Vol. 20, n.° 4, pp. 79-90, 2021

Revista UIS Ingenierías

Journal homepage: https://revistas.uis.edu.co/index.php/revistauisingenierias

Universidad Industrial de Santander

Identifying syntactic structures in corpus: An approach for finding knowledge areas in project-driven disciplines

Identificación de estructuras sintácticas en corpus: un enfoque para encontrar áreas de conocimiento en disciplinas dirigidas por proyectos

Antony de Jesús Henao-Roqueme [1a], Carlos Mario Zapata-Jaramillo [1b]

[1] Grupo de Investigación en Lenguajes Computacionales, Departamento de Ciencias de la Computación y de la Decisión, Universidad Nacional de Colombia, Colombia. Emails: [a] ajhenaor@unal.edu.co, [b] cmzapata@unal.edu.co. Orcid: [a] 0000-0003-2548-952X, [b] 0000-0002-0628-4097.

Received: 23 May 2020. Accepted: 16 November 2020. Final version: 25 June 2021.

Abstract

Bodies of knowledge and project management standards are defined as sets of proven methods and practices widely applied by practitioners for managing projects in particular disciplines. Since bodies of knowledge are discipline-dependent, when they are applied outside their discipline, they fail in accomplishing their purpose. Aiming to improve such bodies of knowledge, some proposals are made by performing comparisons among them. Particularly, some authors propose the adoption of new elements such as knowledge areas as a result of comparison processes. However, such proposals are empirically obtained and they are dependent on the author's judgment. Such proposals can be improved by formalizing the adoption of new elements when comparing bodies of knowledge. Consequently, in this paper, we propose a formalization method for adopting knowledge areas when comparing standards by identifying syntactic structures in project management corpus. By formalizing knowledge area adoption, we allow for improving bodies of knowledge in an author-independent way.

Keywords: syntactic structures; knowledge areas; project management; project-driven disciplines; project management corpus.

Resumen

Los cuerpos de conocimiento y los estándares de gestión de proyectos se definen como conjuntos de métodos y prácticas probados que los practicantes aplican ampliamente para gestionar proyectos en disciplinas particulares. Dado que los cuerpos de conocimiento dependen de la disciplina a la que pertenecen, cuando se aplican por fuera de sus disciplinas fallan en lograr sus propósitos. Para mejorar los cuerpos de conocimiento, en algunas propuestas se realizan comparaciones entre ellos. Particularmente, algunos autores proponen la adopción de nuevos elementos como áreas de conocimiento resultantes de los procesos de comparación. Sin embargo, tales propuestas de obtienen de manera empírica y dependen del juicio subjetivo de los autores. Esas propuestas se pueden mejorar mediante la formalización de la adopción de nuevos elementos cuando se comparan los cuerpos de conocimiento. En consecuencia, en este

ISSN Printed: 1657 - 4583, ISSN Online: 2145 - 8456, **CC BY-ND 4.0**

How to cite: A. Henao-Roqueme, C. M. Zapata-Jaramillo, "Identifying syntactic structures in corpora: An approach for finding knowledge areas in project-driven disciplines," *Rev. UIS Ing.*, vol. 20, no. 4, pp. 79-90, 2021, doi: 10.18273/revuin.v20n4-2021006.

artículo se propone un método de formalización para adoptar áreas de conocimiento al comparar estándares, mediante la identificación de estructuras sintácticas en corpus de gestión de proyectos. Al formalizar la adopción de áreas de conocimiento, se permite la mejora de los cuerpos de conocimiento de manera independiente al autor que los promueve.

Palabras clave: estructuras sintácticas; áreas de conocimiento; gestión de proyectos; disciplinas orientadas por proyectos; corpus de gestión de proyectos.

1. Introduction

Bodies of knowledge and project management standards are defined as sets of proven methods and practices widely applied by practitioners for managing projects in particular disciplines [1]. Aiming to improve project management in such disciplines, organizations define knowledge, rules, tools, and techniques allowing for controlling the environment where projects run. |Organizations like PMI (Project Management Institute), SEMAT (Software engineering Method and Theory), and DAMA (Data Management Association) provide bodies of knowledge and project management standards to codify and define the main terms related to specific disciplines.

PMI includes "a guide to the project management body of knowledge (PMBOK)," a document composed of process groups and knowledge areas. Process groups include processes performed across the project life cycle, and knowledge areas are categorical ways of grouping processes. Also, they are considered specialized dimensions practitioners should manage adequately, so project success likelihood is incremented [1].

Similarly, SEMAT is a project management organization related to software engineering with a "kernel and language for software engineering methods (Essence)," a project management standard [2]. Essence kernel has universal elements covering all software engineering projects and a formal language [2]. Essence kernel comprises alphas, activity spaces, and competencies; alphas are universal dimensions present in all software engineering projects and they represent "the things we always work with;" activity spaces represent "the things to do" in a software engineering project and "they provide descriptions of the challenges a team faces when developing, maintaining, and supporting software systems;" and competencies, represent the abilities, capabilities, and skills required for performing the work of a software engineering project [2].

Finally, DAMA International is a data management organization with the "DAMA guide to the data management body of knowledge (DAMABOK);" this document comprises data management functions and environmental elements; data management functions include activities performed in the data management discipline and environmental elements are categorical ways of grouping data management functions [3].

By comparing such bodies of knowledge and standards we can find some similarities among them—e.g., PMBOK knowledge areas have their counterpart in Essence as alphas and DAMABOK as environmental elements. So, despite such bodies of knowledge and standards use different approaches for managing projects they are consistent among them.

