



UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

# FLORE

## Repository istituzionale dell'Università degli Studi di Firenze

### **Text mining tool for translating terms of contract into technical specifications: Development and application in the railway sector**

Questa è la Versione finale referata (Post print/Accepted manuscript) della seguente pubblicazione:

*Original Citation:*

Text mining tool for translating terms of contract into technical specifications: Development and application in the railway sector / G. Fantoni, E. Coli, F. Chiarello, R. Aprea, F. Dell'Orletta, G. Pratelli. - In: COMPUTERS IN INDUSTRY. - ISSN 0166-3615. - ELETTRONICO. - 124:(2021), pp. 103357.0-103357.0. [10.1016/j.compind.2020.103357]

*Availability:*

The webpage <https://hdl.handle.net/2158/1348781> of the repository was last updated on 2025-01-26T19:36:51Z

*Published version:*

DOI: 10.1016/j.compind.2020.103357

*Terms of use:*

Open Access

La pubblicazione è resa disponibile sotto le norme e i termini della licenza di deposito, secondo quanto stabilito dalla Policy per l'accesso aperto dell'Università degli Studi di Firenze (<https://www.sba.unifi.it/upload/policy-oa-2016-1.pdf>)

*Publisher copyright claim:*

Conformità alle politiche dell'editore / Compliance to publisher's policies

Questa versione della pubblicazione è conforme a quanto richiesto dalle politiche dell'editore in materia di copyright.

This version of the publication conforms to the publisher's copyright policies.

La data sopra indicata si riferisce all'ultimo aggiornamento della scheda del Repository FloRe - The above-mentioned date refers to the last update of the record in the Institutional Repository FloRe

(Article begins on next page)

# **Text Mining Tool for translating Terms of Contract into Technical Specifications: development and application in the railway sector**

**Fantoni G.<sup>1</sup>, Coli E.<sup>2</sup>, Chiarello F.<sup>3</sup>, Apreda R.<sup>4</sup>, Dell’Orletta F.<sup>5</sup>, Pratelli G.<sup>6</sup>**

<sup>1</sup> Department of Civil and Industrial Engineering, University of Pisa (Italy)

[g.fantoni@ing.unipi.it](mailto:g.fantoni@ing.unipi.it)

<sup>2</sup> Department of Information Engineering, University of Pisa (Italy)

[elena.coli@phd.unipi.it](mailto:elena.coli@phd.unipi.it)

<sup>3</sup> Department of Civil and Industrial Engineering, University of Pisa (Italy)

[filippo.chiarello@ing.unipi.it](mailto:filippo.chiarello@ing.unipi.it)

<sup>4</sup> Erre Quadro s.r.l. (Italy)

[apreda.riccardo@gmail.com](mailto:apreda.riccardo@gmail.com)

<sup>5</sup> Istituto di Linguistica Computazionale “Antonio Zampolli” ILC – CNR (Italy)

[felice.dellorletta@ilc.cnr.it](mailto:felice.dellorletta@ilc.cnr.it)

<sup>6</sup> formerly with AnsaldoBreda S.p.A. (now Hitachi Rail S.p.A.), Pistoia (Italy)

[guido.pratelli@gmail.com](mailto:guido.pratelli@gmail.com)

## 1. Introduction

Large scale engineering systems derive their complexity not only by the sheer number of components and subsystems and the diversity of the disciplinary bases needed in design, but by the nature of interdependencies among constituent elements. Complex engineering systems are not decomposable into quasi-independent elements, in such a way that the behaviour and performance of elements can be isolated, and their design can be organized in relatively autonomous units. Furthermore, several interdependencies cannot be identified and modelled at an early stage but emerge only during the design process, and in some unfortunate cases, even after use.

In this paper we focus on interdependencies in large scale engineering systems at an early stage of the design process, the management of technical requirements in the contract acquisition process, which includes the analysis of the tender, the quotation of the technical and economic offer and the treatment of all contractual and legal implications. Based on recent breakthrough results in automatic analysis of Functional-Behaviour-Structure in engineering design research (Fantoni et al., 2013), we develop a novel methodology for the semi-automatic treatment of large technical documentation attached to the tender and used in proposal management; we then test the methodology in a real life industrial setting and show the large increase in performance with respect to currently available solutions. The implementation of the methodology has been carried within a project for AnsaldoBreda S.p.A. (now Hitachi Rail Europe), a leading train manufacturer, by the Consortium for Quality and Innovation Engineering (Qulnn) based at the University of Pisa.

Nowadays there are technologies able to analyze huge amounts of text written in natural language, extract information from them and represent them in a structured way. In the paper the attributes and the performances of such technologies and their application to the automatic analysis of requirements are illustrated, on the basis of Computational Linguistics, that deals with the analysis and the processing of natural language using algorithms and computers.

The technical documentation attached to tenders is typically huge in size and complexity. For large engineering systems it may easily exceed 10000 technical requirements, not necessarily organized in a clear and unambiguous way, or drafted following international standards. These requirements must be extracted and classified (an activity which itself may require 20-30 man-days for large tenders), allocated to design subsystems and assigned to the responsible design units within the company. After the allocation of requirements to units, the proposal preparation takes place with preliminary technical design activity needed in order to produce cost, quality and delivery estimations. During this process, the management of interdependencies between subsystems takes place in the context of review meetings but is rarely carried out in a systematic way. This process takes place under severe time pressure. Consequently, there are several sources of risks that are difficult to recognize at an early stage. First, the allocation of requirements to design units is done by senior engineers based on professional experience and knowledge of the overall system, but due to time pressure it is not supported by a formal analysis. Software support tools currently available on the market offer only list-like check instruments. Senior engineers may not identify all the requirements within a huge body of text, may assign incorrectly the requirements to designers, may fail in capturing the interdependencies between subsystems and components impacted by the same set of requirements, may omit to transmit the relevant information on interdependencies to all project teams involved. Second, if requirements issued by customers are unclear and ambiguous or contain omissions, inconsistencies, mistakes, contradictions (among them and sometimes they are in conflict even with national and international standards), all these problems may remain unnoticed until the detailed design activity takes place. At that point, it may be difficult to obtain contract amendments from the customer. More likely, suppliers will be asked to cope with the contract by

absorbing the costs of unnoticed problems. Legal action from the supplier can destroy the trust needed in the buyer-seller relation. Even if the expert, being particularly careful, could identify errors and omissions and clarify them by negotiating with the client, he would hardly be able to identify misalignments and contradictions between tender requirements which are physically far apart within the document but related at the company system level. Third, after the signature of the contract, any modification resulting from a better appreciation of interdependencies in the tender may originate disputes and may be costly to settle. Summing up, the complexity of the engineering systems implies a number of sources of risks for the supplier.

This complexity has also a number of economic and managerial consequences. The economic analysis of procurement contracts (Laffont & Tirole, 1993; Dimitri et al., 2006) has shown that when there is uncertainty on the contract execution, the supplier is usually at disadvantage, as witnessed by the widespread adoption of contractual penalties (see also Baker et al., 1994; Fehr & Schmidt, 2007; MacLeod, 2007). From a managerial perspective, the complexity of design interdependencies makes it difficult to organize the design activity in an effective way (Baldwin & Clark, 2000; Ulrich & Eppinger, 2008) and all mistakes done in the early stages have far-reaching downstream negative implications (Smith & Eppinger, 1997; Thomke & Bell, 2001). Moreover, public procurement, characterized by huge spending flows, represents a domain where innovation could significantly impact (Nečaský et al., 2014).

Summing up according with ElMaraghy et al. (ElMaraghy et al., 2012), the sources of complexity arise from: the number of the parts, the number of their interactions, the high coupled design (Suh, 2005), the variety of the requested competences to be orchestrate (mechanical, electrical, electronic, control modules, software, human-machine interface, web structure), the complexity of the product development process, that of manufacturing and assembly processes, the management of the supply chain of both design and manufacturing enterprises and their globalisation. Last, the size of the frames of the train, the external and internal logistic issues increase the complexity at all levels. All is worsened by the uncertainty due to the first phases of the design process and time pressure.

From the above arguments it is clear why making the analysis of technical requirements more effective during the contract acquisition stage is important: the return of an early investment into new methods in terms of the downstream benefits from reduced costs and risks, but also from increased reputation and trust from customers, is significant. It is recognized that reducing the time needed for tender analysis and increasing the precision of requirements extraction from huge amounts of tender documents are key factors directly impacting on firms finance and reputation (Rajbabu et al., 2018). Usually, contract title, start date, end date etc. are extracted from tenders by applying mixed approaches: handcrafted features, post processing rules and Machine Learning techniques (Chalkidis et al., 2017). This paper proposed a further step, developing a tool able to automatically extract the requirements and assign them to the most suitable enterprise system for analyzing and processing them.

After a detailed exploration of complexity in tender analysis (section 2) in order to better define the problems we are dealing with, the paper is structured as follows: firstly, the methodology is presented (section 3); secondly, the Interface Matrix, one of the main outputs of methodology application, is presented (section 4); then, the results are presented and discussed through their validation and the presentation of the final tool (section 5). Finally, open issues and future developments are presented (section 6).

## **2. Complexity in tender analysis**

### **2.1 Problem definition**

The selection of the most appropriate procurement method depends on some different fundamental criteria such as cost, time and quality: they are critical for clients to achieve their goals at the highest quality, at the lowest cost and in the shortest time (Hackett et al., 2007).

Tender analysis processes need hard work from people involved in order to achieve optimal and transparent results (Cid-López et al., 2016): these processes are time-consuming, complex, expensive and error-prone.

This issue is particularly relevant in the railway sector, since it is a significant example of Engineering-To-Order (ETO). In this context, the very first phase of design specification is considered in the tender, but it is strongly revised during engineering design phase, only after contract signing: it is difficult to accurately predict production schedule at the contract stage but, at the same time, it is extremely important because the design phase is substantially unique, and the delivery date should be included in the contract even if few information is available. Moreover, design modifications often occur after production beginning, because of the high similarity between intermediate products.

Despite efforts of Original Equipment Manufacturers (OEM) to create product platforms, railway vehicles market demand is still for highly customized systems. In fact, for a new product, requirements come essentially from technical specification and applicable norms. A typical technical specification can contain up to 10K requirements, due to concurrent international standards, national regulations, existing infrastructures etc. Therefore, technical specifications are enormous, interlinked, non-homogeneous, confused, and even contradictory. They are built on past specifications and on the basis of new constraints. Therefore, they appear as the result of “cut and paste” with very low attention to language and format harmonization.

In case of tenders, specifications have to be developed in a very short time (3-4 weeks), in order to estimate costs and set up the technical and economic offer (up to 4 months). Actually, the technical specification should be the right moment to negotiate requirements with customers. Technical risk in this phase is very high: an extensive requirements analysis can help mitigate risks, therefore systems able to undertake all the previous limitations and issues allow the company to be more competitive and reduce the risk of penalties.

The use of standard Requirements Management (RM) Tools requires time for populating the requirements database; it is usually not affordable at this stage because there are no automatic and consolidated tools to manage this phase in the domain.

The time span between a public request of quotation, with the emission of the "terms of contract", and the bid can be as short as a few weeks. The design and manufacturing can require several months (usually between 18 and 36, with some cases of even longer time periods). The issues related to maintenance and warranty belong to an even wider time frame. These three phases are highly interlinked and the consequences of a purely executed bid (unfortunately winning) reflect on the engineering phase (much more complex and with many modification requests) and later on the entire lifecycle of the system.

Requirements analysis is critical to the success of the development of all products and plays an incredible role in case of big projects where thousands of components, hundreds of suppliers and sub-suppliers, and tenths of employees involved in the design team contribute to the final product. Requirements analysis can be a long and arduous process during which many delicate psychological skills are involved. Experience on the formal specification and verification of software requirements

has shown that many problems occur in the passage from informal requirements, expressed in natural language, to formal models.

## **2.2 General challenges of Natural Language Analysis**

The automatic analysis of technical documentation implies two categories of challenges. The first type is the one common to the processing of all documents written in natural language. The second one is related to technical domain (terms, relationships, standards, etc.).

