Using BabelNet for Analysing COBIT 5

Nuno Teles Silva
nuno.teles@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2016

Abstract

Throughout the years Enterprises have acknowledged the importance of investing in the adoption of IT Governance (ITG) frameworks. Good-practice frameworks, such as COBIT 5, emerged as major drivers for implementing effective ITG. COBIT 5 is an important framework that assists Enterprises achieving their objectives for both Governance and Management of Enterprise IT. As the manuals of COBIT are written in a natural language and represented in some informal diagrams, activities such as the assessment and analysis of COBIT PRM are error-prone and time-consuming tasks. Relying on a language with such flexibility, COBIT manuals create great difficulties for organizations to understand and use the framework. Due to the problems that Stakeholders face to assess, analyse and apply the concepts resulting from PRM, in this work, we exert ontology techniques to assess automatically COBIT PRM and semantic techniques to infer if specific PRM elements are used similarly in the same way or context.

Keywords: IT Governance, COBIT 5 PRM, Ontology, Word Sense Disambiguation, Similarity Measure, Lexical Knowledge Base

1. Introduction

The rapid growth of Information Technology (IT) makes IT being recognized as a strategic weapon for rising competitive advantage among Enterprises. Organizations strive to increase overall business performance by using IT in an efficient, practical and strategic way. Therefore, the strategic use of IT became a crucial factor for organizations in order to gain competitive advantage and enhance protection against a dynamic and threatening environment [9].

As studies have shown, misalignment or the lack of alignment between IT and business strategies is one of the main reasons why organizations fail to realize the full potential benefits of their IT investments [4, 17]. Assuring the alignment of IT and business strategy is not a trivial task, as well as accepting the risk of associated IT investment. A way to address the mentioned misalignment is to focus on IT Governance (ITG).

COBIT currently in its fifth edition is a good practice framework for the management and governance of IT. COBIT Enabling Process guide provides a generic process model, Process Reference Model (PRM) that defines all the processes and activities which one should find according to the idea of best practice in an IT department or organization [20, 12]. As this guide is written in a natural language and represented in several informal diagrams, activities such as the assessment of these models imply manual searching for contents in the documents. Consequently, manually inferred assessments are erroneous and inefficient.

Aside from that, as reported by [7], there is a need to investigate COBIT 5 intellectual foundations, design, applicability and internal consistency. COBIT manuals are based on a natural language i.e. a language to be used by people in general for daily communication [21]. Whereas artificial languages are characterized by self-created vocabularies and strict grammars, the syntactic and semantic flexibility of a natural language enables this type of language to be natural to human beings [21]. Relying on a language with such flexibility, COBIT manuals create great difficulties for organizations to understand and use the framework. This happens because there is a potential presence of ambiguity and abstraction between concepts.

In this work, we exert ontologies as a mean to overcome the first aforesaid problem. Ontologies enable the representation of conceptual models as well as, the use of inference mechanisms to automatically validate and analyse such representations. We will reuse an existing ontological representation of COBIT 5 and a set of SPARQL queries that enable automatic analysis of COBIT PRM. The application of the aforementioned queries will generate different representative samples of PRM elements.

We want to use semantic similarity techniques as

a way to perform an internal consistency checking and thus, ascertain if there is a presence of similar elements in PRM. A Prototype to be developed, relying on different semantic techniques, will be executed in the context of a Case Study and the results will be evaluated recurring to a clustering technique.

The remainder of the document is structured as follows. Section 2 describes the related work complementary to this work. Section 3 describes the definition of a Prototype and all the resulting modules that composes it. Section 4, demonstrates the utility of the solution through a Case Study. Section 5 interprets the results of the demonstration. Section 6 concludes the work presenting contributions and lessons learned.

2. Related Work

COBIT 5 [19] provides a comprehensive framework that helps Enterprises create optimal value from IT investments, realizing benefits while reducing risk and resource use.

2.1. COBIT 5 PRM

COBIT 5 includes a Process Reference Model (PRM) that defines and describes in detail all the processes normally found in an Enterprise. Each Process belonging to PRM has common dimensions [20]:

- **Stakeholders** have predefined interests, responsibilities, and roles.
- **Goals** describes the outcomes of processes or Enterprise IT (artifact, state or a capability improvement)
- **Metrics** measure the achievement of each goal.
- **Practices** high-level requirements influenced by policies and procedures.
- **Activities** main actions taken to operate a process.
- **Work Products** support the operation of processes.

2.2. Ontologies Technologies

In order to support the representation of ontologies, many languages were developed in the past decade. W3C has developed a set of standards and technologies, also known as Semantic Web standards.

RDF is an infrastructure for describing metadata about Web resources and information about things that can be identified on the Web [6]. The data model consists of three major components: Resources, Properties, and Statements. RDFschema is a semantic extension of RDF [42] that provides a data modeling vocabulary for describing groups of related resources and relationships. OWL [41] offers a semantic markup language for publishing and sharing ontologies. SPARQL is the W3C standard for accessing and querying RDF graphs.

2.3. Semantic Similarity

Accurate measurement of semantic similarity between words is crucial for various tasks for example, word sense disambiguation [32], clustering [39], information retrieval [40], synonym extraction [10] and ontology alignment [5].