Bodies of knowledge and standards as PMBOK, Essence, and DAMABOK are considered discipline-dependent since they are empirically built, *i.e.*, case studies are driven to compile results and build generalizations allowing for understanding project management. So, such bodies of knowledge depend on the discipline and conditions where case studies are driven [4]. Accordingly, gathering knowledge of a large number of disciplines for compiling a multi-disciplinary project management body of knowledge is considered a complex task [5].

Aiming to improve bodies of knowledge and project management standards, improvement proposals are made by performing comparisons among them. Most improvement proposals are based on the adoption of new elements such as knowledge areas as a result of comparison processes. Gosh, *et al.* [6] compare P2M BOK, ICB, PRINCE2, APM BOK, and SBOK with PMBOK by using a process where high-level synergies, high-level differences, and high-level gaps among bodies of knowledge are established to enhance proposals of bodies of knowledge. Zapata and Henao [7] analyze dimensions of software engineering projects by comparing them with PMBOK dimensions—*i.e.*, knowledge areas—, identify missing dimensions, and propose a new dimension for including in Essence. Simonette *et al.* [8] compare the activity spaces of Essence with PMBOK knowledge areas.

Thesing *et al.* [9] use keywords of project management to determine the most useful paradigm for software project management. Matos and Lopes [10] compare processes and variables between PMBOK and PRINCE2. Takagi and Varajão [11] compare PMBOK, PRINCE2, and PM^2 for incorporating success management as a variable of such guides. Raz and Hillson [12] compare some standards for managing risks, including PMBOK. Finally, Masso *et al.* [13] perform a systematic literature review about software risk management for comparing the usage of the term among guides like PMBOK, PRINCE2, CMMI, and ISO 31000.

The aforementioned proposals are empirical and dependent on the author's judgment since they use informal methods for performing such comparisons. Such proposals can be improved by using a formalized adoption of new elements when comparing project management standards.

In this paper, we propose a method for collecting dimensions from the Essence standard to be mapped to DAMABOK. This method comprises four steps: corpus construction, verb stemming, syntactical structures, and noun extraction. The extracted nouns can be included as equivalent dimensions from Essence to DAMABOK.

We allow practitioners for comparing standards by formalizing the adoption of dimensions between project management standards in an author-independent way. So, we allow for connecting project-driven disciplines and easing knowledge transfer among them.

This paper is organized as follows: In Section 2 we describe the structure of PMBOK, Essence, and DAMBOK. Also, we describe the basis for constructing syntactic structures. In Section 3 we present author-dependent comparison processes among bodies of knowledge and their problems. In Section 4 we solve such problems with a method for extracting nouns from DAMABOK corpus with syntactic structures from Essence. Finally, in Section 5 we discuss conclusions and future work.

2. Theoretical framework

2.1. PMBOK

PMBOK comprises process groups and knowledge areas. Process groups include processes performed across the project life cycle, knowledge areas are categorical ways of grouping processes. Also, they are considered as specialized dimensions practitioners should manage adequately, so project success likelihood is incremented [1]. We present the general PMBOK structure in Figure 1, including the way to relate process groups and knowledge areas.

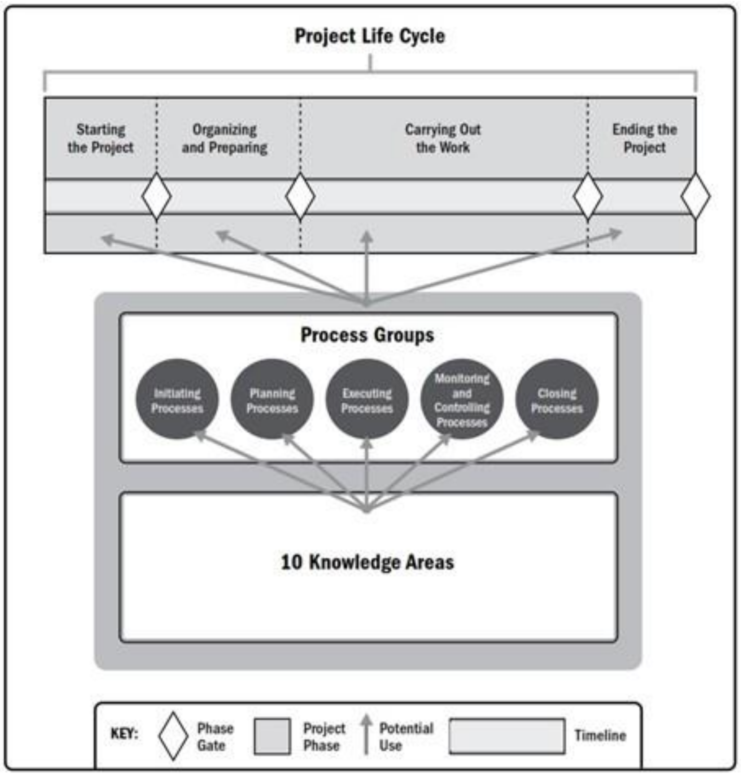

Figure 1. PMBOK structure. Source: [1].

2.2. Essence

Essence is a software engineering project management standard. Also, SEMAT is promoting a theory for software engineering as a way to improve the method and practice transference among teams. To this end, a kernel of universal elements covering all software engineering projects—i.e., elements we always find when running software engineering projects—and a formal language are created in Essence [2].