The access to the knowledge requires understanding the linguistic structures representing content in text at a level of considerable detail. Processing linguistic structures at the depth needed for content understanding presupposes that a considerable amount of domain knowledge is already in place. Moreover, Natural Language is complex and ambiguous. For “Natural Language” we mean any language, written or spoken, which has been formed and evolved naturally through continuous use by humans. Machines, on the other hand, speak by using artificial, simpler, highly predictable, unambiguous languages. “Explaining” to a machine how Natural Language works is not a simple task, but necessary, since within a sentence or a paragraph some words are ambiguous: they can, in other words, have more than one meaning, and must be interpreted on the basis of the rest of the text and the context. Natural Language is subject to polysemy, synonymy and hyperonymy, too:

- Semantic polysemy indicates the property that a word (or even a syntagma or phraseological expression) has to express multiple meanings
- Semantic synonymy indicates the relationship between two lexemes with the same meaning
- Semantic hyperonymy indicates a semantic relationship between two terms, the “hypernym” and the “hyponym”. The hypernym has a larger semantic field than the hyponym and includes it.

This complexity requires the use of Language Technologies to map the underlying conceptual organization of a domain onto the surface language. For the specific task, the last few years have witnessed a growing body of research and practice aimed at developing domain specific ontologies for application in several fields (e.g. legal domain, biomedical domain, etc.). Also, in the field of technical documentation analysis a number of ontologies have been successfully proposed in a variety of research projects, mostly focusing on upper level concepts hand-crafted by domain experts. It goes without saying that realistically large knowledge-based applications in the specific domain will need more and more comprehensive ontologies that should be moreover continuously updated. The problems related to this are basically two: (i) the experts, even the best ones, have a partial vision of phenomena and (ii) ontologies cannot be updated as fast as the field changes, becoming obsolete in a few months (Chiarello, 2018).

To avoid this problem, various techniques for automatically acquiring knowledge from text using information extraction methods have been proposed in the Natural Language Processing (NLP) research community. The NLP consists in applying a software pipeline composed of steps in order to extract information from different kinds of text (Fareri et al., 2020). In this work we used T2K (Text-To-Knowledge) (Dell’Orletta et al., 2006), a system to automatically induce ontological knowledge from texts with an ontology learning system. The system offers a battery of tools for NLP, statistical text analysis and machine language learning, which are dynamically integrated to provide an accurate representation of the content of vast repositories of unstructured documents in technical domains. In particular, we used the term extraction module of T2K specifically developed to acquire domain specific terminology relying on the new contrastive approach described by Bonin et al. (Bonin et al., 2010). The system has been successfully exploited to distinguish between common words and domain-specific terminology in different domains, such as legal domain (Lenci et al., 2009; Bonin

et al., 2010), scientific articles concerning environmental domain (Bonin et al., 2010), product catalogues (Giovannetti et al., 2007) and patents (Fantoni et al., 2013).

The second problem is linked to some elements typical of the domain we are dealing with.

In fact, tenders are usually structured in sections, subsections and paragraphs, containing a certain number of single requirements: they have to be identified. Not only, among these, some are quantitative, and others are qualitative. The latter are more difficult to fit and maybe could be better exploited. Moreover, tenders are characterized by a specific technical jargon and by the presence of normative references to be applied.

### 2.3 State-of-the-art

In order to discuss related works, the authors built the table in Figure 1, classifying most recent and cited papers on the basis of their goal, method and type of results.

The main goals taken into considerations by the authors in the state-of-the-art on requirements analysis are:

- *requirement extraction*: identifying and extracting requirements from different sources. In this case, requirements are “ready to be used” and they have only to be collected (i.e. extracting requirements from the text of a tender);
- *requirement classification*: identifying, extracting and classifying requirements on the basis of different features (functional/non-functional, ambiguous/unambiguous, complete/incomplete, correct/incorrect etc.);
- *requirement elicitation*: identifying and extracting information relevant for requirements understanding from different kinds of text. This information is then analyzed by a specialist to produce requirements. In this case, requirements are not explicitly stated in the text.

The literature on the automatic analysis of texts of requirements through NLP includes different approaches:

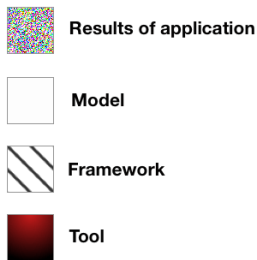
- *expert systems*: these are fed with human knowledge, to solve data problems by imitating the work of the experts. For this reason, its development requires knowledge acquisition from people, involving both knowledge and application domain engineers (Munaiseche et al., 2018);
- *machine learning* (ML): a branch of AI that develops algorithms able to automatically learn from data, identify hidden patterns and make decisions with minimal human intervention (Al-Sahaf et al., 2019);
- *deep learning* (DL): a particular type of ML and is based on computational models and algorithms that imitate the architecture of the biological neural networks in the human brain. In particular, the number of hidden layers of artificial neural networks in deep learning is higher than three and allow to solve complex problems (Akhar & Kaur, 2020);
- *graph-based algorithms*: these are an approach to text summarization. Concepts contained within a text are extracted by using NLP techniques and then represented as nodes in a graph. The edges between nodes represent semantic and topological relations between concepts (Mani & Bloedorn, 1997);
- *question answering systems*: a branch of computer science whose aim is to realize systems able to answer questions expressed in natural language (Damiano et al., 2017), usually querying and finding a match with a Knowledge Base.

The types of results that could be found in the papers are results of application to one or more use cases, models (that are formalizations of design and features), frameworks (that offer a generic

functionality that is the base of tool development) or tools (that are complete from the point of view of functionality, operating and usable).

|        |                           | GOAL                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                         |
|--------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        |                           | REQUIREMENT<br>EXTRACTION                                                                                                                                                                                                                                           | REQUIREMENT<br>CLASSIFICATION                                                                                                                                                                                                                                                                                                                               | REQUIREMENT<br>ELICITATION                                                                                                                                                                                                                                              |
| METHOD | EXPERT<br>SYSTEM          |  (14)  (17)  (6) |  (19)  (20)                                                                                                                                                                               |  (16)                                                                                                                                                                                |
|        | MACHINE<br>LEARNING       |                                                                                                                                                                                                                                                                     |  (8)  (10)  (12)  (18) |  (3)  (4)  (9) |
|        | DEEP<br>LEARNING          |                                                                                                                                                                                                                                                                     |  (1)  (2)  (7)  (15)   |                                                                                                                                                                                                                                                                         |
|        | GRAPH-BASED<br>ALGORITHMS |                                                                                                                                                                                                                                                                     |  (5)                                                                                                                                                                                                                                                                       |  (13)                                                                                                                                                                                |
|        | QUESTION<br>ANSWERING     |  (11)                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                         |

Legend:



(n) Authors

|                                           |                                         |
|-------------------------------------------|-----------------------------------------|
| (1) Jeon, K. et al. (2021)                | (11) Sleimi, A. et al. (2019)           |
| (2) Tiun, S. et al. (2020)                | (12) Abualhaija, S. et al. (2019)       |
| (3) Do, Q.A. et al. (2020)                | (13) Deshpande, G. (2019)               |
| (4) Khan, J.A. et al. (2020)              | (14) Robeer, M. et al. (2016)           |
| (5) Sarmiento-Calisaaya, E. et al. (2020) | (15) Winkler, J. & Vogelsang, A. (2016) |
| (6) Li, Y. et al. (2015)                  | (16) Lucassen, G. et al. (2017)         |
| (7) Sabir, M. et al. (2020)               | (17) Bakar, N.H. et al. (2016)          |
| (8) Shah, U.S. et al. (2020)              | (18) Jindal, R. et al. (2016)           |
| (9) Morales-Ramirez, I. et al (2019)      | (19) Dalpiaz, F. et al. (2018)          |
| (10) Al Kilani, N. et al (2019)           | (20) Rosadini, B. et al. (2017)         |

Figure 1: State-of-the-art table

Analyzing the table, the insights we could give are the following:

- most of the analyzed papers aim at classifying the requirements, that is also the goal pursued by our work; among these, many use Machine Learning and Deep Learning methods, while the method we adopted uses an expert system. The reasons behind the choice of this method are better explained in section 3;
- most of the works dealing with requirement elicitation are applied to other text with respect to technical specifications (e.g. customer reviews), in order to extract relevant information to simplify new specifications drafting;
- the basic extraction of requirements is mostly performed using expert systems, but one paper (Sleimi et al., 2019) shows an attempt to develop a most complex query system based on question answering method;
- only 5 of the 20 analyzed papers develop a tool able to run on servers and be used in real production scenarios, that is one of the main design goals of our project.

For what concerns the different domains addressed by the papers:

- only a few papers are linked to the analysis of physical products specifications (Winkler & Vogelsang, 2016; Rosadini et al., 2017; Abualhaija et al., 2019; Jeon et al., 2021). Among these, only two present applications in real industrial settings, in particular one in the railway sector and one in the transport sector;
- many papers aim at the extraction, classification and elicitation of software requirements (Li et al., 2015; Jindal et al., 2016; Deshpande, 2019; Tiun et al., 2020; Do et al., 2017; Sarmiento-Calisaya et al., 2020; Sabir et al., 2020; Shah et al., 2020);
- one paper deals with legal documents (Sleimi et al., 2019), demonstrating that requirement analysis is not only a task related to the industrial environment;
- many of the analyzed works deal with the domain of users reviews, stories and online discussions, automatically extracting requirements information in order to design new specifications (Robeer, 2016; Bakar et al., 2016; Lucassen et al., 2017; Dalpiaz et al., 2018; Morales-Ramirez et al., 2019; Al Kilani et al., 2019; Khan et al., 2020).

## 2.4 Design goals

The demand for reliable and time-efficient requirements management techniques appears currently not fully satisfied. Actually, standard RM Tools deal more with requirement tracking and tracing during the elaboration of specifications instead of playing a key role in requirement splitting, selection and organization.

Semantic technologies are capable of retrieving relevant information from natural language texts (as it happens for example in personalized advertising on Gmail) and when trained on a specific domain can also help in technical tasks (e.g. patent mapping, specifications analysis, etc.).

The aim of the research was to develop a tool able to assist Project Engineers and System Engineers during the most repetitive, low added-value part of tender phases allowing them to save time and to bring strategic advantage to their companies. Such a software must not substitute but has to integrate the current RM Tools.

The identified functionalities are the following:

- F1. Extraction of Requirements from a continuous text written in natural language (tender)
- F2. Identification of normative references;
- F3. Assignment of each Requirement to the relevant Subsystem(s) (from the structural and organisational perspective);
- F4. Assignment of a "Score" to each Requirement which quantifies the overall significance (from a functional and performance perspectives);
- F5. Direct interface to commercial RM tools (DOORS, Requisite pro, Tiger Pro, etc.);
- F6. Generation of readable annotated documents (easy-readable formats: annotated, highlighted, underlined doc and pdf, spreadsheets, etc.).

## 2.5 Characteristics of the relevant entities

More specifically in tender analysis specific challenges to satisfy the design goals above have to be solved. The point F1 of section 2.4 leverages the ability of the software to recognize the sentence structure (noun, verbs, etc.) and separate phrases disambiguating punctuation signs and distinguish sections, subsections and paragraphs. See example in Figure 2, taken from a real tender.

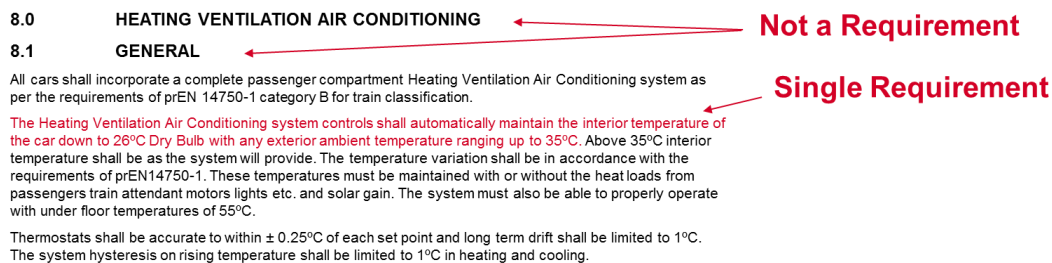


Figure 2. Example of sections to be distinguished in a real tender

The point F2 of section 2.4 is oriented to identify normative references, which the tenders are usually plentiful of, allowing to automatically extract a list of applicable norms for the tender. This was done according to a series of rules as for example through the comparison with a vocabulary of normative fields (e.g.: EN, ISO, TSI, UIC, etc.) or through context-based recognition of the unknown normative by means of: (i) near-normative terms as: “according to”, “in accordance with”, “compliant with”, or (ii) unknown sequences of letters and numbers not belonging to the class of physical units (m, kV, km/h, rpm, etc.), as shown in Figure 3.