Measuring semantic similarity between two words remains a complicated exercise. A measure of semantic similarity is a function that quantifies the resemblance between two words. Word semantic similarity [13] approaches can be categorized as corpus-based and knowledge-based similarity measures.

Two acclaimed corpus-based measures are LSA and HAL. HAL and LSA are similar in the ultimate goal i.e. both methods capture the meaning of a word using co-occurrence information [24]. Nevertheless, in LSA the co-occurrence is measured against text units of paragraphs or documents whereas in HAL, a word-by-word matrix is produced based on word co-occurrences of a corpus text, using a moving window of a predefined width (normally ten).

In past few years different knowledge-based measures have shown to be beneficial when used in specific contexts. In general, semantic similarity is influenced by a number of different information sources (depth, path length, information content and gloss).

First of all, works such as Rada et al. [34] form the basis of some edge-based methods. Rada et al. demonstrated that the minimum number of edges separating two concepts in a *is-a hierachy* is a metric for measuring the conceptual distance and therefore, for assessing similarity.

Information theory-based works, for instance, Resnik [35] proposed a theory to calculate the semantic similarity of two concepts based on the notion of information content share. In agreement with [22] the information content of a concept depends on the probability of encountering an instance of that concept in a corpus. That is, the probability is calculated as the frequency of a concept divided by the total frequency. Alternative works such as [37], propose new ways to compute IC. Sanchez et al., considers the whole set of subsumers (leafs) when computing the information content values.

Li et al. present a new measure that combines different information sources for example, the benefactions of path length, depth, and local density. Path length and depth contributions are derived from a lexical database using a nonlinear function [22]. IC

is used to represent the local density of concepts in a corpus.

Gloss-based methods appeared as promising means of measuring semantic similarity through gloss information [32]. The first method was originally proposed by Lesk [2]. Lesk method compares two concepts analysing the overlaps that might occur within two definitions. Adapted Lesk measure [2, 32], introduces a new scoring algorithm that takes into consideration the size of possible overlappings of semantically related concepts. Gloss Vector measure estimates the semantic relatedness based on occurrences of second order cooccurrence vectors [32].

2.4. Short Text Similarity

Most of the existing studies concerning Text Similarity focus on calculating the similarity between documents or long sentences [16] while there are only a few publications available to calculate the similarity between very short texts or sentences [38]. Techniques for detecting similarity between short texts have been categorized into three main groups [24]: word co-occurrence, corpus-based and features-based methods.

Word co-occurrence methods, also called as the bag of words methods are widely applied in IR systems. By using an equivalent representation for documents and queries it is possible to retrieve relevant documents based on the similarity between a query and a document.

Alternative approaches [24] use lexical dictionaries to compute the similarity between two sentences. Pairs of words between two sentences are formed and meanings are transmitted into a set of patterns. The similarity is computed using a simple pattern matching algorithm.

Li et al. [24], presented a full dynamic, automatic and adaptable method for computing sentence similarity. The proposed method forms a joint word set using all the distinct words in the pair of sentences. A semantic vector for each of the sentences is built using information from a lexical knowledge base and word significance derived from a corpus. Subsequently, two word order vectors are created for each sentence and at the end, overall sentence similarity is defined as a combination of semantic similarity and word order similarity.

Another category focuses on extracting statistical information of words in huge corpora. Two acclaimed methods are LSA and HAL. LSA and HAL can be applied at the sentence level. Sentence vectors are composed by adding together each word vector within a sentence. Similarity between two sentences is calculated using a distance metric such as cosine distance.

The main idea of the features-based method is to represent a sentence using a set of predefined features [24]. Each word belonging to a sentence is represented considering a feature set of predefined features. Features such as HUMAN, SOFTNESS, and POINTNESS could be used to conceive labeled training data from the similarity of the compared sentences. Finally, a classifier could learn a function from the training data and generate similarity values in presence of test sets.

2.5. Lexical Knowledge Base

Lexical Knowledge Bases (LKB) are digital knowledge bases that provide lexical information on words of a particular language [14]. LKB stores different types of information: senses i.e. associations between lemmas to meanings, and semantic relations. BabelNet [27] follows the traditional structure of a lexical knowledge base. Main concepts and relations present in BabelNet result from the intersection of important two sources [28]:

- **WordNet** provides a set of synsets and semantic relations. A synset is a concept identified by a set of multilingual lexicalizations.
- **Wikipedia** a multilingual Web-based encyclopedia that contributes with Wiki-pages as concepts and semantically unspecified relations.

According to [29, 30], information encoded in the text dump of BabelNet can be effectively accessed by means of a programmatic access. Important operations defined in BabelNet classes are *getSynsets*, *getSenses*, *getSynset* and *getGloss* [31].

2.6. Word Sense Disambiguation

Word Sense Disambiguation (WSD) is the capacity to computationally determine which sense of a word is chosen by its use in a particular context [26]. WSD is usually performed on one or more texts i.e collection of words $(w1, w2, ..., wn)$. WSD task is described as follows: assign the appropriate senses to all or some of the words, creating a mapping from words to senses. There are three main approaches to conduct WSD [26]: **supervised**, **unsupervised** and **knowledge-based**.