Essence kernel is categorized in the following areas of concern: client, solution, and endeavor. Some universal dimensions—they are called alphas in Essence—are present in any software engineering project and they are specialized in such areas of concern. Alphas represent "the things we always work with," and they allow for tracking the health and progress of software engineering projects via alpha states. Also, Essence comprises activity spaces and competencies. Activity spaces represent "the things to do" in a software engineering project. "They provide descriptions of the challenges a team faces when developing, maintaining, and supporting software systems;" and competencies represent the abilities, capabilities, and skills required for performing the work in a software engineering project [2]. Essence kernel alphas and their relationships are presented in Figure 2. Be advised that Essence kernel alphas can be seen as the counterpart of PMBOK knowledge areas. Also, Essence kernel activity spaces can be seen as the counterpart of PMBOK process groups.

2.3. DAMABOK

Aiming to improve data management in enterprises, DAMA is promoting a body of knowledge (DAMABOK) to provide standard definitions, guiding principles, good practices, tools, and techniques. DAMABOK is intended to address common issues in data management disciplines.

DAMABOK comprises data management functions related to activities we always perform in data management (see Figure 3) and environmental elements to serve as the main categories of the process (see Figure 4). Consequently, management functions are composed of activities performed in data management functions and environmental elements are categorical ways of grouping data management functions [3]. Be advised that data management functions can be seen as the counterpart of PMBOK knowledge processes. Also, environmental elements spaces can be seen as the counterpart of PMBOK knowledge areas and Essence kernel alphas.

2.4. Syntactic structures

Chomsky [14] defines syntactic structures as trees for generalizing language sentences. Some rules are used for syntactically structuring natural language and they are based on grammatical categories. Some of the grammatical categories are nouns (N), verbs (V), prepositions (P), adjectives (Adj), and adverbs (Adv).

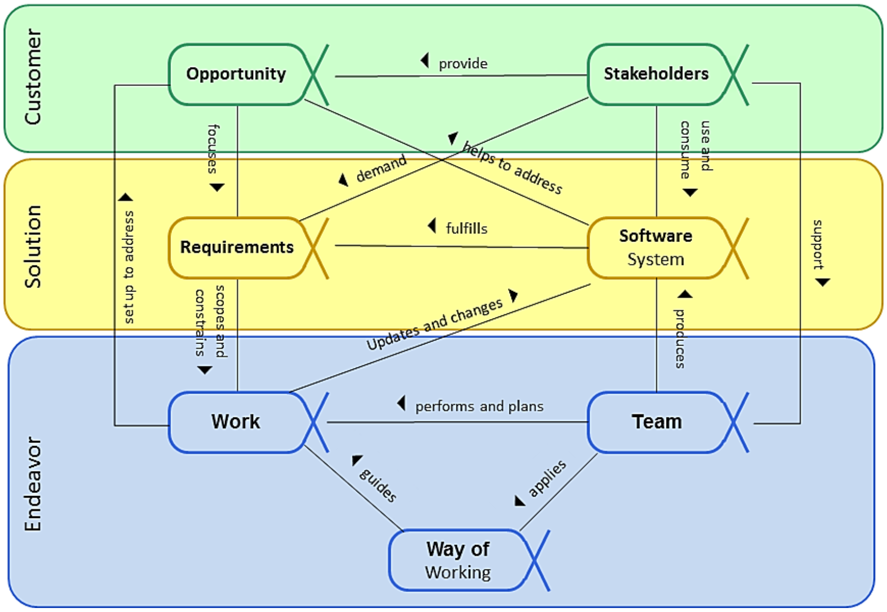

Figure 2. Alphas of the Essence kernel. Source: [2].

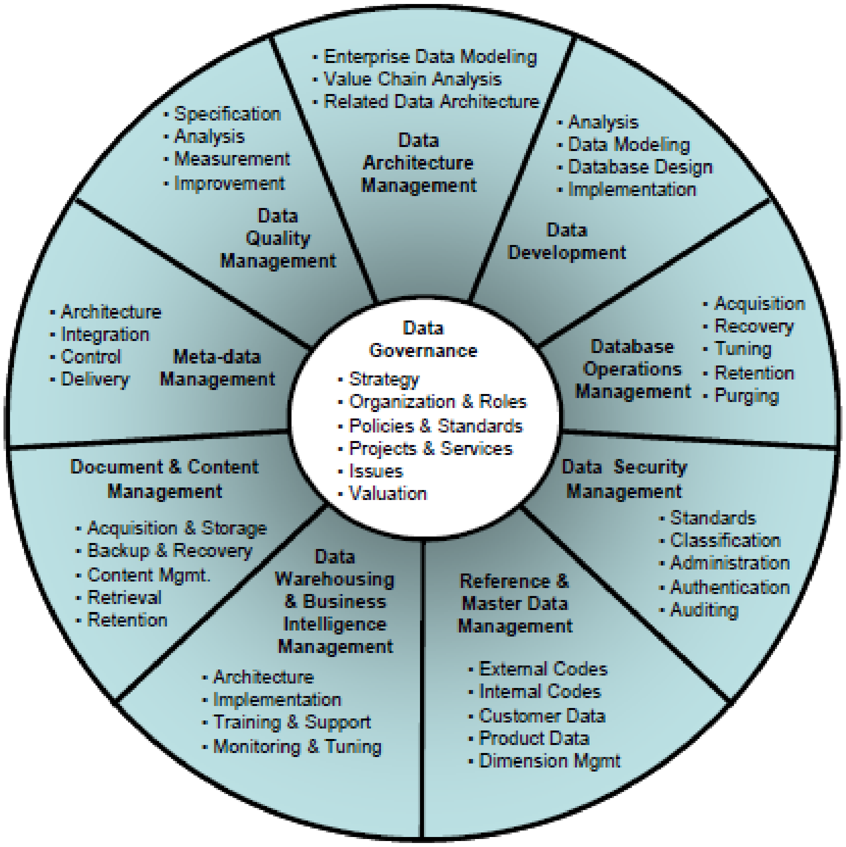

Figure 3. DAMABOK data management functions. Source: [3].