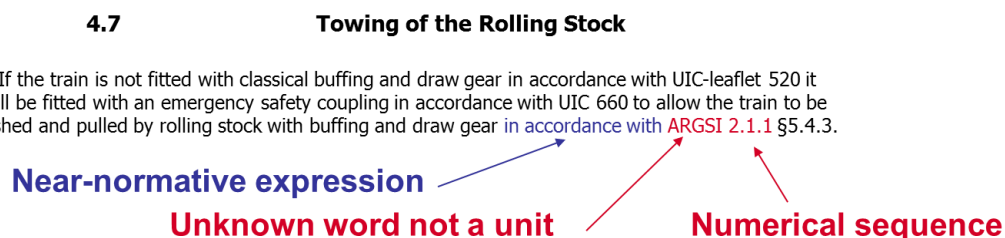


Figure 3. Example of recognition of Normative references

The point F3 of section 2.4 concerns the assignment of each requirement to one or more systems between the ones which the rolling stock is composed of, according to a Product Breakdown Structure (PBS) which is described later in the paper. Albeit classification is mainly driven by recognition of keywords and key-phrases according to a structured Knowledge Base (described later in section 3), this may not be sufficient especially for very short and or itemized requirements, where no keywords or key-phrases are present and/or keywords disambiguation is needed. This issue is solved by context sensitivity: a requirement is assumed to be more probably related to the neighbouring ones. See example in Figure 4.

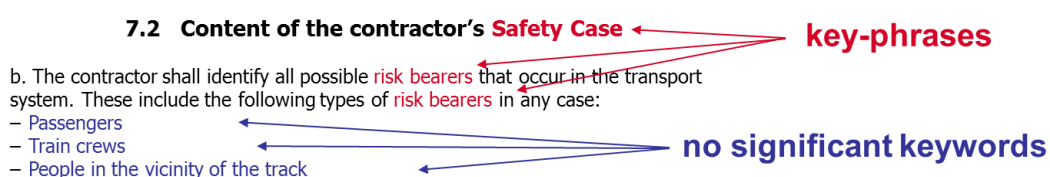


Figure 4. Example of context sensitivity

The point F4 of section 2.4 deals with the quantification of the “quality” of the definition of each requirement. Albeit a tender should contain only objective technical requirements, it is not unusual that “qualitative” requirements are expressed in natural language, as in the example reported below; the discussion of these claims often results in attrition and is a dramatic time stealer for the project team. Early identification of these “bad” requirements by a low “significance score” will be greatly useful in order to ask the issuer to better define, or amend, such statements. See example in Figure 5.

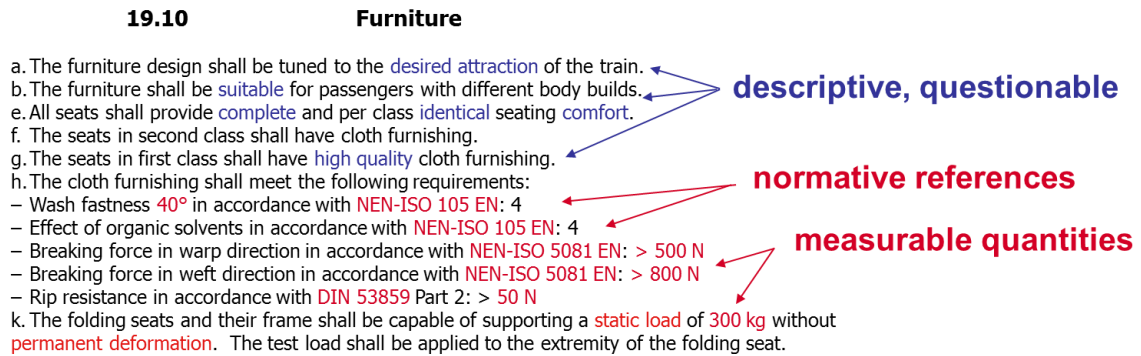


Figure 5. Example of quantitative and qualitative requirements identification

The last two points define the output interface of the software. One is towards Requirements Management commercial tools via Requirements Interchange Format (RIF), a cross-tool, cross-platform, XML-based format for the standardized exchange of requirement information. In this format all the requirements are itemized and the information about system assignment, significance score etc. is present in the database. The same database can be also outputted in spreadsheet format for the non-experts.

The second output option is enriched tagged text: this is an open-source text editor format (.odt) where the complete original text of the tender is re-formatted with a specific focus on legibility, so that critical information is highlighted in a user-friendly way without overwhelming the reader:

- requirements are clearly separated;
- a visual tag shows significance score;
- significant fields (keywords and key-phrases, norms, etc.) are highlighted;
- subsystem assignment is present in the comments field.

### 3. Methodology

The methodology is based on a software module for the identification and splitting of candidate requirements through syntactic and morphological rules. This approach has been chosen on the basis of some considerations: though in the recent literature Machine Learning and, in particular, Deep Learning have proven to be the strongest candidate to deal with NLP tasks, they have some drawbacks that make them not fit for all situations. There are two main costs that can make language models, e.g. BERT (Devlin et al., 2018), unfit. The computational cost and data cost can both be hard to deal with, whereas the first one can always be faced, providing access to faster computational systems, the latter can be prohibitive.

Indeed, as in the case of tenders' domain that is limited, the use of systems involving the use of external sources, could be restrictive. In fact, it is hard to access and organize such a number of tender's texts in order to assemble a training set appropriate to train a language model from scratch. In this situation, a solution could be resorting to a pre-trained language model. However, the particularity of these texts and the difference between them and those employed to train large language models, make this solution hardly feasible too. Moreover, the enormous work of standardization, proceduralization and formalization in this domain, could allow the successful application of an expert system approach, based on ontologies. We thus preferred this approach, so as to exploit the typical structure of tenders, that we know from experience. This way we are able to make a resource from these atypical structures rather than an obstacle.

The software application detects (i) if each candidate requirement is a real requirement and (ii) to which system assigning it. For these two tasks each requirement is compared with the Knowledge Base of systems, subsystems, components, etc. The Knowledge Base has been created by using linguistic tools for domain terminology extraction able to extract and organize the domain terms in ordered trees from general to more detailed and specific terms. Finally, by using a tailored function it is possible to compare each candidate requirement and to assess both its quality and the reliability of the assignment to the right system/subsystem. The workflow is shown in Figure 6.

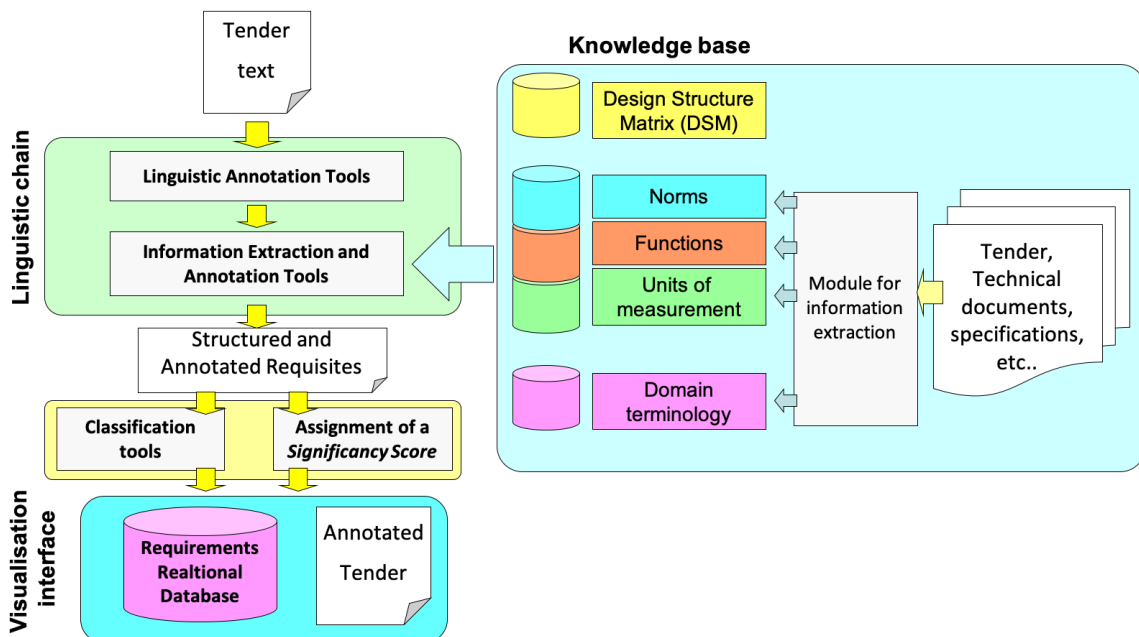


Figure 6. Methodology workflow

The overall procedure we adopted is shown in Figure 6; the developed system is composed of three parts: (i) a linguistic chain, (ii) a Knowledge Base and (iii) a visualization interface.

Once the text of the tender has been processed using the Computational Linguistic tools and transformed in an internal representation, the Knowledge Base is used to disambiguate the lexicon, find all the correlated concepts, and extract the relationship between such concepts. At the end of the process the natural language description has been transformed into a unique, more formal representation of the artefact. Finally, visualization software uses the information about concepts and their relationships to generate concise diagrams and maps that can be used by the human expert/user for further analyses.

The detailed steps are described in subsections 3.1-3.6, in particular: in 3.1 Linguistic Chain module (from tender text to structured and annotated requirements), in 3.2 Knowledge Base construction, in 3.3 the functioning of Classification tools and in 3.4 the process of Significance Score assignment.

### **3.1 Translating terms of contract into requirements**

Generally speaking, the term extraction process consists of a few fundamental steps. The first three are related to the Linguistic Annotation: 1) sentence splitting and token identification, 2) Part-Of-Speech (or POS) tagging, and 3) stemming. After these necessary processes the extraction of domain specific terminology starts by 4) identifying term candidates (either single or multi-word terms) from text, and 5) filtering them from non-terms.

In this application and with reference to Figure 6, the term extraction process, performed in the first part of Linguistic Chain block (in green), already provides partial information about the Significance Score, through the values of relative frequency, contrastive relevance and associative strength. However, the last two phases of the Linguistic Chain are finalized only after the additional knowledge coming from the theory of engineering design and from the database has been brought in. Therefore, we postpone the description of the final part of the chain (in yellow) in subsections 3.3 and 3.4, until after the introduction of the Knowledge Base (section 3.2).

Let's see briefly what each step of the Linguistic Annotation:

- Sentence splitting and Tokenizer. These two software modules split the text into sentences and then segment each sentence in orthographic units called tokens. Sentence splitting is the task of rewriting a single sentence into multiple sentences without losing its meaning (Sulem et al., 2018), while tokenization is the process of identifying the best likely sequence of words from a sequence of text (Sunkpho & Hofmann, 2020).
- POS tagging. It is the process of assigning unambiguous grammatical categories (or morphological interpretation) to words in context. Although the high accuracy scores can reach 97% (in English standard newspaper<sup>1</sup>), POS tagging remains a central problem because a POS tagging error may affect all the following steps of NLP (Watson, 2006; Yoshida et al., 2007). For the present application we make use of the most recent version of the Felice-POS-tagger described by Dell'Orletta (Dell'Orletta, 2009), which has been tailored for the present application.
- Lemmatization. For each word in the text, the POS tagger also detects its root form. Lemmatization is the task of considering as a whole the different forms of a word (Ghavami, 2020). This is especially important for inflected (or sometimes derived) words and the

---

<sup>1</sup> [http://aclweb.org/aclwiki/index.php?title=POS\\_Tagging\\_%28State\\_of\\_the\\_art%29](http://aclweb.org/aclwiki/index.php?title=POS_Tagging_%28State_of_the_art%29), Wikipedia page running under the auspices of The Association for Computational Linguistic.

following analyses (statistical analysis of single word and multi-word) can usually take advantage of such reduction.

- Stop words removal. As in patents a critical role is covered by what we call standard legal sentences, also in tenders a “legal jargon” exists and consists of several expressions that characterise the way institutions (typically government bodies) write requests for tenders. Examples of this kind of expressions are “in accordance to”, “according to”: these structures are useful for identifying relevant elements in tenders (such as standards), but, from a formal point of view, they don’t add value to the analysis. For this reason, even if such sentences play a secondary role in the analysis, it has been decided to remove them to reduce noise in results. Moreover, stop words (common words in language) have been removed by using NLTK<sup>2</sup> list of English stop words, in order to improve the extraction task.