Supervised methods use machine-learning techniques and for this reason, require vast amounts of annotated data and a huge human effort for building them. Some annotated data might not be available for all the words of interest.

Unsupervised methods rely on unlabeled corpora to produce a set of clusters. As there is no share of a reference inventory of senses, the evaluation of cluster results is a arduous process [26].

Knowledge-based approaches exploits knowledge resources such as dictionaries, thesauris and ontologies. Since the resources they rely on are progres-

sively upgraded, they are important methods to consider [26].

Conforming to [32], measures of semantic relatedness can be used to perform word sense disambiguation. A target polysemous word w_t in a text T can be disambiguated by choosing the sense s_{ti} which maximizes the $relatedness(s_{ti}, s_{jk})$ to other words w_j in a given window of context.

Babelfy is an approach which aims to perform both multilingual WSD and Entity Linking. The approach exploits semantic relations between word meanings and named entities from BabelNet [28]. According to [25], there are three main steps to disambiguate sentences:

1. A set of semantic signatures (related vertices) is generated and connected by means of random walks.

2. For each of the fragments of text having at least one lexicalization (in BabelNet) their possible meanings are listed.

3. A semantic interpretation is produced by connecting the candidate meanings using the semantic signatures. All the fragments are disambiguated by means of a centrality measure.

As stated in [25], Babelfy system is accessible through a web interface and Java RESTful API. Important operations defined in Babelfy classes, are *babelfy*, *getSource*, *getScore*, *setAnnotationResource* and *addAnnotatedFragments*.

2.7. Clustering

Clustering is known as an attractive approach for finding similarities in data and putting similar data into groups [15]. The goal is that the objects within a group should be similar to one another and dissimilar from the objects in other groups.

Some clustering techniques rely on computing the number of clusters a priori. A well-accepted technique that uses this strategy is the k-centers method [11].

Affinity propagation (AP) is a graph theoretic clustering method that does not require a number of clusters to be prescribed a priori, it discovers a number of exemplars (clusters) automatically through a message-passing procedure. AP takes as input a similarity matrix s and a set of $s(k, k)$ [11]. These values are commonly referred to as "preferences".

Considering each data point as a node in a network, the message-passing method iteratively transmits real-valued messages until one of the following conditions is met [11]: a fixed number of iterations or changes in the messages fall below a threshold or alternatively after decisions stay constant. There are two kinds of message exchanged between data points [11]:

- "Responsibility" $r(i, k)$, reflects the accumulated evidence for how well-suited point k is to serve as the exemplar for point i, taking into account other potential exemplars for point i.

- "Availability" $a(i, k)$, reflects the accumulated evidence for how appropriate it would be for point i to choose point k as its exemplar, taking into account the support from other points.

3. Proposal

The research problem identifies potential problems understanding and using COBIT manuals. In this section, we propose the definition of a Prototype adequate to extract similarity values between short-text elements belonging to PRM.

3.1. Prototype Requirements

From the analysis of the research problem and solution objectives, several functional requirements regarding the Prototype were captured:

RQ1: It has to generate an overall score of semantic similarity, given work products and practice names.

RQ2: It has to disambiguate polysemous words within the sentences.

RQ3: It has to compute sentence similarity by using a short-text similarity measure.

RQ4: Similar work products and practices names must be identified recurring to a clustering technique.

3.2. Prototype Architecture

Conforming to [7], there is a need to investigate COBIT 5 intellectual foundations, design, applicability and internal consistency.

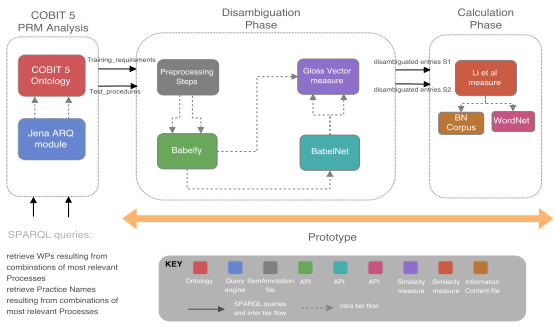

Figure 1: Prototype architecture.

Mostly of the short text similarity methods show some limitations when used in certain contexts. Methods such as the bag of words, LSA and HAL for example, use vectors of limited size to compute similarity. As a sentence is usually represented into

a very high dimensional space, extremely sparse vectors could arise and lead to less accurate calculation results. Additionally, important syntactic information is ignored.

Li et al. proposed a method that overcomes the previous limitations; it adjusts dynamically the semantic information according to the size of the input sentences and additionally, considers syntactic information. The authors in [23], mentioned that the proposed method does not currently conduct word sense disambiguation for polysemous words. Previous works tend to calculate the similarity between two sentences based on the comparison of their nearest meanings and this method is no exception. Nonetheless, it is important to disambiguate sentences and integrate with WSD since the nearest meanings do not represent their actual meanings [18]. We had this into consideration, so the developed Prototype integrates with WSD.

The high-level architecture of the Prototype is presented in Fig.1 and comprises two different phases:

- **Disambiguation Phase** - includes all the activities that matter to the preprocessing and disambiguation of the sentences.
- **Calculation Phase** - includes all the activities that matter to the calculation of an overall score of similarity.