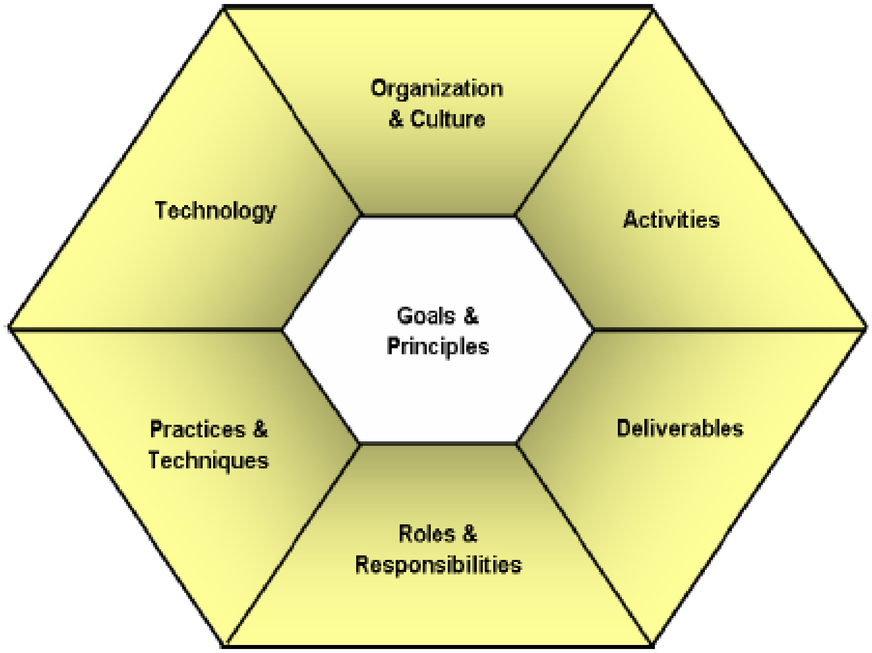

Figure 4. DAMABOK environmental elements. Source: [3].

Some syntactic structures are presented in Table 1.

Table 1. Syntactic structures

Structure	Abbre-viation	Syntactic rule
Sentence	S	(i) NP + VP
Noun phrase	NP	(i) Determinant (D) + Adj + N (ii) NP + conjunction (C) + NP
Verbal phrase	VP	(i) Aux + V + NP (ii) Aux + V + PP
Prepositional phrase	PP	(i) P + NP + VP
Adjectival phrase	ADJP	(i) Adj + PP

Source: the authors based on [14].

We present the syntactic structure of the sentence "The software system is implemented by developers" in Figure 5.

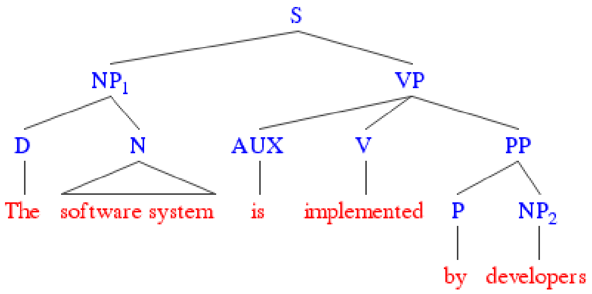

Figure 5. Example of syntactic structure. Source: the authors based on [11].

Be advised that the syntactic structure in Figure 5 also matches other sentences like the following:

- "The software system is developed by developers"
- "The software system is implemented by the team"
- "The software system is required by stakeholders"

Consequently, we can generalize several sentences by using one unique syntactic structure.

3. Background

Aiming to improve bodies of knowledge and project management standards, improvement proposals are made by performing comparisons among them.

Ghosh *et al.* [6] compare project management standards and bodies of knowledge as P2M BOK, ICB (IPMA Competence Baseline), PRINCE2, APM BOK, SBOK, and PMBOK. Ghosh *et al.* [6] base their work on finding high-level synergies, high-level differences, and high-level gaps among bodies of knowledge. As a result, new elements such as knowledge areas, processes, process outputs, tools and techniques, milestones, or competencies are proposed.

Some IPMA elements are proposed to be incorporated into PMBOK as process outputs (failure criteria and success criteria), tools and techniques (balance scored card, successive principle, interface management, individual profile assessment, group dynamics, moving through project forward and backward, systems and lateral thinking, and information and communication technology), organizational process assets (standard operating procedures), and milestones (schedule milestone and estimation milestone).

PRINCE2 elements can enhance PMBOK in the following way: principles (continued business justification, managing by milestone, managing by execution, product focus, and tailoring to the project environment), process inputs (business case), knowledge areas (planning), and processes (starting up the project, initiating the project, directing the project, controlling stage, and managing stage boundaries).

Some elements of P2M BOK can be incorporated into PMBOK as tools and techniques (balance scored card and assessment of business eligibility and economic efficiency) and processes (project organization management, project goal management, and project information technology management). APM BOK can provide PMBOK with the following elements: processes (project success and benefits management, value management, issue management, and handover and closeout) and competencies (technology management, configuration management, project reviews, communication, teamwork, leadership, conflict management, and negotiation).

Finally, some elements of SBOK can be incorporated into PMBOK as processes (define project language, perform quality control, and deviation correction) and events (agile project execution, increment planning, and alignment meeting). Even though new elements are suggested, the way to find them is very subjective and the authors just try to fill in the gaps among different guides without a holistic view.