## 3.2 Knowledge Base construction

As stated before, the Knowledge Base is fundamental to identify requirements in the tender text and to assign each one of them to a PBS system. In fact, the Knowledge Base could be related with further corpus of the same domain, to intercept, recognize and extract information from natural language texts pre-processed as described in section 3.1, in an automatic and effective way.

The process of Knowledge Base construction could be described by exploring the differential structures within it: (i) general-purpose Knowledge Base, (ii) rolling-stock domain-specific Knowledge Base. On one side, the general-purpose Knowledge Base consists of information of interest in engineering such as structures, functions and behaviours. The characteristics and extraction of these elements are described in subsection 3.2.1. On the other side, the domain-specific Knowledge Base contains expressions characterizing rail industry domain: its construction is described in detail in subsection 3.2.2. Particular attention was paid to the extraction of norms and measurement units: the process is separately described in subsection 3.2.3, since norms and measurement units are both general and domain-specific, so they are a “transversal” component of the Knowledge Base. Another key element of the Knowledge Base is the Design Structure Matrix, but we postponed its description in section 4 since a significant part of the work was focused on its analysis and improvement.

### 3.2.1 General-purpose Knowledge Base

A well-structured and rich Knowledge Base is a key ingredient to disambiguate, gather, select and organize information from technical documents. Of course, the information of interest in engineering, and therefore in the construction of the general-purpose Knowledge Base, can be divided into three main branches: structures (including the systems, subsystems, components, features, geometries, materials, etc.), functional concepts (which include the user goals and performances) and physical behaviours and properties. They belong to the Function-Behaviour-Structure (FBS) modelling approaches (Umeda et al., 1990; Umeda et al., 1995; Umeda et al., 1996; Gero, 1990; Gero & Kannengiesser, 2002), usually used to design objects but recently also to automatically analyze patents (Fantoni et al., 2013).

Each of the three classes of design entities (functions, behaviours and structures) presents its own characteristics and its own particular embedding on the language side.

Functions are usually represented by verbs, especially when the functioning is described explicitly, in operating condition or by periphrases such as *to achieve* + *noun*.

---

<sup>2</sup> <https://www.nltk.org>

Behaviours can be expressed with a verb (e.g. *triboelectrificate*) but also with *noun + noun* and *adjective + noun* referring to a plurality of physical effects (e.g. *phase change* or *Bernoulli principle*) or simply with single adjectives (e.g. *piezoelectric*).

Structures are mainly substantives belonging to the Bill of Materials (BOM) or PBS, they can also refer to materials or geometrical elements, etc., for each type of entity it is necessary to perform a dedicated extraction procedure, since every level has its own grammatical structure and information can be extracted from different sources.

### 3.2.2 Domain-specific Knowledge Base

Domain terminology plays a key role in general and in technical documents in particular. Both single and, even more important, multi-word terms are fundamental to detect crucial concepts in any technical document. Examples of single words are *brake*, *clutch*, *pump*, etc., while multi-words are *pneumatic brake*, *electrical clutch*, *hydraulic pump*, etc. The value added by multi-word is higher than single ones since they are more specific and less ambiguous.

The automatic extraction of a corpus of terms characterising the different subsystems in the PBS has been done by coherently collecting descriptive technical documents (technical specifications, manuals, technical descriptions, etc.) relevant to each of the first level subsystems of the PBS. We followed an approach to terminology extraction similar to that proposed by Bonin et al. (Bonin et al., 2010).

The input text is firstly tokenized, morphologically analyzed (i.e. POS tagged) and lemmatized passing through a pipeline of state-of-the-art NLP tools for the analysis of English texts. The POS tagged text, obtained with the tagger described by Dell'Orletta (Dell'Orletta, 2009) is searched for on the basis of linguistic filters aimed at identifying a) nouns, expressing candidate single terms and b) POS patterns covering the main nominal modification types which express candidate complex terms. It is the case of morpho-syntactic templates such as *adjective + noun* (e.g. *piezoelectric actuator*), *noun + preposition + noun* (e.g. *coefficient of friction*), etc.

Subsequently, the candidate terms are ranked on the basis of the score of a statistical filter, the C-NC Value measure described by Frantzi & Ananiadou (Frantzi & Ananiadou, 1999) and Vintar (Vintar, 2004).

Afterwards, a contrastive method is applied against the top list of this pre-ranked candidate single and multi-word terms. This procedure allows focusing, firstly, on the retrieval of valid technical terms, thanks to the statistical filters, and secondly on domain pertinence, in two distinct but consequent moments.

The top-list of single and multi-word terms are contrasted firstly against the term list extracted from an open-domain corpus and secondly against a top list of terms acquired from a corpus at the level of the different regulated-domain. In both contrastive phases, the contrastive function (CSmw) newly introduced by Bonin et al. (Bonin et al., 2010) is used. The CSmw score is oriented to prune common words from the list of domain-relevant terms.

The domain terminology algorithm described above is designed for massive text analysis (> 1M words): one of the challenges we encountered was that for some systems, not many documents were easily available, so their corpora of terms were relatively small. The extraction algorithm performed quite poorly in these cases, and an expert validation step showed to be necessary. The experts were identified as specialized workers with more than 7 years of experience in design and requirements analysis. One expert was identified for each PBS system: so, each expert is responsible for the system and for all the subsystems belonging to it. These working figures were involved in the subsequent processes of (i) requirements significance score attribution (subsection 3.4) and (ii) Interface Matrix construction (section 4). In this specific case, the validation process was

carried out by (i) providing the list of elements to be validated for the subsystems to the expert responsible for those specific subsystems (the elements were in the first column of a document in .xls format), (ii) asking each one of them to validate the elements using the second column (in different ways according to the different validations to be carried out).

For each system the algorithm extracted around 200 key-phrases; typically, the extracted terms proved to be not significant anymore over a certain threshold value between 50 and 100. Before the defined threshold, expert judgment stated that up to 60% of recognized key-phrases were valid. Figure 7 shows false positives highlighted in red, that were removed from the Knowledge Base. It is worth noting that the occurrence of invalid key-phrases rises abruptly after #50.

Example: validation of key-phrases for sub-system "Interior appointments"

|                             |                        |                          |                                  |                          |
|-----------------------------|------------------------|--------------------------|----------------------------------|--------------------------|
| 1 hand rail                 | 15 seat frame          | 29 seat bottom           | 43 horizontal handrails          | 57 passenger compartment |
| 2 vertical stanchions       | 16 interior layout     | 30 thick stock           | 44 crash pad                     | 58 steel bolt            |
| 3 seat shell                | 17 interior design     | 31 access panel          | 45 seat type                     | 59 drain hole            |
| 4 rubber flooring           | 18 visible fastener    | 32 design review         | 46 fire safety requirement       | 60 test report           |
| 5 seat cushions             | 19 plymetal sheet      | 33 grab rail             | 47 passenger seating             | 61 structural member     |
| 6 passenger seat            | 20 seating arrangement | 34 permanent deformation | 48 polyester resin               | 62 steel frame           |
| 7 rigidized stainless steel | 21 seat layout         | 35 cab partition         | 49 melamine panel                | 63 cab door              |
| 8 stainless steel           | 22 minimum thickness   | 36 permanent set         | 50 door opening                  | 64 control box           |
| 9 finish panel              | 23 side wall           | 37 sharp edge            | 51 form of construction          | 65 carbon steel          |
| 10 window mask              | 24 ceiling panel       | 38 above the             | 52 color scheme                  | 66 flame spread          |
| 11 transverse seat          | 25 side door           | 39 pocket liner          | 53 sufficient strength           | 67 vehicle structure     |
| 12 seat design              | 26 wainscot panel      | 40 door control box      | 54 fiberglass-reinforced plastic | 68 strength requirement  |
| 13 door pocket              | 27 lining material     | 41 padded insert         | 55 stainless steel bolt          | 69 fire safety           |
| 14 interior liner           | 28 ceiling height      | 42 vertical stanchion    | 56 glass fiber                   | 70 different key         |

Figure 7. Validation of key-phrases for subsystem "Interior appointments"

Coming to single words belonging to a specific domain, the algorithm can potentially extract hundreds of keywords, which typically become not significant after 200. Below this threshold, expert judgment stated that up to 80% of recognized keywords were valid but most of them (>90%) cannot be deterministically associated to only one subsystem. The reason relies on the inter-correlation among subsystems. Actually, very few "golden words" can really define an unambiguous link to one single system. Therefore, during the domain keywords screening it is important to distinguish between nondeterministic and wrong terms such as misspells; albeit less valuable, non-deterministic terms shall not be disregarded as they contribute significantly to the automatic text comprehension when aided by context sensitivity and system interface recognition. Figure 8 highlights nondeterministic terms in yellow and wrong terms in red.

Example: validation of keywords for sub-system "Heating Ventilation and Air Conditioning"

|                  |   |                |   |                 |   |                     |   |              |   |
|------------------|---|----------------|---|-----------------|---|---------------------|---|--------------|---|
| 1 refrigerant    | n | 15 compressor  | n | 29 solenoid     | n | 43 dataset          | n | 57 set-point | n |
| 2 valve          | x | 16 digital     | n | 30 baseboard    | n | 44 over-temperature | n | 58 gas       | n |
| 3 ventilation    | n | 17 volume      | n | 31 air          | n | 45 controller       | n | 59 screen    | n |
| 4 climate        | n | 18 duct        | n | 32 monitor      | n | 46 humidity         | n | 60 furnace   | x |
| 5 fan            | n | 19 slave       | n | 33 boiler       | n | 47 blowers          | n | 61 leak      | n |
| 6 evaporator     | n | 20 liquid      | n | 34 conditioner  | n | 48 flow             | n | 62 cursor    | x |
| 7 fresh          | n | 21 heat        | n | 35 byte         | n | 49 floor            | n | 63 coil      | n |
| 8 heating        | n | 22 cooling     | n | 36 on/off       | x | 50 relay            | n | 64 outlet    | n |
| 9 node           | x | 23 thermal     | n | 37 capacity     | n | 51 thermostatic     | n | 65 valve     | n |
| 10 blower        | n | 24 temperature | n | 38 og           | x | 52 coolant          | n | 66 fins      | n |
| 11 expansion     | n | 25 fiberglass  | x | 39 conditioning | n | 53 contactor        | n | 67 list      | x |
| 12 heater        | n | 26 condensing  | n | 40 para         | x | 54 dampers          | n | 68 filter    | n |
| 13 refrigeration | n | 27 answer      | x | 41 extractors   | n | 55 coils            | n | 69 overhead  | n |
| 14 condenser     | n | 28 suction     | n | 42 extractor    | n | 56 undercar         | x | 70 pressure  | n |

Figure 8. Validation of keywords for subsystem "Heating Ventilation and Air Conditioning"

After the processing, the entire set of domain terminology filled the Knowledge Base that was organized according to the various subsystems (PBS) and divided in (i) single-word terminology (e.g. brake, door, etc.), and (ii) multi-word terminology (e.g. emergency brake, emergency brake handle, driver's cab outdoor, etc.) and classified in four further classes:

- gold terminology: extracted from texts considered highly reliable;
- discriminating terminology: terms highly discriminating with respect to the subsystems;
- non-discriminating terminology: terms only mildly discriminating

- stop words: general terms to be removed before the following weighting and classification phases to reduce noise.

Such organization allows the system to weigh the different information extracted and annotated on the requirements in different ways.

### 3.2.3 Norms and measurement units extraction

From an algorithmic perspective normative references, national and international standards and units of measurement are really intertwined since they all are sequences of letters and numbers not recognized by the linguistic chain.

A hybrid approach has been used for both standards and units of measurement identification.

For what concerns the identification of standards, it has been done firstly by a series of rules of normative fields (e.g.: EN, ISO, TSI, UIC, etc.) and secondly through context-based recognition of near-normative terms (*according to*, *in accordance with*, etc.) and unknown sequences of letters and numbers not referring to physical units.

The creation of an exhaustive list of units of measurement has been done by mixing different approaches: from data gathering to Machine Learning. First a series of standard open access sources of units of measurement (as for example Wikipedia) have been merged with commercial ones. The suitable algorithm was trained through standard expressions and sentences containing such units of measurement detected in the documents. Finally, similar sentences not containing the already known units of measurements have been found and analyzed manually. Such an approach lets us detect really particular units of measurements as for example “persons per square meters” or “passengers/m<sup>2</sup>” or “persons/sec” and to build a wide repository (more than 3000 units) and detection rules.