3.2.1 Disambiguation Phase

The disambiguation phase shall start before the processing of the sentences being carried out. First of all, a co-occurrence matrix was created the same way as exposed for Gloss Vector measure. Such matrix exposes the number of times in which a word co-occurs with all distinct words of text corpora. The reference corpora (all gloss definitions of WordNet) was preprocessed recurring to map and reduce techniques.

The process begins effectively at step 1, where the content of work products and practice names is a target to preprocessing. In this case, some preprocessing steps were applied to transform the text into a structured format: specific characters were removed and acronyms words were replaced by their extensible forms.

At step 2, the operation *"babelfy"* of Babelfy API is executed. Since there is intention of restricting the disambiguated entries only to WordNet sources, a constraint was created for that purpose. Similarly, pre-annotated fragments were devised to help the disambiguation process. Special cases, for example, expressions such as *"return on investment"* or *"human resources"* will be disambiguated into different synsets, one for each word if no pre-annotated fragment is used. However, BabelNet, specially WordNet specifies a unique synset to express that sequence of words. To prevent the occurrence of such problems, semantic annotations were created to restrict the Babelfy instance and exclude the remaining synsets.

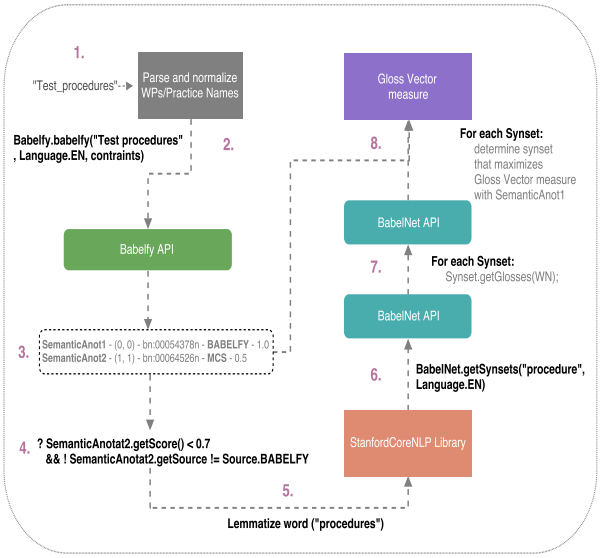

Figure 2: Disambiguation Phase.

Babelfy outputs a set of disambiguated instances at step 3. Each produced semantic annotation is associated with a confidence score (Babelfy score). When score goes below 0.7, a back-off strategy based on the most common sense is used [3]. An alternative strategy was adopted to refine all the most common senses. The fragment corresponding to the semantic annotation is sent to a lemmatizer that computes the lemmatization process. At step 6, all the BabelSynsets that corresponds to that lemma are retrieved through BabelNet API. For each BabelSynset, it gloss, is augmented with definitions providing from direct semantic relationships such as hypernyms, meronyms and gloss disambiguated.

Finally, the previously disambiguated instances (with satisfactory scores) are reused to discover which BabelSynset maximizes the score of the Gloss vector measure.

3.2.2 Calculation Phase

Calculation Phase begins when all the activities that matter to the disambiguation of the sentences are completed with success.

At step 1, each fragment of text within a sentence has a corresponding semantic annotation i.e. a BabelSynset representing its actual meaning. First of all, during the step 1, a joint set J is constructed

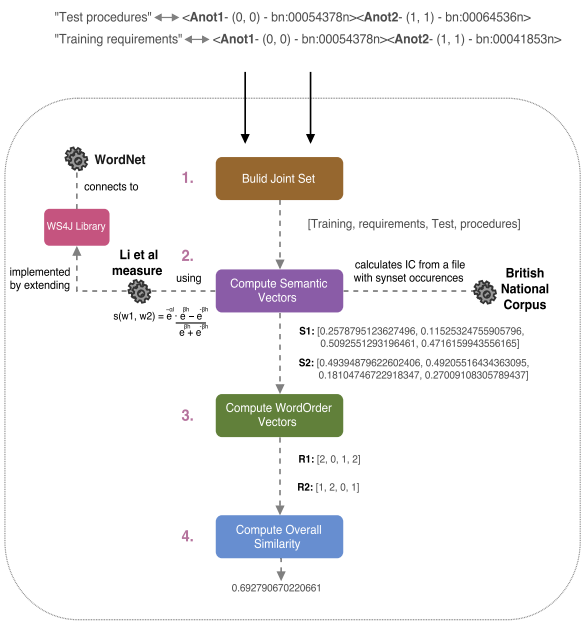

Figure 3: Calculation Phase.

dynamically by representing the semantic information for the compared sentences. Each unique word or fragment of text (of the input sentences) appears in the joint set. At step 2, two raw semantic vectors T, $\check{s_1}$ and $\check{s_2}$ are generated with the same size of the joint set. Each entry in the semantic raw vector is calculated by following the conditions:

- If w_i word of J appears in the Sentence T_i, $\check{s_i}$ is set to 1.
- If w_i is not contained in T_i, a semantic similarity score is computed between w_i and each word in the sentence T_i using the Li et al. measure. The most similar word w_i in T_i (with the maximum score) is selected. If the maximum score exceeds a preset threashold, $\check{s_i}$ sets to maximum, otherwise $\check{s_i}$ sets to zero.