Zapata and Henao [7] analyze PMBOK to find missing dimensions related to the Semat Essence standard. They discover some empirical equivalences based on the names (for example *requirements* alpha in the Essence kernel can be similar to the *requirements management* knowledge area in PMBOK) as a way to discover a new alpha called *risk*, corresponding to the *risk management* knowledge area. Zapata and Henao [5] also propose some states of the *risk* alpha: uncertain (when threats, impact, and needs of the risk management are unknown), identified (when risks are identified, resources are established, and risk management is planned), understood (when quantitative and qualitative analyses are done and risk committee has been achieved), planned (when risk responses, monitoring, and control plans have been developed and budget is established), and under control (when risk impact and likelihood have been reduced and resources are available for risk response). Just one additional element is proposed to be incorporated into Essence by using empirical similarity.

Simonette *et al.* [8] analyze the relationships between the Semat Essence standard and PMBOK in the context of the internet of things. To this end, they map the PMBOK processes into some activity spaces of the Semat Essence kernel, for example: collect requirements and define scope into understanding the requirements; define activities into shape the system; perform quality assurance to test the system; perform qualitative and quantitative risk analysis into understanding the requirements; and so on. Again, some sort of empirical similarity is used, with little impact in a general theory about software management.

Thesing *et al.* [9] use some elements of project management like project scope, organizational context, characteristics of the project team, time requirements, and budget requirements to determine a model for deciding between paradigms to manage software projects in specific situations. In this case, no new elements are suggested, but the authors show a way to compare to provide a holistic view about project management applied to software projects.

Matos and Lopes [10] map PMBOK process groups into PRINCE2 processes as follows: initiating into starting up and directing; planning into initiating and planning; executing and controlling into controlling a stage, managing product delivery, and directing; and closing into closing. Also, they map what they called *variables* (PMBOK knowledge areas and PRINCE2 themes and other elements) for both bodies of knowledge in the same way: integration into combined processes and components and change control; scope, time, and cost into plan and business case; quality into quality, configuration management, and control; risk into risk; communications into combined processes and components; and human resources into the organization. The authors use some sort of empirical matching for the mapping process, but no objective metric is used.

Takagi and Varajão [11] propose the usage of design science for establishing a way to compare PMBOK, PRINCE2, and PM^2 and then incorporate success management in some parts of the PM^2 process. Even though design science can be considered a systematic way to solve business problems, the authors still lack some rigor in the way to incorporate success management in the rest of the process.

Raz and Hillson [12] compare some standards and bodies of knowledge related to risk management and map processes among them related to some phases (risk identification, analysis, and treatment). They discover the name *risk identification* into most of the standards and bodies of knowledge they map, but some differences arise when mapping *risk analysis* into risk estimation, risk evaluation, risk assessment, estimating frequency, and so on. In this case, the authors discover discrepancies in the terminology when comparing the different documents, but they do not propose new elements.

Similarly, Masso *et al.* [13] study risk in software projects by performing a systematic literature review about the topic by including guides like PMBOK, PRINCE2, CMMI, and ISO 31000. Even though the authors are only collecting the information related to the topic, they use a more objective way to analyze the information by using the co-occurence of words. However, they are not suggesting new terms for risk management and they avoid the usage of syntactic structures.

Findings of the background are summarized in Table 2. From such a table, we can conclude the aforementioned proposals are empirical since the authors sometimes use the same names for creating equivalences among standards and bodies of knowledge.

However, when the name of the element is treated differently, the authors use their own judge for matching the elements. Consequently, no formal methods are used for performing such comparisons. Also, they are empirical and error-prone. In addition, the comparison is made between standards/bodies of knowledge belonging to the same discipline (*i.e.*, project management), but the equivalences should be made between two different disciplines. Consequently, a formal approach for adopting elements when comparing standards between different disciplines needs to be established.

Table 2. Summary of findings

Ref	Approach	Addition of new elements	Holistic vision	Documents
[6]	Empirical mapping	Yes	No	IPMA, PMBOK, PRINCE2, P2M, APM
[7]	Empirical mapping	Yes	No	Essence, PMBOK
[8]	Empirical mapping	No	No	Essence, PMBOK
[9]	Empirical mapping	No	No	PMBOK, PRINCE2
[10]	Empirical mapping	No	No	PMBOK, PRINCE2
[11]	Design science	Yes	No	PMBOK, PRINCE2, PM^2
[12]	Empirical mapping	No	No	PRAM. PMBOK, IRM
[13]	Co-ocurrrence of words	No	No	PMBOK, PRINCE2, CMMI, ISO 31000

Source: authors.

4. A corpus-based approach for identifying matching structures between two different disciplines

As we previously mentioned, author-dependent judgment should be removed from the comparison process among standards belonging to different disciplines. Related to standards and bodies of knowledge, Hart, and Baehr [15] suggest such documents are much more than collections of knowledge assets gathered and networked in codified form. They also advocate other knowledge assets must be discovered and integrated. Such a process can be improved by formalizing the adoption of new elements when comparing standards and bodies of knowledge from different disciplines.

In this section, we propose a method for comparing Essence and DAMABOK and formalizing the adoption of knowledge areas when comparing standards from different disciplines. Be advised that the method should be applied for all dimensions—knowledge areas—in different standards—, *e.g.*, all dimensions from a project management standard should be mapped into another as a holistic view. As a result, we can obtain knowledge areas to be adopted. We define and exemplify our method for the *software system* dimension in Essence and we obtained the equivalent knowledge area to be adopted in DAMABOK.