### 3.3 The Classification Function

The input of the Classification Function are annotated requirements, used to assign requirements to PBS systems. Once the keywords present in a requirement are recognized, it can be attributed to the system to which the intercepted words belong.

Given a previously annotated requirement, the Classification Function calculates a probabilistic score for the assignment to each subsystem in the PBS. Using suitable weights (fine-tuned through empirical experiments and the use of optimization algorithms), the score is computed combining the following variables: (i) the number of domain multi-word terms, (ii) the number of domain single-word terms and (iii) context information as for example the classifications of previously analyzed requirements and their relative scores.

Since a requirement can be related to more than a single subsystem the Classification Tool is able to provide a multiple assignment and determine automatically an assignment score (that measures the confidence in the assignment). All subsystems are ordered according to the assignment score and the first three of them are selected on the base of the following constraints:

- value of the score assigned to subsystem x;
- value of the ratio between the score of x and that of subsystem x-1;
- external constraint on the maximum number of assignable classes (usually 2 or 3).

### 3.4 Assignment of a Significance Score

The Classification module elaborates the information in the annotated requirement to assign a Significance Score to the requirement, via a statistic function. It allows to evaluate (i) if a part of the

text is a requirement, (ii) the completeness of a requirement (the lower the score, the lower the completeness). The Significance Score is computed combining the following variables:

- the number and quality ("gold", discriminating, non-discriminating) of (a) domain single-word terms (b) domain multi-word terms;
- the presence of (c) normative references (d) units of measurement (e) functional verbs (f) terms indicating properties;
- the presence of an interface between the subsystems assigned during the classification phase. In particular, the Classification module introduces a penalty if there is no interface between the subsystems and a bonus if there is an interface between second level subsystems.

In particular, the occurrence of each type of object (normative reference, presence of interfaces, discriminating multi-word terms, etc.) in the sentence is summed and the total is divided by the total number of words in the sentence. The sum also takes into consideration the weights assigned to each object as explained in the previous subsection.

The resulting value can go from 0 to 1, with a threshold value of 0.05 below which the text string in question is not recognized as a requirement. This threshold has been defined experimentally together with AnsaldoBreda S.p.A. (now Hitachi Rail Europe) experts after some registration tests. If a text is not recognized as a requirement, it is however proposed to assign it to the PBS systems in order to be able to have a verification by the technical expert.

## **4. Interface Matrix**

The Interface Matrix, as previously said, is not only a source used for the requirements identification and assignment, but also one of the main outputs of this work. In fact, the improvement of the already existing Design Structure Matrix was a necessary condition in order to use this element for feeding the Knowledge Base.

### **4.1 Differences between Design Structure Matrix and Interface Matrix**

Product components are usually grouped into modules that are assembled using a specific design architecture to facilitate future design changes, product variety management, mass customization and manufacturing processes using delayed product differentiation (AlGeddawy & Elmaraghy, 2011). One of the most common tools used to represent interactions among components in a product is the Design Structure Matrix (DSM) (Eppinger & Browning, 2012).

DSM elements are usually represented by binary values: '1' indicates interaction, while '0' indicates its lack. Grouping product components into modules can be accomplished by clustering DSM into blocks consisting mostly of '1' elements. Different techniques can be used to cluster them into modules (AlGeddawy & Elmaraghy, 2013).

In case of rolling stock, a standardised PBS is defined at transnational level in European Norm EN15380-2 with the aim of easing the exchange of information about requirements between the different bodies; this PBS is adopted by AnsaldoBreda S.p.A. (now Hitachi Rail Europe) as well as, likely, by the other OEMs. This PBS is tree-structured into three levels of increasing detail (Level 1: 18 elements, Level 2: 105 elements, Level 3: 651 elements) and is essentially based on train physical architecture.

An additional list of "non-physical systems" have been added by AnsaldoBreda S.p.A. (now Hitachi Rail Europe) in its internal PBS, in order to track those requirements pertaining to a single engineering discipline but not related to any particular component; examples are noise emission, fire safety and RAMS (Reliability Availability Maintainability & Safety).

The main issue in AnsaldoBreda S.p.A. (now Hitachi Rail Europe) is that a train is a complex system because it consists of many subsystems (mechanical, pneumatic, air conditioning, etc.). These subsystems interact with each other. Because of these physical interactions, there are also organizational interactions between the responsible of each area/system. The idea is that these organizational interactions between different responsible people become crucial in the first phase of tender analysis. The internal worker in charge of assigning each part of the tender to one area should take into consideration these interfaces between subsystems because probably one part of the tender influences and needs the effort of more than one responsible worker.

The developed Interface Matrix (IM) is more than a simple DSM, because it is linked to organizational and decision-making aspects.

### **4.2 The Interface Matrix in more depth**

Even if the PBS has a clear mechanical/electrical hierarchical structure, non-physical systems and the interfaces among all the subsystems are difficult to be managed. Actually, interfaces mean design dependency and domino effects between one decision on a system and its consequences beyond the system itself.

Let us take as example the following requirement taken from a real tender:

*“Incidental deformation of the carbody, vibrations and pressure differences between the different areas that occur in normal operation shall not affect the functioning or adjustment of the doors”*

While this requirement pertains to one single system (the doors), the essential technical information needed for the system design cascades from a variety of non-physical systems involved: structural mechanics (deformation of the carbody), running dynamics (vibrations) and aerodynamics (pressure differences). This is a typical case of an interface with a non-physical system, but many cases of interfaces that involve different physical systems exist as well.

In an interface requirement like the one mentioned before, there will be keywords linked to different systems. In order to judge whether the presence of heterogeneous keywords is a symptom of bad classification, or on the contrary a sign of interface requirement, the need for a DSM arose. The problem was solved by means of a so-called Interface Matrix.

An IM has been built by using all the elements of PBS up to second level (155 counting also the non-physical systems) on rows and columns of a square matrix. Each of the 24,025 cells shows the type of the interface, if present, between its row and column systems according to the following list: 0 = no interface; 1 = mechanical interface; 2 = electrical interface; 3 = signal interface; or the possible combinations of the three (4, 5, 6, 7 are intuitively defined). Moreover, interfaces with/between non-physical (N-P) systems are classified through logic operators Yes/No.

The type of interface has been inserted in order to ease the interface detection by the software, thanks to very specific words in the sentence (e.g. voltage, current, or unit of measurement).

Due to the lack of “big data”, the Interface Matrix has been built by hand relying on expert judgment; this has been a long and tedious work and clearly limits the size of what is practically realizable. In principle, the same IM can be obtained in an automatic way if a proper-sized, structured Knowledge Base is present.

The depth of the architecture hierarchy of components, modules and subassemblies defines its granularity (AlGeddawy & ElMaraghy, 2013). The granularity of the description allows also to properly group subsystems and to have a good matching between subsystems and responsible people.

The final result obtained is a square matrix (shown in Figure 9), symmetrical by definition, where the following interfaces have been defined: 1661 mechanical interfaces; 1132 electrical interfaces; 859 signal interfaces; 302 multiple interfaces; 2267 interfaces with/between non-physical systems. IM is therefore a sparse matrix; this mathematical property has led to an intuition on a way to interpret the degree of inter-system connections and therefore the quality of DSM representation.

|    |                                       |                       | DF<br>Additional<br>devices | EA<br>Running<br>gear | EB<br>Supportin<br>g<br>structures | EC<br>Wheelset<br>s,<br>bearing,<br>guide,<br>wheels | ED<br>Suspensi<br>on,<br>damping,<br>balancing<br>gear | EE<br>Driving<br>systems<br>(active) | EF<br>Driving<br>systems<br>(passive) |
|----|---------------------------------------|-----------------------|-----------------------------|-----------------------|------------------------------------|------------------------------------------------------|--------------------------------------------------------|--------------------------------------|---------------------------------------|
| FD | Power conversion                      |                       | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
|    |                                       | TCU                   | 0                           | 3                     | 3                                  | 3                                                    | 0                                                      | 0                                    | 0                                     |
|    |                                       | Traction converter    | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
|    |                                       | Traction motor        | 0                           | 1                     | 1                                  | 1                                                    | 0                                                      | 0                                    | 0                                     |
|    |                                       | Brake resistor        | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
| FE | Power dissipation                     |                       | 0                           | 2                     | 2                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
| FF | Power storage                         |                       | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
| GA | Control apparatus for train operation |                       | 0                           | 3                     | 3                                  | 3                                                    | 3                                                      | 3                                    | 3                                     |
| GB | Power supply controls                 |                       | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
| GC | Driving and braking controls          |                       | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
|    |                                       | TCMS                  | 0                           | 3                     | 0                                  | 0                                                    | 3                                                      | 3                                    | 3                                     |
|    |                                       | ATC                   | 0                           | 3                     | 0                                  | 3                                                    | 0                                                      | 0                                    | 0                                     |
|    |                                       | ATO                   | 0                           | 3                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
|    |                                       | WSP                   | 0                           | 3                     | 0                                  | 3                                                    | 0                                                      | 0                                    | 0                                     |
|    |                                       | ERTMS                 | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
| GD | Power generation controls             |                       | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
| GE | System data acquisition               |                       | 0                           | 3                     | 3                                  | 3                                                    | 0                                                      | 0                                    | 0                                     |
| GF | Power storage controls                |                       | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
| HA | Auxiliary operating equipment         |                       | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
| HB | Converter equipment                   |                       | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
|    |                                       | Auxiliary equip conv  | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
|    |                                       | Auxiliary transformer | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
|    |                                       | Battery charger       | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
| HC | Battery device                        |                       | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
| HD | On-board supply                       |                       | 0                           | 0                     | 0                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |
| HE | Cooling unit supply system            |                       | 0                           | 1                     | 1                                  | 0                                                    | 0                                                      | 0                                    | 0                                     |

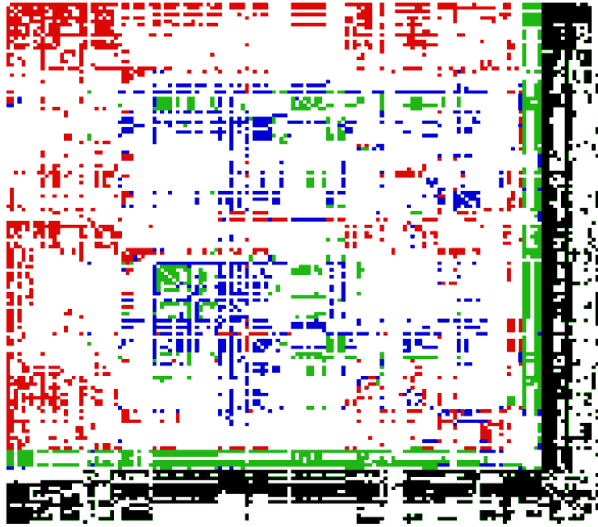
Figure 9. Extract of Interface Matrix

Based on experts judgment, an effort has been made to modify the standard PBS by rearranging the low-level components in different groups (which, in mathematical terms, can be described as changing the base of the IM vector space without modifying its dimension) in order to be more adherent to common engineering practice and reduce the number of interfaces between different systems and disciplines. The result is shown in Figure 10: the same IM based on improved PBS representation appears less sparse than the original one, with more elements clustered around its diagonal.

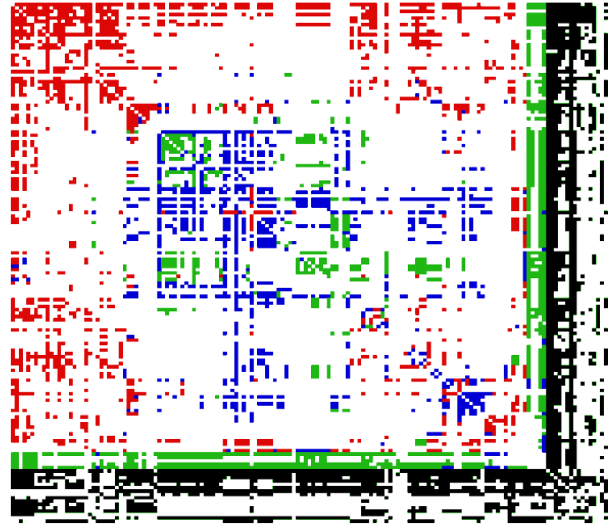
This suggests a way to “measure” a DSM based on mathematical properties of its IM (diagonality, sparsity), or on the other hand, a method to improve a DSM by applying a base change driven by any of the many diagonalization/decomposition algorithms available. On a purely theoretical basis, we could state that the optimal DSM is the one whose components are the eigenvectors of its IM.