Li et al. measure was implemented by extending the WS4J library. As the WS4J library already provides operations that allows users to compute the shortest path and at the same time find the depth to the lcs, the implementation is much simplified. Alongside this, the significance of each entry in the semantic raw vector is weighted by the information content derived from BNC corpus. IC is calculated through a file with SynsetIds occurrences.

During the step 3, word order vectors are created taking into account joint set word indexes. Finally, at step 4 overall similarity is calculated.

4. Demonstration

This section proves that the aforestated Prototype is adequate to process semantic information and therefore, to generate similarity values between different elements belonging to PRM.

4.1. Representative sample calculation

Before exerting the Prototype in practice, an existing ontological representation of COBIT 5, available at http://goo.gl/xc2WiO, conjointly with a set of SPARQL queries was used to generate distinct samples. As depicted in Fig.4, different combinations of relevant processes, i.e individuals having higher number of work products, were applied to extract different values of work products. By using this approach, we have a better probability of selecting the PRM elements having more influence in the execution flow of processes.

```
PREFIX process: <http://cobit/processes#>
SELECT DISTINCT ?workProduct
WHERE {
        ?startProcess a process:Process .
        ?destProcess a process:Process .
        ?startProcess process:composedOf ?practiceOutput .
        ?destProcess process:composedOf ?practiceInput .
        ?practiceOutput process:outputs ?workProduct .
        ?workProduct process:isInput ?practiceInput .
        FILTER ( ?startProcess = ?consideredOrigProcess
                 && ?destProcess = ?consideredDestProcess)
        VALUES (?consideredOrigProcess ?consideredDestProcess) {
          ( process:BAI01 process:BAI03 )
          ( process:BAI01 process:APO01)
          ( process:BAI03 process:BAI01 )
          ( process:BAI03 process:APO01)
          ( process:APO01 process:BAI01)
          ( process:APO01 process:BAI03)
        }
GROUP BY ?workProduct ORDER BY ASC(?workProduct)
```

Figure 4: SPARQL query that combinates processes to generate work products.

The size of the samples was calculated by virtue of a simple statistical formula. Yamane [1], provides a simplified formula. With a confidence level of 95%, precision e of 0.05 and N, a population size of 478, the application of the formula $n = \frac{N}{1+N \cdot e^2}$ estimates that our sample needs at least 217 elements for work products. To satisfy such constraint the query was limited to the most 28 relevant processes. For simplification reasons, Fig.4, just shows the three most relevant processes.

Two PRM samples were generated and part of their content is presented in Figs.5 and 6.

workProduct
process:Acceptance_criteria
process:Access_logs
process:Action_plan_to_adjust_licence_numbers_and_allocations
process:Aggregated_risk_profile,_including_status_of_risk_management_actions
process:Aligned_HR_performance_objectives
process:Allocated_access_rights
process:Allocated_levels_of_authority
process:Allocated_roles_and_responsibilities
process:Analysis_of_rejected_initiatives
process:Approved_acceptance_and_release_for_production
process:Approved_acceptance_test_plan
process:Approved_acquisition_plan
process:Approved_changes_to_baseline
process:Approved_detailed_design_specification
process:Approved_quality_reviews
process:Approved_strategic_options
process:Architecture_governance_requirements

Figure 5: An excerpt of the generated sample for Work Products.

practiceName1	practiceName2
Use and share knowledge.	Analyse and report performance.
Use and share knowledge.	Analyse risk.
Use and share knowledge.	Articulate risk.
Use and share knowledge.	Assess business impact.
Use and share knowledge.	Build solutions.
Use and share knowledge.	Close a programme.
Use and share knowledge.	Close a project or iteration.
Use and share knowledge.	Close service requests and incidents.
Use and share knowledge.	Collect data.

Figure 6: An excerpt of the generated sample for Practice Names.

The sample for work products has 237 distinct elements while for practice names have 184 distinct elements. Both samples are a good cover of COBIT total work products (almost 50% i.e. 237/478) and practice names (88% i.e. 184/210).

4.2. Similarity calculation

In this section, we apply the Prototype defined in Sections 3.1 and 3.2 to both of PRM samples. In the first place, the Prototype was executed with the sample corresponding to work products. The result (see Fig.7) was a matrix with 237 rows and 237 columns. For practice names (see Fig.8), the SPARQL query produces 27966 results.

workProduct1	workProduct2	Similarity
Compliance audit results	Known error records	0.0500
Compliance audit results	Evaluation of innovation benefits	0.0500
Compliance audit results	Test result logs and audit trails	0.6668
Compliance audit results	Programme audit plans	0.4554
Compliance audit results	Incident resolutions	0.4038
Compliance audit results	Test plan	0.4040
Compliance audit results	Vision communications	0.3922
Compliance audit results	Results of internal control monitoring and reviews	0.6823
Compliance audit results	Risk assessment initiatives	0.3241
Compliance audit results	Incident response actions and communications	0.5978