We use GATE [16] (General architecture for text engineering) as the platform for performing the method described in this Section. To apply the method, we build two corpus based on Essence and DAMABOK. From now on, Essence corpus should be named as initial corpus and DAMABOK corpus as target corpus. Consequently, the method is graphically summarized in Figure 6 and described step by step in sections 4.1 to 4.4.

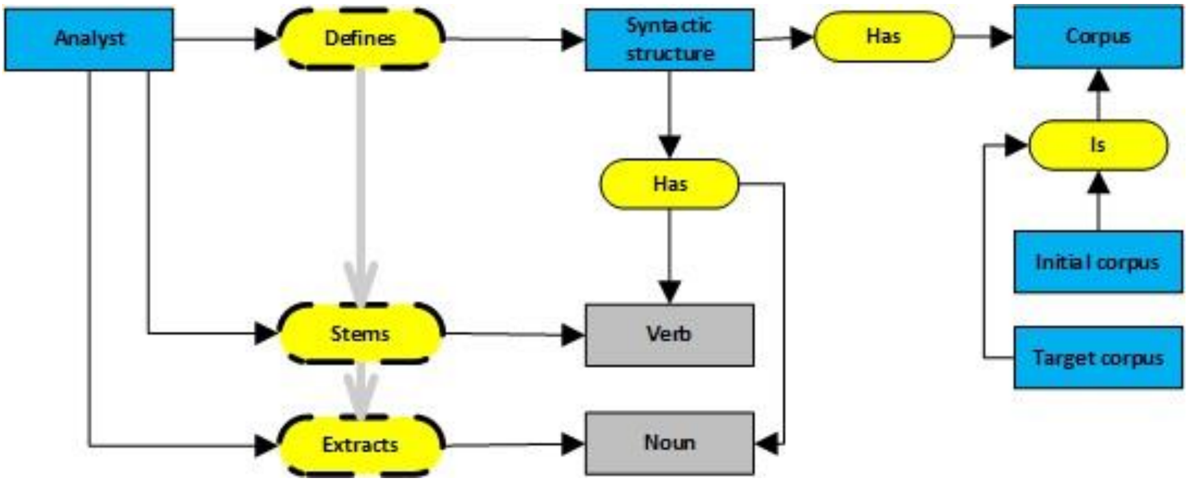

Figure 6. Graphical summary of the method. Source: authors.

4.1. Definition of syntactic structures from the initial corpus

The starting point of the method we are proposing is related to the selection of the element of the initial corpus to be compared with the target corpus. The *software system* alpha [17] is the primary outcome of a software engineering endeavor with three characteristics: functionality, quality, and extensibility. Such characteristics should be the focus of the information we can gather about the *software system* alpha.

After we gather the information from the initial corpus, we need to define syntactic structures where verbs related to the *software system* dimension appear in such a corpus. Some of the syntactic structures defined are presented in Table 3.

Table 3. Syntactic structures

Structure	Abbre-viation	Syntactic rule
Syntactic structure 1	NP + PP + the + software + system	"The specification and development of the software system"; "the actual use and exploitation of the software system; etc…
Syntactic structure 2	VP + the + software + system	"Create, update, and change the software system"; "Shape the system"; "Deploy the system"; etc…
Syntactic structure 3	The + software + system + VP	"Software system is produced"; "the system to be produced"; "the software system is identified"; etc…

Source: the authors based on [14].

Be advised that syntactic structures presented in Table 3 should not be fully matched—*e.g.*, we have sometimes the noun phrase *software system* in the corpus, but we have *the* and *software* as optional words since in the initial corpus *system* can be treated as a synonym of the *software system*. Also, we define some syntactic structures recursively—*e.g.*, in syntactic structure 2, we allow for VP to be matched several times, as happens in sentence *create, update, and change the software system*.

Such structures must be programmed in JAPE (Java Annotation Patterns Engine), a GATE component for creating patterns to be matched by the corpus when processing the structures [13]. For example, we create the following rule:

```
Rule: Rule4
Priority: 45
(
   (
      ({Token.string == "the"})*
      |
      ({Token.string == "The"})*
   )
   (
      ({Token.string == "software"})*
      |
      ({Token.string == "Software"})*
   )
   (
      {Token.string == "system"}
      |
      {Token.string == "System"}
   )
{Token.chunk=="B-VP"}
({Token.chunk=="I-VP"})*

 ):orgName
-->
:orgName.pattern3 = {kind="pattern3", rule = "Rule4"}
```

In this case, we are looking for verbal phrases (see syntactic structure 3 in Table 3).

4.2. Initial verb stemming

In this step, we execute in GATE all of the rules related to syntactic structures defined in the previous step. Some of the sentences we obtain in this process are the following:

- The actual use and exploitation of the software system.
- The specification and development of the software system.
- The requirements and funding for the software system.
- Their acceptance of the system.
- The use of the system.
- The overall design and architecting of the system.
- The use of the software system.
- The actual use and exploitation of the software system.
- The requirements for the software system.
- The success of the software system.
- The development and operation of the software system.
- Example of the nature and complexity of the system.
- The success of the software system.

- The development of the software system.
- The development and operation of the software system.
- Feedback on the system.
- The value that the software system has.
- The deployment of the software system.
- Ensure of the software system.
- The usage of the software system.
- The releases of the software system.
- The users of the system.

Then, we extract all matched verbs from the initial corpus and we obtain the stem of verbs extracted by performing a steaming process with the NLTK (Natural Language Tool Kit) from the python library. We obtain 41 verb stems as we show in Table 4 when stemming syntactic structures of the initial corpus.

