'06)*. pp. 49–58. IEEE Computer Society, Minneapolis (2006).
4. Sadiq, S., Governatori, G.: A Methodological Framework for Aligning Business Processes and Regulatory Compliance. *Handbook of business process management: 2. Strategic alignment, governance, people and culture*. pp. 159–176. Springer-Verlag Berlin Heidelberg, Berlin & Heidelberg (2010).
5. Logrippo, L.: Requirements and compliance in legal systems: a logic approach. *Requirements Engineering and Law, 2008. RELAW '08*. pp. 40–44. IEEE Computer Society Press, Barcelona, Spain (2008).
6. Ghanavati, S., Amyot, D., Peyton, L.: Towards a Framework for Tracking Legal Compliance in Healthcare. *Proceedings of the 19th International Conference on Advanced Information Systems Engineering (CAiSE'07)*. pp. 218–232. Springer-Verlag Berlin, Trondheim, Norway (2007).
7. Haider, S.I.: Validation standard operating procedures. Informa Healthcare, New York (2006).
8. Liu, Y., Muller, S., Xu, K.: A Static Compliance-Checking Framework For Business Process Models. *IBM Syst. J.* 46, 335–361 (2007).
9. Elgammal, A., Turetken, O., Van Den Heuvel, W.-J., Papazoglou, M.P.: Using Patterns for the Analysis and Resolution of Compliance Violations. *Int. J. Coop. Inf. Syst. IJCIS.* 21, 31–54 (2012).
10. Governatori, G., Shek, S.: Rule Based Business Process Compliance. *CEUR.* 874, 1 – 8 (2012).
11. Sadiq, S., Governatori, G.: Managing Regulatory Compliance in Business Processes. In: Brocke, J. and Rosemann, M. (eds.) *Handbook on Business Process Management 2*. pp. 159–175. Springer-Verlag, Berlin (2010).
12. Sapkota, K., Aldea, A., Younas, M., Duce, D.A., Banares-Alcantara, R.: Towards Semantic Methodologies for Automatic Regulatory Compliance Support. *Proceedings of the 4th workshop on Workshop for Ph.D. Students in Information & Knowledge Management (PIKM '11)*. pp. 83–86. ACM Press, Glasgow (2011).
13. Sapkota, K., Aldea, A., Younas, M., Duce, D.A., Banares-Alcantara, R.: RP-Match: A Framework for Automatic Mapping of Regulatory Guidelines with Organizational Processes. *The 10th IEEE International Conference on e-Business Engineering (ICEBE 2013)*. pp. 257–264. IEEE Computer Society Press, Coventry, UK (2013).
14. Sapkota, K., Aldea, A., Younas, M., Duce, D.A., Banares-Alcantara, R.: Semantic-ART: a framework for semantic annotation of regulatory text. *Proceedings of the Fourth Workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR '11)*. pp. 23–24. ACM Press, Glasgow (2011).
15. Sapkota, K., Aldea, A., Younas, M., Duce, D.A., Banares-Alcantara, R.: Extracting Meaningful Entities from Regulatory Text. *Proceedings of the Fifth*

- International Workshop on Requirements Engineering and Law (RELAW '12). pp. 29–32. IEEE Computer Society Press, Chicago (2012).
16. Sarawagi, S.: Information extraction. *Commun. ACM.* 1, 261–377 (2007).
 17. Appelt, D.E., Onyshkevych, B.: The common pattern specification language. TIPSTER Worskhop (TIPSTER '98). pp. 23–30. Association for Computational Linguistics, Baltimore, Maryland (1998).
 18. Thakker, D., Osman, T., Lakin, P.: GATE JAPE Grammar Tutorial, [http://gate.ac.uk/sale/thakker-jape-tutorial/GATE JAPE manual.pdf](http://gate.ac.uk/sale/thakker-jape-tutorial/GATE%20JAPE%20manual.pdf).
 19. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web.* pringer-Verlag, Secaucus, NJ, USA (2007).
 20. Hoekstra, R., Breuker, J., Di Bello, M., Boer, A.: The LKIF Core Ontology of Basic Legal Concepts. In: Casanovas, P., Biasiotti, M.A., Francesconi, E., and Sagri, M.T. (eds.) *Proceedings of the 2nd Workshop on Legal Ontologies and Artificial Intelligence Techniques (LOAIT'07).* pp. 43–63. CEUR-WS.org, Stanford, California, USA (2007).
 21. Wyner, A., Hoekstra, R.: A legal case OWL ontology with an instantiation of Popov v. Hayashi. *Artificial Intelligence and Law.* pp. 83–107 (2012).
 22. Sapkota, K.: *Semantic Frameworks for Regulatory Compliance Support* □: PhD Thesis, (2013).
 23. Sesen, M.B., Suresh, P., Banares-Alcantara, R., Venkatasubramanian, V.: An Ontological Framework for Automated Regulatory Compliance in Pharmaceutical Manufacturing. *Comput. Chem. Eng.* 34, 1155–1169 (2010).