Figure 7: An excerpt of Work Products similarity results.

practiceName1	practiceName2	Similarity
Analyse risk.	Define the organisational structure.	0.1933
Analyse risk.	Understand business expectations.	0.1877
Analyse risk.	Manage supplier risk.	0.5039
Analyse risk.	Manage programme and project risk.	0.4044
Analyse risk.	Manage requirements risk.	0.5274
Analyse risk.	Design detailed solution components.	0.2250
Analyse risk.	Optimise asset costs.	0.2610
Analyse risk.	Secure information assets.	0.3252
Analyse risk.	Manage backup arrangements.	0.2598
Analyse risk.	Manage critical assets.	0.2610

Figure 8: An excerpt of Practice Names similarity results.

5. Evaluation

This section shows the evaluation of the proposed solution. Some clustering techniques are explored as a basis to group sentences of PRM.

5.1. Affinity Propagation application

Cluster analysis of texts is a well-established problem in IR. Generally, documents are typically represented as data points (vectors) in a high dimensional vector space. Since data points using this representation lie in a metric space [33], there are plenty algorithms indicated to cluster such points.

In our case, we have different pairwise similarities that resulted from the demonstration step. Algorithms, such as AP, are viable options to apply in this case. AP algorithm was initialized with the following information:

- Number of iterations was set to 100.
- Preferences entries were defined using the minimum similarity value.
- Dumping factor was set to 0.5.
- Thresholds of 0.15 and 0.10 were used.

The AP algorithm is executed several times until the objective function is maximized. According to [8], the objective function maximizes the net similarity, S, which is the sum of the similarities of non-exemplar data points to their exemplars plus the sum of exemplar preferences.

5.2. Results of the Cluster Analysis

Before applying the AP algorithm, 90 of 237 work products were identified and removed from the initial set. Work products in such situation had average values of similarity (to all data points) lower than 0.15. The same applies for practice names, where 40 in 184 were removed using a lower threshold, 0.10.

The execution of AP produced 33 clusters for work products and 50 for practice names. After generating the results, silhouette method was employed as cluster validity mechanism. The formula described in [36] was applied at cluster level. Positive values obtained for work products and practice names, respectively, 0.12 and 0.06, demonstrate that the objects lie well within clusters with satisfactory results.

5.3. Work Products Results Interpretation

Pairwise similarity values were assembled to compute intra-cluster similarity value. Major conclusions that result from the analysis of the cluster content and Figs.9 and 10 are:

- Examples of clusters relating work products with higher intra-cluster values (higher than 0.6) are *wp-cluster-#14*, *wp-cluster-#18* and *wp-cluster-#1*.
- Examples of clusters relating work products with lower intra-cluster values (lower than 0.5)

are *wp-cluster-#25*, *wp-cluster-#3* and *wp-cluster-#15*.

- Most of the time cluster sizes and intra-cluster similarity values occurs inversely. For example, clusters such as, *wp-cluster-#14* and *wp-cluster-#18* has lower sizes whereas higher intra-cluster similarity values.

- A total of 79 out of 147 work products (53% of our sample) reside in clusters with intra-cluster similarity above 0.5. If we decrease the limit to 0.4 we note that 143 out of 147 work products (93% of our sample) exceed that limit. To support this, there is great affinity between WP *"Remedial actions and assignments"* and *"Noncompliance remedial actions"*.

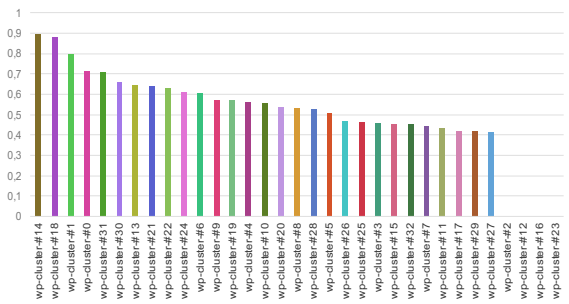

Figure 9: Intra-cluster similarity for Work Products.

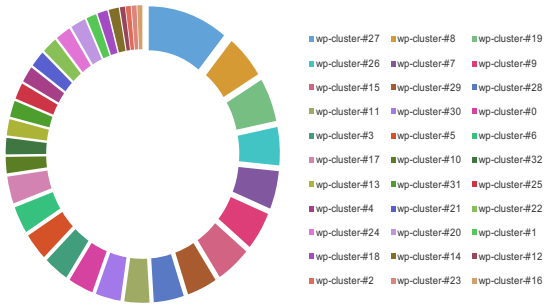

Figure 10: Cluster sizes of Work Products.

5.4. Practice Names Results Interpretation
Major conclusions that result from the analysis of the cluster content and Figs.11 and 12 are:

- Examples of clusters relating practice names with higher intra-cluster values (higher than 0.6) are *wp-cluster-#40*, *wp-cluster-#15* and *wp-cluster-#39*.

- A total of 9 out of 12 clusters with higher intra-cluster similarity contains practice names pertaining to different domains. For example the pair, {*"Perform a post-implementation review."*,*"Conduct post-resumption review."*} belongs to domains BAI and DSS.