In practical terms, the improvement of the PBS appears clear when considering: (i) higher separation of interface types (mechanical, electrical, signal); (ii) clusterization of interfaces within higher-level subsystems; (iii) reduction of point-to-point correlations. This implies several advantages as an easier assignment of requirements to areas of expertise, a more efficient support in writing technical specifications, an easier processing of Failure Mode, Effects and Criticality Analysis (FMECA), Interface Hazard Log, etc.

**DSM based on old PBS**  
**(bad representation)**



**DSM based on new PBS**  
**(improved representation)**



**Legend**

■ **mech. interface** ■ **elec. interface** ■ **signal interface** ■ **N-P interface**

*Figure 10. Comparing DSM based on the old PBS and DSM based on the new PBS*

The final matrix of the interfaces becomes a key element used by the algorithms for the identification and allocation of requirements to the responsible team.

Actually, when the software identifies in a requirement terms simultaneously related to different subsystems, before assigning the requirement to multiple systems it checks the presence of the correct interface in the new DSM.

Actually, the same term can have different meanings and thus it belongs to different subsystems only for linguistic reasons. Only if interface reasons exist the requirement is assigned to different subsystems.

## 5. Results

### 5.1 Validation of the results

To validate the entire process AnsaldoBreda S.p.A. (now Hitachi Rail Europe) chose as a test bed the terms of contract related to the tender for a high-speed train. The entire document was 245 pages long, contained 97320 words and had a size of 1,27 MB. In 2 minutes and 12 seconds the software application extracted 8269 strings of text: 6169 of them have been identified as Technical Requirements. All of them have been assigned to the subsystem of the new DSM. In order to test the reliability of the software, three different validations have been carried out, as explained in the next paragraphs.

#### 5.1.1 Standards identification validation

In about 6000 Technical Requirements identified by the software, 543 of them contain references to technical standards whose acronyms are registered in the Knowledge Base. 100 requirements have been selected from the processed specifications divided into two halves: 50 indicated by the tool as containing references to standards and 50 indicated by the tool as not containing references to standards. Then, the correctness of the identifications was checked by the project manager of the bid (since he has an overview of the whole system and the entire process). Globally:

- all the 50 requirements that the tool indicated as not containing references to standards, were verified by the human expert and did not contain references to standards (true negatives = 50; false negatives = 0);
- the tool correctly identified 49 standards out of 50 (true positives = 49);
- the tool identified only one false positive;
- the tool did not identify as standards some acronyms of train names that can lead to a wrong assignment (such as *RLN000453*).

To conclude, we could say that:

- the precision, the ratio between the number of true positives and the sum of true positives and false positives, is 98%;
- the recall, the ratio between the number of true positives and the sum of true positives and false negatives, is also 98%.

The 50 requirements used for the testing represent about 10% of those referring to technical standards in the processed tender, so they are a statistically sufficient sample to make assessments on the reliability of the algorithm.

Basically, only one assignment error was found in the selected sample. On the basis of previous considerations, it can be inferred that the software is more than 90% reliable in identifying the international standards in a tender.

#### 5.1.2 Requirement completeness validation

Also in this case, two samples were selected from the output of the tool: 50 sentences which the tool classified as requirements and 50 sentences which the tool classified as non-requirements. The two samples contain the 50 elements listed in ascending order, according to the value attributed to the completeness of the information.

The following considerations can be made by scrolling through the table of requirements:

- the first 10 requirements are marked by the minimum threshold value (0.05); therefore, their information content has not been considered exhaustive or complete by the software. A significant example has been highlighted: one of the requirements in the text prescribes that "the requirements related to fire safety are all met" but does not say exactly what they are. Searching for this sentence within the specifications, it is found that it is a criterion that precludes the acceptance of the corridors of communication between one carriage and another, a detail therefore not negligible. The low value attributed to this requirement represents a warning to the designer who will have to investigate in detail which parameters are actually to be met for the component in question. Moreover, we highlight the software correctly assigns this requirement to the following three systems: vehicle body, fitting out; coupling system;
- as the value of completeness increases, the requirements become more and more comprehensive;
- in the completeness assessment the occurrence of first or second level systems in the text of the PBS structure increases the value attributed to a requirement (some requirements do not contain technical specifications but have a high score due to the presence of several PBS systems in the text);
- the requirement completeness is weighed on the number of words contained in the text (as explained in subsection 3.4). This is why short text strings (e.g. paragraph headings) containing only keywords or the name of one of the first or second level PBS systems are rated higher than the unit. In fact, even if these are no longer requirements, they are a very important reference for the software in the context analysis in the next phase of assigning text to PBS.

Analyzing the table of non-requirements, we would like to highlight the following remarks:

- most of the strings do not describe technical characteristics, but they are management or commercial requirements. Since the tool aims at identifying only technical requirements, they are not identified as requirements. However, this information is not lost: the requirement is assigned to one or more PBS systems and therefore to the most suitable team of designers able to analyze it;
- swiping the table, as the score approaches the threshold value, there are some cases in which the software did not recognize requirements. This happens for the inverse dependence between string length and completeness value, with the same number of identified keywords (thus, in the last rows of the table there are long sentences).

The tool correctly identifies technical requirements in the text, associating them with a higher value as their content becomes more specific. It also allows to focus attention on requirements with low scores, which may contain important information for design purposes. Evaluating the quality of the proposed tool, we intentionally avoid making statistical considerations on the values attributed to single requirements. For this purpose, it would be necessary to make the tool processing a sufficient number of text strings (at least one thousand), previously evaluated by an expert, as done for the following test.

### **5.1.3 Requirement assignment validation**

As previously said, each one of the 6169 Technical Requirements identified by the software have been automatically assigned to the subsystems of the new DSM (the Interface Matrix). In particular, the software gives a ranking of the most likely subsystems to which the requirement could be assigned, following the process described in subsection 3.3. The software assignments were

compared to the assignments of a human expert. In particular, in order to test the reliability of the software assignment, AnsaldoBreda S.p.A. (now Hitachi Rail Europe) provided a file with 2606 requirements manually extracted from the same tender and assigned to the subsystems by the manager of the bid. This allows a direct comparison between the assignments of the human expert and those of the software.

Remarking that the tool returns a ranking of the most likely subsystems for each requirement, the correctness of the attributions was verified by measuring two different reliability indexes for each subsystem:

- (i) ratio between the number of requirements correctly assigned to one subsystem by the tool and the number of requirements assigned to that subsystem by the human expert, by checking if the first subsystem in the ranking of tool assignments is the correct one;
- (ii) ratio between the number of requirements correctly assigned to one subsystem by the tool and the number of requirements assigned to that subsystem by the human expert, by checking whether the first or the second subsystem in the ranking of tool assignments is the correct one.

The two ratios are then transformed in percentages. As expected, the percentage of correct assignments considering if the first or second subsystem in the ranking is the correct one is higher than the percentage of the correct assignments considering if only the first subsystem in the ranking is the correct one. In fact, considering the first two subsystems of the ranking, the probability that one of the two is the correct one increases. The percentages for each subsystem are shown in Figure 11.

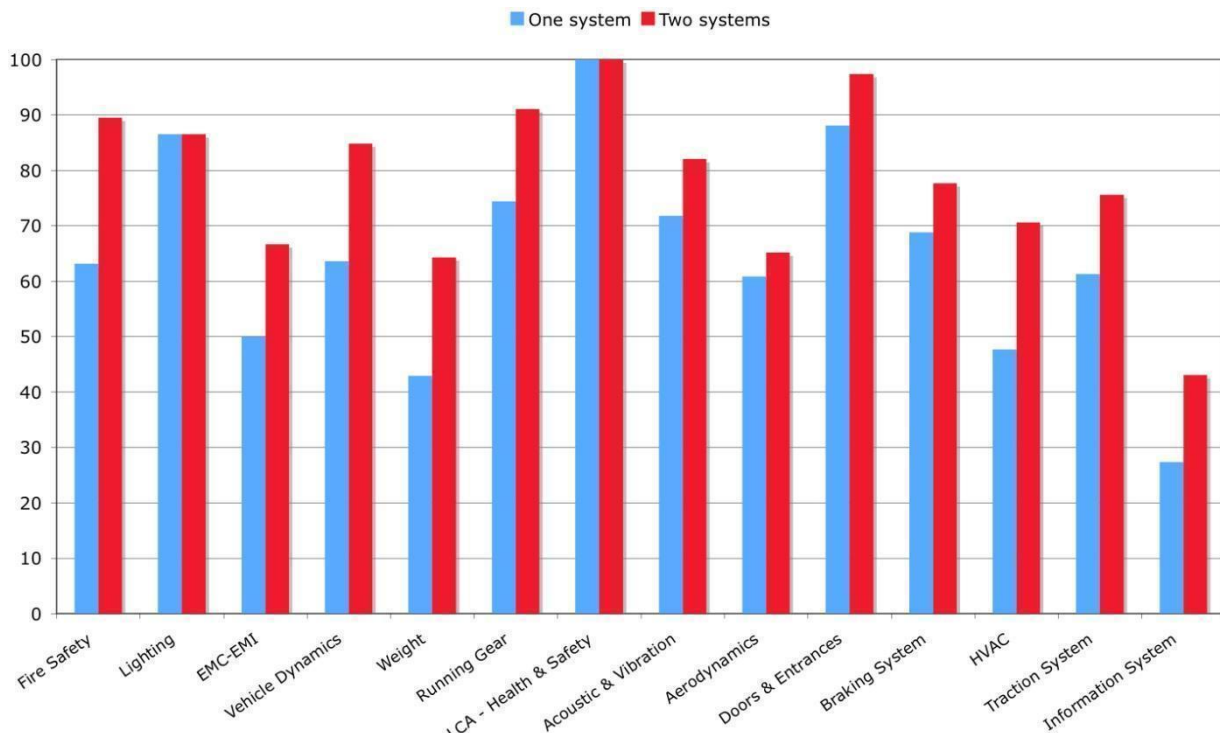


Figure 11. For each subsystem, percentage of correctly assigned requirements according to the first index (i) in blue and to the second index (ii) in red

The global percentages of requirements correctly assigned by the software are shown in Table 1: in the table was also added the percentage of correctness achieved by considering the first three subsystems of the tool ranking.

|                                                                                 | Percentage of requirements correctly assigned by the software |
|---------------------------------------------------------------------------------|---------------------------------------------------------------|
| Verification of correct assignment in the first subsystem of the ranking        | 60%                                                           |
| Verification of correct assignment in the first two subsystems of the ranking   | 76%                                                           |
| Verification of correct assignment in the first three subsystems of the ranking | 84%                                                           |

*Table 1. Percentage of requirements correctly assigned by the software in the first subsystem, in the first two subsystems and in the first three subsystems of the ranking*

Verifying the first three most likely assignment options the percentage of reliability finally reaches the 84%. After the third subsystem of the ranking, the assignment probability to the other ones of the sequence significantly declines: this led to the choice to set the number of subsystems suggested by the tool in its final version to three.

#### **5.1.4 Threats to validity**

Discussing in further details the validity of the proposed tool, our research has some limitations that are typical of empirical validation processes.

Firstly, the Knowledge Base construction process could be improved by adding new input documents that we might not have considered. We could have also missed some keywords or POS patterns crucial for the identification of Knowledge Base elements. Although adding new inputs and new extraction rules could for sure improve the performance of the Knowledge Base in the extraction and classification task, we think that the validation process shows more than confident results.

Secondly, the comparison between the output of the tool and the human expert analysis could lead to an unreliable validation in case of human mistakes in requirements identification and classification. In our case, this effect was mitigated by assigning the manual task to the project manager of the bid, a senior manager with almost 7 years of experience.

Finally, due to the high volume of material processed, a verification by individual experts (as the one conducted in subsection 5.1.3) also for the steps described in subsections 5.1.1 and 5.1.2 has been considered out of scope for the present research. In these cases, the reliability of the tool could be measured only through its use, examining in detail the systems that are recognized with greater difficulty, which may have too little Knowledge Base or contain incorrect terms.