 24. Ge, J., Qiu, Y.: Concept Similarity Matching Based on Semantic Distance. *Proceedings of the Fourth International Conference on Semantics, Knowledge and Grid (SKG'08).* pp. 380–383. IEEE Computer Society, Beijing, China (2008).
 25. Lin, D.: An Information-Theoretic Definition of Similarity. In: Shavlik, J.W. (ed.) *Proceedings of the Fifteenth International Conference on Machine Learning (ICML '98).* pp. 296–304. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1998).
 26. Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet:: Similarity: Measuring the Relatedness of Concepts. *Proceeding of the Demonstration Papers at HLT-NAACL 2004 (HLT-NAACL--Demonstrations '04).* pp. 38–41. ACL Press, Stroudsburg, PA, USA (2004).
 27. Grobe, M.: RDF, Jena, SparQL and the “Semantic Web.” *Proceedings of the ACM SIGUCCS Fall Conference on User Services Conference (SIGUCCS '09).* p. 131. ACM Press, St. Louis, Missouri, USA (2009).
 28. Eudralex: The Rules Governing Medicinal Products in the European Union, http://ec.europa.eu/health/documents/eudralex/cd/index_en.htm.
 29. Kiyavitskaya, N., Zeni, N., Breaux, T.D.: Automating the Extraction of Rights and Obligations for Regulatory Compliance. In: Li, Q., Spaccapietra, S., Yu, E., and Olivé, A. (eds.) *Lecture Notes in Computer Science.* pp. 154–168. Springer-Verlag Berlin, Barcelona, Spain (2008).
 30. Gao, X., Singh, M.P., Mehra, P.: Mining Business Contract for Service Exceptions. *IEEE Trans. Serv. Comput.* 5, 333–344 (2011).
 31. Mu, Y., Wang, Y., Guo, J.: Extracting Software Functional Requirements from Free Text Documents. *Proceedings of International Conference on Information and Multimedia Technology, 2009. ICIMT '09.* pp. 194–198. IEEE Computer Society Press, Jeju Island, Republic of Korea (2009).

32. Dijkman, R., Dumas, M., Garcia-Banuelos, L., Kaarik, R.: Aligning Business Process Models. 2009 IEEE International Enterprise Distributed Object Computing Conference. pp. 45–53. IEEE (2009).
33. Castellanos, C., Correal, D., Murcia, F.: An Ontology-Matching based Proposal to Detect Potential Redundancies on Enterprise Architectures. 30th International Conference of the Chilean Computer Science Society (SCCC). pp. 118–126. IEEE Computer Society (2011).
34. Hashmi, M., Governatori, G., Wynn, M.T.: Business Process Data Compliance. In: Bikakis, A. and Giurca, A. (eds.) Rules on the Web: Research and Applications SE - 4. pp. 32–46. Springer Berlin Heidelberg (2012).
35. Sadiq, S., Governatori, G., Namiri, K.: Modeling control objectives for business process compliance. In: Alonso, G., Dadam, P., and Rosemann, M. (eds.) Proceedings of 5th International Conference, BPM 2007. pp. 149–164. Springer-Verlag, Berlin (2007).
36. McCarthy, D., Gella, S., Reddy, S.: DSS: Text Similarity Using Lexical Alignments of Form, Distributional Semantics and Grammatical Relations. Proceedings of the First Joint Conference on Lexical and Computational Semantics (SEM'12). pp. 557–564. ACL Press, Montreal, Canada (2012).
37. Barzilay, R., Elhadad, N.: Sentence Alignment for Monolingual Comparable Corpora. Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP'03). pp. 25–32. ACL Press, Stroudsburg, PA, USA (2003).
38. Mohler, M.A.G., Bunescu, R., Mihalcea, R.: Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency Graph Alignments. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT'11). pp. 752–762. ACM Press, Stroudsburg, PA, USA (2011).
39. Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A.: SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), in Conjunction with the First Joint Conference on Lexical and Computational Semantics (SEM 2012). pp. 385–393. ACL Press, Montreal, Canada (2012).
40. Hawalah, A., Fasli, M.: A Graph-Based Approach to Measuring Semantic Relatedness in Ontologies. Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS'11). pp. 1–12. ACM Press, New York (2011).
41. Chen, Y., Wang, J., Cheng, Z., Jing, L., Zhou, Y.: An Algorithm to Compute Similarity between Danger Objects Based on Ontology for Danger-Aware Systems. Proceedings of the 2nd International Symposium on Aware Computing (ISAC'10). pp. 128–135. IEEE Computer Society, Tainan, Taiwan (2010).
42. Yu, Z., Zhou, X.: Combining Vector Space Model and Category Hierarchy Model for TV Content Similarity Measure. Proceedings of the Third International Conference on Multimedia and Ubiquitous Engineering (MUE'09). pp. 130–136. IEEE Computer Society, Qingdao, China (2009).