Table 4. Verbs and stems from the initial corpus

Verb	Stem	Verb	Stem
Use	Use	Update	Updat
Exploit	Explot	Shape	Shape
Specify	Specifi	Fund	Fund
Develop	Develop	Integrate	Integr
Design	Design	Address	Address
Operate	Oper	Make	Make
Deploy	Deploy	Produce	Produc
Release	Releas	Recognize	Recogn
Test	Test	Identify	Identifi
Scope	Scope	Judge	Judg
Deliver	Deliv	Evolve	Evolv
Generate	Gener	Explain	Explain
Progress	Progress	Enhance	Enhanc
Organize	Organiz	Accept	Accept
Verify	Verifi	Supplement	Supplement
Support	Support	Replace	Replac
Evaluate	Evalu	Exercise	Exercis
Implement	Implement	Document	Document
Define	Defin	Build	Build
Outline	Outlin	Retire	Retir
Create	Creat		

Source: authors.

4.3. Definition of syntactic structures in the target corpus

In this step, we define a new set of syntactic structures based on stem verbs obtained in the previous step. For simplicity, we just define the “VP + the + NP” syntactic structure for the target corpus. Be advised, that VP is replaced with each verb stem presented in Table 4 and matched in the target corpus. So, we allow for finding NP related to VP—*i.e.*, knowledge area candidates—extracted in the previous step. The JAPE rule is the following (exemplified with three stems):

```
Rule: Rule1
Priority: 45
(
   (
      (
         {Token.string ==~ "[Uu]se", Token.chunk ==
"B-VP"}
         |
         {Token.string ==~ "[Uu]se", Token.chunk ==
"I-VP"}
      )
      |
      (
         {Token.string ==~ "[Ss]pecifi", Token.chunk
== "B-VP"}
         |
         {Token.string ==~ "[Ss]pecifi", Token.chunk
== "I-VP"}
      )
      |
      (
         {Token.string ==~ "[Dd]evelop", Token.chunk
== "B-VP"}
         |
         {Token.string ==~ "[Dd]evelop", Token.chunk
== "I-VP"}
      )
      |
      (
   ({Token.string ==~ "[Tt]he"})*

   (
      {Token.chunk == "B-NP"}
      |
      {Token.chunk == "I-NP"}
   )+
):orgName
-->
:orgName.stems = {rule = "Rule1"}
```

4.4. Noun extraction of the target corpus, lemmatization, and frequency analysis

In this step, we execute in GATE all of the syntactic structures defined in the previous step and extract candidate nouns (154) to be considered knowledge areas. Next, we lemmatize such nouns, and we perform frequency analysis. So, we can choose a candidate to be the counterpart knowledge area of the *software system* dimension from the initial corpus into the target corpus. In Figure 7 we present the ten most common nouns extracted from the target corpus.

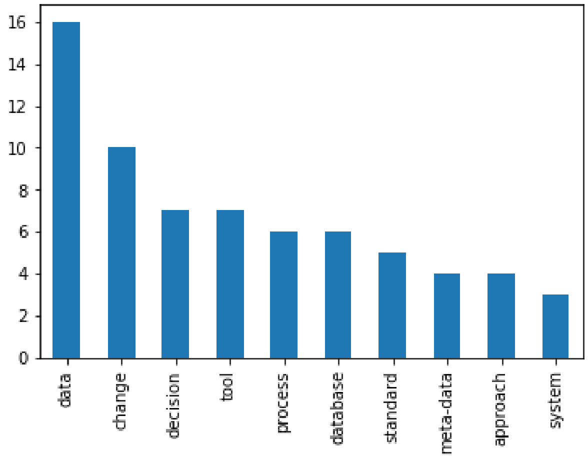

Figure 7. 10 Most common nouns extracted from the target-corpus. Source: authors.

Finally, from Figure 7 we select *data* knowledge area to be adopted in DAMABOK as a result of the comparison process with the *software system* dimension, belonging to the Semat Essence standard. In other words, we can say *data* in DAMABOK plays the same role as a *software system* in Essence based on the frequency analysis of Figure 7. You can note both standards belong to different disciplines and both words are completely different from each other—*i.e.*, their lemmas are unrelated. The method we define in this paper is based on corpus-based appearances of the words in similar scenarios since the verbs extracted from the initial corpus matched the occurences of the nouns in the target corpus. In this way, we can expect a generalization of this role in a meta-discipline. In fact, Henao [18] defines a project management multidisciplinary kernel called *quintessence* as a way to generalize all the bodies of knowledge and standards about project-driven disciplines. Henao [18] also defines an alpha called *result* as a generalization of the *software system* alpha of the Semat Essence kernel.

If we consider a *software system* as a sub-alpha of *result*, we can also say as a conclusion that *data* is also a sub-alpha of *result* in the domain of data management (see Figure 8). Several disciplines keep the same names for some dimensions, *e.g.*, *requirements* alpha of the Essence kernel and *managing requirements* knowledge area of PMBOK.

Some other names are different, *e.g.*, *opportunity* alpha of the Essence kernel and *business case* alpha of quintessence.

We can review the equivalences among those names with the method we define in this paper, as a way to deal with such differences among names.

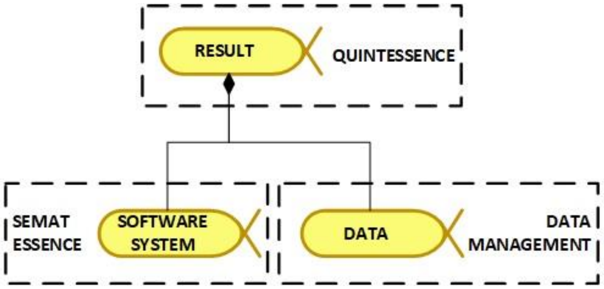

Figure 8. Similarities among alphas in three different disciplines. Source: authors.