## **5.2 Final tool**

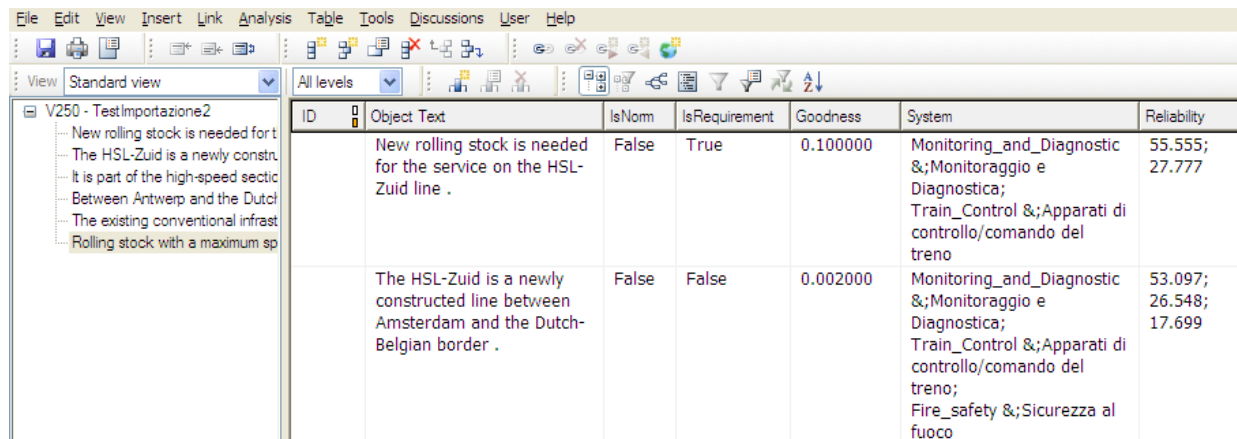
The tool usability is one of the most relevant aspects: a simple graphical interface that allows the use of the tool without accessing the operating system has been designed.

The tool gives as output a RIF file that could be imported not only in DOORS<sup>3</sup> (the RM Tool) but also in other formats (.docx, .xls, .txt). The RIF output file stores the extracted information according to

<sup>3</sup> <https://www.ibm.com/us-en/marketplace/rational-doors>

the general structure defined by the RIF format. RIF is a standard format for exchanging information and is among the best-known tools for writing and managing requirements (e.g. DOORS, Requisite Pro, etc.). The RIF is organized in an XML schema, easy to be managed by standard Requirements Management Software, and embeds the structure of requirements and their attributes, types, access permissions and relationships.

In Figure 12 the RIF file once imported in DOORS is shown. The format of the output file allows the human expert an easy verification, correction and management of the information extracted by the system. It is easy to use, filter, customize etc. also for non-experts, it contains complete information, particularly suitable for testing purposes.



| ID | Object Text                                                                               | IsNorm | IsRequirement | Goodness | System                                                                                                                                           | Reliability            |
|----|-------------------------------------------------------------------------------------------|--------|---------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
|    | New rolling stock is needed for the service on the HSL-Zuid line .                        | False  | True          | 0.100000 | Monitoring_and_Diagnostic & Monitoraggio e Diagnostica; Train_Control & Apparat di controllo/comando del treno                                   | 55.555; 27.777         |
|    | The HSL-Zuid is a newly constructed line between Amsterdam and the Dutch-Belgian border . | False  | False         | 0.002000 | Monitoring_and_Diagnostic & Monitoraggio e Diagnostica; Train_Control & Apparat di controllo/comando del treno; Fire_safety & Sicurezza al fuoco | 53.097; 26.548; 17.699 |

Figure 12. The output file once imported in DOORS

For each Object Text (first column), that are the sentences in which the tender has been splitted, is indicated:

- (i) the presence or not of normative references (isNorm column);
- (ii) if the Object Text is or not a requirement (isRequirement column);
- (iii) the score given to text information completeness in order to decide if it can be considered a requirement or not (threshold set at 0.05, Goodness column);
- (iv) the assigned subsystem(s) (System column);
- (v) the probability percentages of requirement assignment to each one of the systems of the System column (Reliability column).

Moreover, the tool gives as output the enriched tagged tender text. It focuses on legibility, a clear separation of requirements, visual tag on significance score, the text highlights keywords and key-phrases, norms, etc. Subsystem assignation is present in the comments field.

## 6. Open Issues and Future Development

Considering the results of the present paper, it has to be considered that the project rollup was in 2010. Due to the embargo, the results could be published only now. This implies that the development, not so much in terms of text analysis techniques and software, but rather that related to the processes of AnsaldoBreda S.p.A. (now Hitachi Rail Europe) have partially modified the current company reality, which is very different from the one described in this paper.

However, the adopted methodology and the results shown in this work remain significant. In fact, AnsaldoBreda S.p.A. (now Hitachi Rail Europe) reality and problems, explained in section 1 and section 2, are still relevant today for many companies. The need for tools as the one described in the present paper, emerges also in the research trajectory that usually anticipates the business needs. In fact, searching for the query (*“requirement analysis” OR “requirement engineering”*) AND (*“automatic” OR “automated”*) AND (*“natural language processing” OR “NLP” OR “text mining”*) on Scopus Database<sup>4</sup>, we can notice an upward trend, as shown in Figure 13. The query was built associating a block of expressions about requirement analysis (comprising the expression “requirement engineering” that is the process to which requirement analysis activity belongs, this was necessary because in many papers requirement analysis was more generally referred as requirement engineering), to a block of expressions about automation and one about NLP. The logic underpinning the query takes into account all the papers focused on the application of text mining techniques for automating the requirement analysis activity.

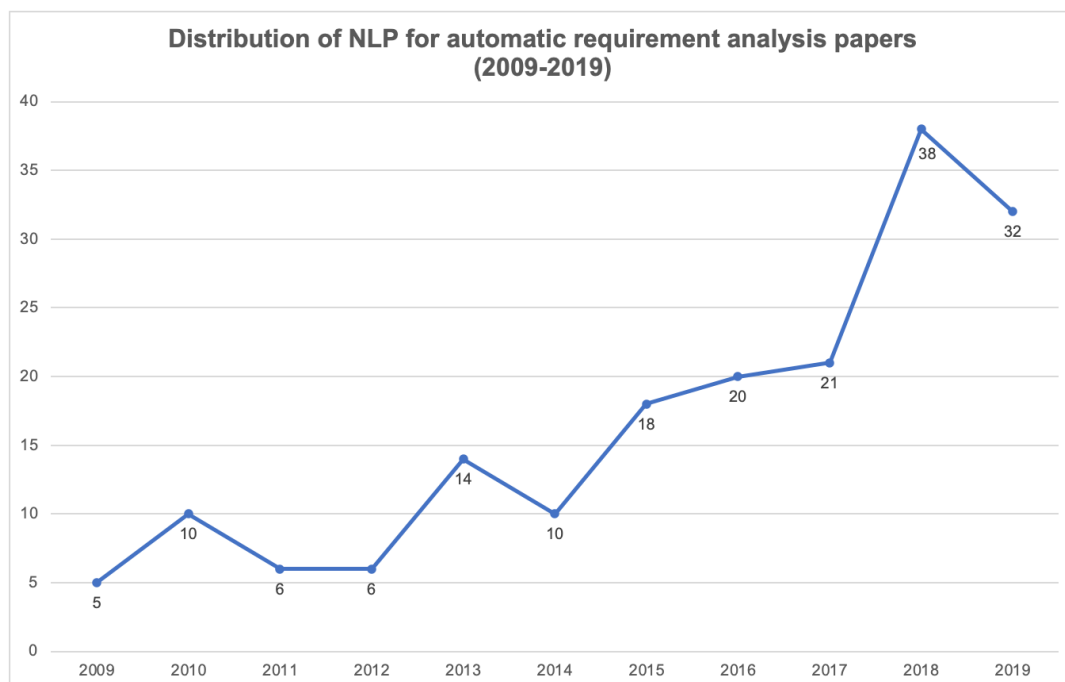


Figure 13: Distribution of NLP for automatic analysis papers

From a business perspective, the relevance of the topic is demonstrated by the attention that DIN SPEC 91345:2016 norm pays to the assets in the information world. The norm presents a reference architecture for standardizing Industry 4.0 paradigm, starting from the concept of “asset”. Assets are entities that create value for a company and exist not only in the physical world but also in the human world: for this reason, they could be stored in a .txt file in a computer or in a paper document, but also in the human brain. Anyway, an asset that in the physical/human world exists in different

<sup>4</sup> <https://www.scopus.com>

instances, is always the same in the information world. The aim of Industry 4.0 is “translating” physical assets in information assets, that is exactly the big aim behind this work.

On the basis of the activities described in this paper, the project reached the following results:

- the developed tool for the automatic analysis of tenders is able to analyze a 245 pages document with 97320 words in almost 2 minutes;
- the tool recognized 6169 technical requirements with a reliability of around 90%;
- the tools assigned each requirement to the subsystems of the PBS with more than 80% reliability;
- the tool is able to identify and evaluate the completeness of the technical requirements, correctly attributing a significance score whose value increases with the completeness of the information within them. This feature allows the recognition of ambiguous requirements that could generate waste of the time and errors during tender analysis;
- a Knowledge Base containing more than 50 standard abbreviations, more than 3000 units of measurement and almost 16000 technical domain-specific terms, has been developed and embedded in the tool. The 16000 terms are organized in the Knowledge Base according to a train PBS, but taking into account also those systems defined in the paper as "non-physical", i.e. those systems whose requirements have a cross-cutting impact on the design of all devices;
- the new Interface Matrix, crucial for the algorithms of requirements identification and assignment, was built.

Summing up, the results achieved allow a significant reduction of the time needed to read a tender text and decompose it in requirements in the first stage of preparation. Moreover, since requirements are immediately assigned to the three most likely PBS subsystems and associated to a score that defines their completeness and non-ambiguity, the reading process of the human expert is more effective and could be focused on details.

However, there are still some aspects to work on to improve the software performances:

- for some subsystems, the documentation is poor and doesn't allow to make substantial analyses;
- the training sets on which the optimization algorithms run for computing the weights in the classification and assignment functions are a few;
- some subsystems are not well-separated from the others so the documentation can be embedded in other sets;
- the results of correct assignment have been performed by checking the system against a human being. The project manager of the bid has been chosen since he managed the systems and knew the entire process and problems he encountered during the requirements-to-specifications process, but this could be a cognitive bias we introduced in the work.

Moreover, there are many opportunities for applying the developed methodology to different systems.

An interesting application is related to the automatic analysis of technical specifications of other complex products in the transportation industry (e.g. cars, aeroplanes, ships) but also in other industries (e.g. electronics, white goods, retail), in order to extract properties, functions, normative references, etc. In these applications, the methodology to be followed is the same, except for some changes to the sources for the Knowledge Base construction, if necessary.

The methodology could be applied also in order to simplify the process of Request for Quotation (RFQ) analysis, too. In fact, in order to present quotations to customers on the basis of their solution and needs defined in an RFQ, the company has to extract features, requirements, physical characteristics, etc. and to compare them with features, requirements, physical characteristics, of products or services already offered and priced. In this case, the Knowledge Base should be built following the same approach. Anyway, in this case, the tool should perform not only the task of extraction and annotation, but also a comparison between the information extracted from the RFQ and those extracted from the specifications of the company's products. The aim is to return an ordered list of products/services that have similar characteristics to those of the solution required in the RFQ that could help the quotation manager in establishing the quotation.

Finally, the same methodology could be useful for identifying and classifying requirements implicitly contained within customer comments or reviews, in order to use them for improving existing products (modifying the specifications) or for creating completely new ones (drafting a new specification). A potential source of information can be social media, that has been recently used in order to retrieve technical related products information using NLP tools (Chiarello et al., 2020). In this case, the language used by users is mostly not technical, so the application would require a strong modification of the Knowledge Base.

## **Acknowledgements**

This work was supported by AnsaldoBreda S.p.A. (now Hitachi Rail Europe) and Erre Quadro s.r.l.. They are kindly acknowledged for their financial support. Moreover, the authors are in debt to Mr. Luca Conti who took over project coordination for AnsaldoBreda S.p.A. (now Hitachi Rail Europe), Ms. Caterina Pannocchia who gathered the information and technical documents and who manually tagged and revised the automatically gathered data by the application. The authors also acknowledge the Qulnn s.c.a.r.l and in particular Mr. Giacomo Petrini and Matteo Carmassi.