- Examples of clusters relating practice names with lower intra-cluster values (lower than 0.5) are *wp-cluster-#44*, *wp-cluster-#21* and *wp-cluster-#6*.

- A total of 41 out of 144 of practice names (only 0.28% of our sample) reside in clusters with intra-cluster similarity above 0.5. Several clusters appear with an intra-cluster value of 0 i.e size of 1. The existence of those clusters mean, that after execution of AP algorithm, no other element shares the same exemplar i.e. practice name.

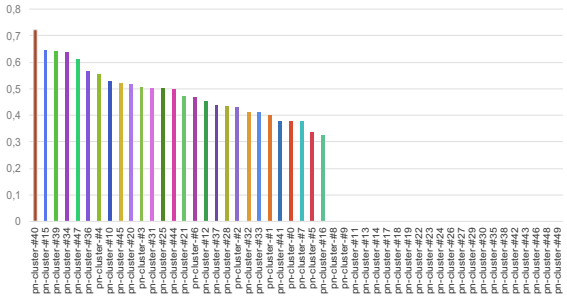

Figure 11: Intra-cluster similarity for Practice Names.

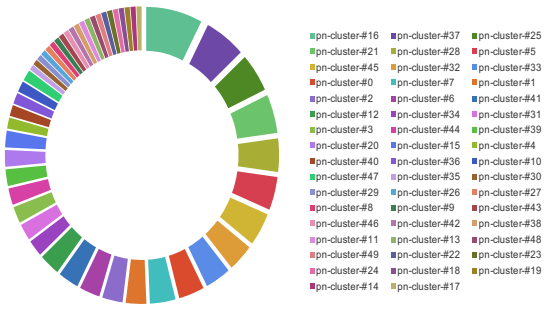

Figure 12: Cluster sizes of Practice Names.

6. Conclusion

COBIT 5 provides a comprehensive framework that assists Enterprises achieving their objectives for the Governance and Management of Enterprise IT. Some critics still argue that there is still a lack of theoretical foundation from an academic viewpoint that explores COBIT as a research artifact.

As this official guide is written in a natural language and represented in some informal diagrams, activities such as the assessment of these models imply manual searching for contents in the documents and for this reason, turn out to be massively resource consuming.

On the other hand, COBIT manuals create great difficulties for organizations to understand and use the framework. From the previous problems, we concluded that a scientific analyse and research on top of these models is missing.

Ontology techniques were used to narrow this gap i.e. to understand and assess the logical structures and generating semantics of PRM.

Semantic techniques were used to ensure an internal consistency checking and thus, ascertain if there is a presence of similar elements in PRM.

6.1. Lessons learned

- DSRM provided us guidance on how to conduct the evaluation of IT artifacts and principally how to structure a successful research work.

- Assessments made on top of PRM could benefit from having a representation with Ontologies. Previously manually assessments of COBIT PRM would cost a huge amount of time and thereby are susceptible to errors. Using ontologies techniques, supplementary assessments made on top of PRM achieve satisfactory results and take just a couple of minutes to execute.

- Measuring semantic similarity between two sentences could be a challenging exercise. If COBIT manuals were written in a normalized vocabulary this task would be simplified. Therefore, we had to implement some preprocessing steps in order to normalize the content of some sentences as well as to disambiguate polysemous words.

- Similarity techniques are crucial to conduct some natural language processing tasks. Different algorithms have pros and cons that might be evaluated. Specific algorithms use a precompiled word list to represent sentences into a very high dimensional space, others require manual compilation and ignore syntactic information within the texts.

- AP method is useful to analyse and find similarities in data. We concluded that a total of 79 out of 147 work products (53% of our sample) reside in clusters with intra-cluster similarity above 0.5. Besides, only a total of 41 out of 144 of practice names (0.28% of our sample) reside in clusters with intra-cluster similarity above 0.5. Such observations allow us to conclude that WPs have the tendency to be semantically similar to each other while the same does not occur in practice names.

References

[1] S. Ajay and B. Micah. Sampling techiniques & determination of sample size in applied statistics research: an overview. *IJECM*, 2014.

[2] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. *IJCAI*, 2003.

[3] J. Camacho-Collados, C. D. Bovi, A. Raganato, and R. Navigli. A Large-Scale Multilingual Disambiguation of Glosses. In *Proceedings of the Tenth International LREC*, 2016.

[4] Y. E. Chan. Why Haven't we Mastered Alignment? The Importance of the Informal Organization Structure. *MIS Quarterly*, 2002.

[5] V. Cross and X. Hu. Using semantic similarity in ontology alignment. In P. Shvaiko, J. Euzenat, T. Heath, C. Quix, M. Mao, and I. F. Cruz, editors, *OM*, CEUR Workshop Proceedings. CEUR-WS.org, 2011.

[6] J. Davies, D. Fensel, and F. Van Harmelen. *Towards the Semantic Web Towards: Ontology-driven Knowledge Management*. Wiley, 2003.

[7] S. De Haes, W. Van Grembergen, and R. S. Debreceny. COBIT 5 and Enterprise Governance of Information Technology: Building Blocks and Research Opportunities. *JIS*, 2013.