5. Conclusions and future work

In this paper, we proposed a method for finding knowledge areas in standards/bodies of knowledge. This method is based on examining the linguistic behavior of an element related to a knowledge area, a theme, or an alpha (depending on the discipline we are working in) belonging to a discipline and projecting such behavior in another discipline (in this paper we selected a project-driven discipline vs. a data-oriented discipline). The method is based on the syntactic structures linked to the noun/noun phrase we are comparing to extract the common verbal phrases including such a noun/noun phrase. After that, we discover the stem of the verbs linked to such verbal phrases and we use them to discover nouns/noun phrases in another discipline. In this way, we allow for removing author-dependent judgment in comparisons among different disciplines. We created a prototype in GATE for demonstrating the method and we use it for discovering the equivalence of the software system alpha belonging to the Semat Essence kernel into the data management discipline: the noun data. We solved the problem of empirical judgment when comparing terms coming from different disciplines.

We have some suggestions for future work and research:

- Adding more syntactic structures to compare among other standards and bodies of knowledge. In particular, we can discover the characteristics of some common elements to all disciplines. For example, roles, competencies, phases, milestones, activities, etc.
- Validating the method with other comparisons. Equivalences to the Semat Essence standard, the quintessence, or the PMBOK can be detected in any other disciplines. For example, health management, mineralogy, economy, and so on.

References

[1] *A guide to the project Management Body of Knowledge PMBOK Guide*, 2000 ed. Newton Square, PA, USA: Project Management Institute, Inc., 2000, doi: 10.1177/875697280103200310.

[2] "Kernel and Language for Software Engineering Methods (Essence) version 1.2," OMG Doc. No. formal/18-10-02, 2018.

[3] D. F. Ayala, *The DAMA Guide to the Data Management Body of Knowledge, First ed.* Bradley Beach, NJ, USA: DAMA international, 2009.

[4] H. Smyth, P. Morris, "An epistemological evaluation of research into projects and their management: Methodological issues," *International Journal of Project Management*, vol. 25, no. 4, pp. 423-436, 2007, doi: 10.1016/j.ijproman.2007.01.006.

[5] P. Morris, L. Crawford, D. Hodgson, M. Shepherd, J. Thomas, "Exploring the role of formal bodies of knowledge in defining a profession—The case of project management," *International Journal of Project Management*, vol. 24, no. 8, pp. 710-721, 2006, doi: 10.1016/j.ijproman.2006.09.012.

[6] S. Ghosh, D. Forrest, T. DiNetta, B. Wolfe, D. Lambert, "Enhance PMBOK by Comparing it with P2M, ICB, PRINCE2, APM and Scrum Project Management Standards," *PM World Today*, vol. 14, no. 1, pp. 1-77, 2012.

[7] C. Zapata, A. Henao, "Alfa Riesgo: Un elemento universal presente en todos los esfuerzos de ingeniería de software," in *CONISOFT*, Puebla, MX, 2016.

[8] M. Simonette, M. Magalháes, E. Spina, "PMBOK and Essence: partners for IoT projects", en *Software engineering: methods, modeling, and teaching*, Bogotá: Editorial Bonaventuriana, 2017, pp. 211-223.

[9] T. Thesing, C. Feldmann, M. Burchardt, "Agile versus waterfall project management: decision model for selecting the appropriate approach to a project," *Procedia Computer Science*, vol. 181, pp. 746-756, 2021, doi: 10.1016/j.procs.2021.01.227.

[10] S. Matos, E. Lopes, "Prince2 or PMBOK—a question of choice", *Procedia Technology*, vol. 9, pp. 787-794, 2013, doi: 10.1016/j.protcy.2013.12.087.

[11] N. Takagi, J. Varajão, "Integration of success management into project management guides and methodologies—position paper," *Procedia Computer Science*, vol. 164, pp. 366-372, 2019, doi: 10.1016/j.procs.2019.12.195.

[12] T. Raz, D. Hillson, "A comparative review of risk management standards," *Risk Management*, vol. 7, no. 4, pp. 53-66, 2005, doi: 10.1057/palgrave.rm.8240227.

[13] J. Masso, F. Pino, C. Pardo, F. García, and M. Piattini, "Risk management in the software life cycle: a systematic literature review," *Computer standards & interfaces*, vol. 71, pp. 103431, 2020, doi: 10.1016/j.csi.2020.103431.

[14] N. Chomsky, *Syntactic structures*. Berlin: Mouton Publishers, 1957, doi: 10.1515/9783110218329.

[15] H. Hart, C. Baehr, "Sustainable practices for developing a body of knowledge," *Technical communication*, vol. 60, no. 4, pp. 259-266, 2013.

[16] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrel, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M. Greenwood, H. Saggion, J. Petrak, Y. Li, W. Peters, *Developing Language processing components with GATE Version 6 (a user guide)*. Sheffield, South Yorkshire, 2011.

[17] I. Jacobson, H. Lawson, P.-W. Ng, P. McMahon, M. Goedicke, *The essentials of modern software engineering: free the practices from the method prisons!*. Milton Keynes, Inglaterra: ACM Press, 2019, doi: 10.1145/3277669.

[18] A. Henao, "Towards a theory for defining a project management multidisciplinary kernel: an approach based on abstract level progress health attributes," M.Sc. Thesis, Universidad Nacional de Colombia, Medellín, 2018.