[8] D. Dueck. *Affinity Propagation: Clustering data by passing messages*. PhD thesis, 2009.

[9] M. Faryabi, A. Fazlzadeh, B. Zahedi, and H. A. Darabi. Alignment of Business and IT and Its Association with Business Performance: The Case of Iranian Firms. *IJBM*, 2012.

[10] O. Ferret. Testing semantic similarity measures for extracting synonyms from a corpus. *ELRA*, 2002.

[11] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 2007.

[12] S. Goeken, Matthias and Alter. Towards Conceptual Metamodeling of IT Governance Frameworks Approach Use Benefits. In *42th HICSS*, 2009.

[13] W. Gomaa and A. Fahmy. A survey of text similarity approaches. *IJCA*, 2013.

[14] I. Gurevych, J. Eckle-Kohler, and M. Matuschek. *Linked Lexical Knowledge Bases: Foundations and Applications*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2016.

[15] K. Hammouda. A Comparative Study of Data Clustering Techniques. *Tools of Intelligent Systems Design: Course Project SYDE 625*, 2000.

[16] V. Hatzlvassiloglou, J. L. Klavans, and E. Eskin. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In *1999 Joint SIGDAT EMNLP*, 1999.

[17] J. C. Henderson and N. Venkatraman. Strategic Alignment: Leveraging Information Technology for Transforming Organizations. *IBM Syst. J.*, 1999.

[18] C. Ho, M. A. A. Murad, R. A. Kadir, and S. C. Doraisamy. Word Sense Disambiguation-based Sentence Similarity. In *Proceedings of the 23rd COLING: Posters*, Stroudsburg, PA, USA, 2010. ACL.

[19] ISACA. *COBIT 5: A Business Framework for the Governance and Management of Enterprise IT*. 2012.

[20] ISACA. *COBIT 5: Enabling Processes*. 2012.

[21] M. C. Lee, J. W. Chang, and T. C. Hsieh. A Grammar-Based Semantic Similarity Algorithm for Natural Language Sentences. *TSWJ*, 2014.

[22] Y. Li, Z. A. Bandar, and D. McLean. An approach for measuring semantic similarity between words using multiple information sources. *IEEE TKDE*, 2003.

[23] Y. Li, D. Mclean, Z. Bandar, J. D. O. Shea, and K. Crockett. Sentence Similarity Based on Semantic Nets and Corpus Statistics (full paper). 2006.

[24] K. Li, Yuhua and McLean, David and Bandar, Zuhair A. and O'Shea, James D. and Crockett. Sentence Similarity Based on Semantic Nets and Corpus Statistics. *IEEE TKDE*, 2006.

[25] A. Moro, F. Cecconi, and R. Navigli. Multilingual Word Sense Disambiguation and Entity Linking for Everybody. In *Proceedings of the 13th ISWC*, 2014.

[26] R. Navigli. Word Sense Disambiguation: A Survey. *ACM Surv.*, 2009.

[27] R. Navigli and S. P. Ponzetto. *BabelNet: Building a very large multilingual semantic network*. 2010.

[28] R. Navigli and S. P. Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *AI*, 193, 2012.

[29] R. Navigli and S. P. Ponzetto. BabelNetXplorer: A Platform for Multilingual Lexical Knowledge Base Access and Exploration. *Companion Volume to the Proceedings of the 21st WWW Conference*, 2012.

[30] R. Navigli and S. P. Ponzetto. Multilingual WSD with just a few lines of code: the BabelNet API. In *Proceedings of the 50th Annual Meeting of the ACL*, 2012.

[31] R. Navigli and F. Vannella, Daniele Cecconi. BabelNet API guide.

[32] T. Pedersen, S. Banerjee, and S. Patwardhan. Maximizing semantic relatedness to perform word sense disambiguation. *UMSI report*, 2005.

[33] S. J. M. Phil, A. C. M. Sc, and M. Phil. Survey on Clustering Algorithms for Sentence Level Text. *IJCTT*, 2014.

[34] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. In *IEEE SMC*, 1989.

[35] P. Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the 14th IJCAI Vol 1*. Morgan Kaufmann Publishers Inc., 1995.

[36] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *JCAM*, 1987.

[37] D. Sánchez and M. Batet. A New Model to Compute the Information Content of Concepts from Taxonomic Knowledge. *IJSWIS*, 2012.

[38] Shashank, S. Kaur, and S. Singh. *Improving Accuracy of Answer Checking in Online Question Answer Portal Using Semantic Similarity Measure*. Springer Singapore, 2016.

[39] G. J. Torres, R. B. Basnet, A. H. Sung, S. Mukkamala, and B. M. Ribeiro. A Similarity Measure for Clustering and Its Applications. *IJECES*, 2008.

[40] G. Varelas, E. Voutsakis, P. Raftopoulou, E. G. M. Petrakis, and E. E. Milios. Semantic similarity methods in WordNet and their application to information retrieval on the web. *Proceedings of the seventh ACM international workshop on WIDM 05*, 2005.

[41] W3C. OWL Web Ontology Language Reference, W3C Recommendation, 2014.

[42] W3C. RDF Schema 1.1, W3C Recommendation, 2014.