## References

- Abualhaija, S., Arora, C., Sabetzadeh, M., Briand, L. C., & Vaz, E. (2019). A Machine Learning-Based Approach for Demarcating Requirements in Textual Specifications. 2019 IEEE 27th International Requirements Engineering Conference (RE).
- Akhar, D. Kaur, I. (2020). Artificial intelligence, machine learning and deep learning: definitions and differences. *Clinical and Experimental Dermatology*, 45 (1), pp. 131-132.
- Al Kilani, N. A., Tailakh, R., & Hanani, A. (2019). Automatic Classification of Apps Reviews for Requirement Engineering: Exploring the Customers Need from Healthcare Applications. 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS).
- Al-Sahaf, H., Bi, Y., Chen, Q., Lensen, A., Mei, Y., Sun, Y., Tran, B., Xue, B., Zhang, M. (2019). A survey on evolutionary machine learning. *Journal of the Royal Society of New Zealand*, 49(2), 205-228.
- AlGeddawy T, ElMaraghy H (2011). Product variety management in design and manufacturing: challenges and strategies. *Proceeding of the 4th International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV2011)*, Montreal, Canada.
- AlGeddawy, T., ElMaraghy H. (2013). Optimum granularity level of modular product design architecture, *CIRP Annals - Manufacturing Technology* 62 (2013) pp. 151–154.
- Bakar, N. H., Kasirun, Z. M., Salleh, N., & Jalab, H. A. (2016). Extracting features from online software reviews to aid requirements reuse. *Applied Soft Computing*, 49, 1297-1315.
- Baker, G., Gibbons R., Murphy K.J. (1994). Subjective performance measures in optimal incentive contracts. *The Quarterly Journal of Economics* 109(4), 1125–1156.
- Baldwin C.Y., Clark K. (2000). *Design rules: The power of modularity*. Cambridge, Mass., The MIT Press.
- Bonin F., Dell'Orletta F., Venturi G. and Montemagni S. (2010), Contrastive filtering of domain-specific multi-word terms from different types of corpora. In *Proceedings of the workshop "Multiword Expressions: from Theory to Applications"* (MWE 2010), 23rd International Conference on Computational Linguistics (COLING2010), Beijing, China, August 28, pp. 76-79.
- Bonin F., Dell'Orletta F., Venturi G. and Montemagni S. (2010), Singling out Legal Knowledge from World Knowledge. In *Proceedings of the IV Workshop on "Legal Ontologies and Artificial Intelligence Techniques"* (LOAIT'10), Fiesole, 7th July 2010.
- Bonin F., Dell'Orletta, F., Venturi, G., and Montemagni, S. (2010), A Contrastive in Approach to Multi-word Term Extraction from Domain Corpora, in *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, La Valletta, Malta, 19-21 May, pp. 3222-3229.
- Chalkidis, I., Androutsopoulos, I., Michos, A. (2017). Extracting contract elements, *Proceedings of the 16th Edition of the International Conference on Artificial Intelligence and Law* 19–28.
- Chiarello, F., Trivelli, L., Bonaccorsi, A., & Fantoni, G. (2018). Extracting and mapping industry 4.0 technologies using wikipedia. *Computers in Industry*, 100, 244-257.
- Chiarello, F., Bonaccorsi, A., & Fantoni, G. (2020). Technical Sentiment Analysis. Measuring Advantages and Drawbacks of New Products Using Social Media. *Computers in Industry*, 123, 103299.
- Cid-López, A., Hornos, M. J., Carrasco, R. A., & Herrera-Viedma, E. (2016). Applying a linguistic multi-criteria decision-making model to the analysis of ICT suppliers' offers. *Expert Systems with Applications*, 57, 127-138.

- Dalpiaz, F., Schalk, I. V., & Lucassen, G. (2018). Pinpointing Ambiguity and Incompleteness in Requirements Engineering via Information Visualization and NLP. *Requirements Engineering: Foundation for Software Quality Lecture Notes in Computer Science*, 119-135.
- Damiano, E., Spinelli, R., Esposito, M., & Pietro, G. D. (2017). An Effective Corpus-Based Question Answering Pipeline for Italian. *Intelligent Interactive Multimedia Systems and Services 2017 Smart Innovation, Systems and Technologies*, 80-90.
- Dell'Orletta F. (2009). Ensemble system for Part-of-Speech tagging, In: *Evaluation of NLP and Speech Tools for Italian*, 2009 (Reggio Emilia, Italy, December 2009). *Evalita 2009*.
- Dell'Orletta, F., Lenci, A., Marchi, S., Montemagni, S. and Pirrelli V. (2006). Text-2-Knowledge: a Computational Linguistic Platform for the extraction of knowledge from texts. In *Proceedings of the SLI-2006 Conference*: 20-28, Vercelli.
- Deshpande, G. (2019). SReYantra: Automated Software Requirement Inter-Dependencies Elicitation, Analysis and Learning. *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*.
- Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
- Dimitri N., Piga G., Spagnolo G. (2006). *Handbook of Procurement*. Cambridge, Cambridge University Press.
- Do, Q. A., Bhowmik, T., & Bradshaw, G. L. (2020). Capturing creative requirements via requirements reuse: A machine learning-based approach. *Journal of Systems and Software*, 170, 110730.
- ElMaraghy, W., ElMaraghy, H., Tomiyama, T., Monostori, L. (2012). Complexity in engineering design and manufacturing, *CIRP Annals-Manufacturing Technology* 61(2012)793–814.
- Eppinger SD, Browning TR (2012). *Design structure matrix methods and applications*. Engineering Systems, vol. xii. MIT Press, Cambridge, MA 334.
- Fantoni G.; Apreda R.; Dell'Orletta F.; Monge M., (2013) Automatic extraction of Function-Behaviour-State information from patents, *Advanced Engineering Informatics*.
- Fareri, S., Fantoni, G., Chiarello, F., Coli, E., Binda, A. (2020). Estimating Industry 4.0 impact on job profiles and skills using text mining. *Computers in Industry*, 118, 103222.
- Fehr, E., Schmidt K. (2007). Contracts and fairness. Adding a stick to the carrot? The interaction of bonuses and fines. *The American Economic Review* 97(2), 177–181.
- Frantzi, K., and Ananiadou, S. (1999), The C-value / NC Value domain independent method for multi-word term extraction, in *Journal of Natural Language Processing*, 6(3), pp. 145-179.
- Gero J.S., (1990). Design Prototypes: A Knowledge Representation Schema for Design, *AI Magazine*, vol. 11, no. 4, pp. 26-36.
- Gero J.S., Kannengiesser U. (2002). *The Situated Function-Behaviour-Structure Framework*, Artificial Intelligence in Design, Kluwer, Dordrecht, pp 89–104.
- Ghavami, P. (2020). *Big data analytics methods: Analytics techniques in data mining, deep learning and natural language processing*. Boston: De Gruyter

Giovannetti E., Marchi S., Montemagni S., Bartolini R. (2007). Ontology-based Semantic Annotation of Product Catalogues. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, (RANLP-2007), 27-29 Settembre 2007. pp.235-239.

Hackett, M. Robinson, I., & Statham, G. (2007). *The Aqua Group Guide to Procurement, Tendering & Contract Administration*. Revised and updated (3rd ed.). Oxford: Blackwell Publishing.

Jeon, K., Lee, G., & Jeong, H. D. (2020). Classification of the Requirement Sentences of the US DOT Standard Specification Using Deep Learning Algorithms. *Lecture Notes in Civil Engineering Proceedings of the 18th International Conference on Computing in Civil and Building Engineering*, 89-97.

Jindal, R., Malhotra, R., & Jain, A. (2016). Automated classification of security requirements. *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*.

Khan, J. A., Liu, L., & Wen, L. (2020). Requirements knowledge acquisition from online user forums. *IET Software*, 14(3), 242-253.

Laffont, J., Tirole J. (1993). *Theory of incentives in procurement*. Cambridge, Mass., The MIT Press.

Lenci A., Montemagni S., Pirrelli V., Venturi G. (2009), Ontology learning from Italian legal texts. In Joost Breuker, Pompeu Casanovas, Michel C.A. Klein, Enrico Francesconi (eds.), *Law, Ontologies and the Semantic Web Channelling the Legal Information Flood*, *Frontiers in Artificial Intelligence and Applications*, Springer, Volume 188, ISBN 978-1-58603-942-4, pp. 75-94-

Li, Y., Guzman, E., Tsimoura, K., Schneider, F., & Bruegge, B. (2015). Automated Requirements Extraction for Scientific Software. *Procedia Computer Science*, 51, 582-591.

Lucassen, G., Robeer, M., Dalpiaz, F., Jan Martijn E. M. Van Der Werf, & Brinkkemper, S. (2017). Extracting conceptual models from user stories with Visual Narrator. *Requirements Engineering*, 22(3), 339-358.

MacLeod, W. B. (2007). Reputations, relationships and contract enforcement. *Journal of Economic Literature* 45, 595–628.

Mani, I., Bloedorn, E. (1997). Multi-document summarization by graph search and matching. *Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence (AAAI'97/IAAI'97)*. AAAI Press, 622–628.

Morales-Ramirez, I., Kifetew, F. M., & Perini, A. (2019). Speech-acts based analysis for requirements discovery from online discussions. *Information Systems*, 86, 94-112.

Munaiseche, C. P., Amel, J. P., Rantung, V. P., Rorimpandey, G. C., Sangkop, F. I., & Rompas, P. T. (2018). Expert System Implementation for the Diagnosis of Skin Diseases using Forward Chaining Method. *Proceedings of the 7th Engineering International Conference on Education, Concept and Application on Green Technology*.

Nečaský, M., Klímek, J., Mynarz, J., Knap, T., Svátek, V., & Stárka, J. (2014). Linked data support for filing public contracts. *Computers in Industry*, 65(5), 862-877.

Rajbabu, K., Srinivas, H., Sudha, S. (2018). Industrial information extraction through multi-phase classification using ontology for unstructured documents. *Computers in Industry*, 100, 137-147.

Robeer, M., Lucassen, G., Jan Martijn E. M. Van Der Werf, Dalpiaz, F., & Brinkkemper, S. (2016). Automated Extraction of Conceptual Models from User Stories via NLP. *2016 IEEE 24th International Requirements Engineering Conference (RE)*.

Rosadini, B., Ferrari, A., Gori, G., Fantechi, A., Gnesi, S., Trotta, I., & Bacherini, S. (2017). Using NLP to Detect Requirements Defects: An Industrial Experience in the Railway Domain. *Requirements Engineering: Foundation for Software Quality Lecture Notes in Computer Science*, 344-360.

Sabir, M., Chrysoulas, C., & Banissi, E. (2020). Multi-label Classifier to Deal with Misclassification in Non-functional Requirements. *Trends and Innovations in Information Systems and Technologies Advances in Intelligent Systems and Computing*, 486-493.

Sarmiento-Calisaya, E., Cárdenas, E. H., Cornejo-Aparicio, V., & Alzamora, G. S. (2020). Towards the improvement of natural language requirements descriptions. *Proceedings of the 35th Annual ACM Symposium on Applied Computing*.

Shah, U. S., Patel, S. J., & Jinwala, D. C. (2020). Constructing a Knowledge-Based Quality Attributes Relationship Matrix to Identify Conflicts in Non-Functional Requirements. *Journal of Computational and Theoretical Nanoscience*, 17(1), 122-129.

Sleimi, A., Ceci, M., Sannier, N., Sabetzadeh, M., Briand, L., & Dann, J. (2019). A Query System for Extracting Requirements-Related Information from Legal Texts. *2019 IEEE 27th International Requirements Engineering Conference (RE)*.

Smith, R.P., Eppinger S.D. (1997). A predictive model of sequential iteration in engineering design. *Management Science*, 43, 1104-1120.

Suh N.P. (2005). *Complexity Theory and Applications*, Oxford University Press, Inc.

Sulem, E., Abend, O., Rappoport, A. (2018). Simple and Effective Text Simplification Using Semantic and Neural Methods. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Sunkpho, J., Hofmann, M. (2020). Thai Words Segmentation Using an Unsupervised Learning Technique. *Recent Advances in Information and Communication Technology 2020 Advances in Intelligent Systems and Computing*, 79-92.

Thomke S., Bell D.E. (2001). Sequential testing in product development. *Management Science*, 47(2), 308-323.

Tiun, S., Mokhtar, U. A., Bakar, S. H., & Saad, S. (2020). Classification of functional and non-functional requirement in software requirement using Word2vec and fast Text. *Journal of Physics: Conference Series*, 1529, 042077.

Ulrich, K.T., Eppinger S. (2008). *Product design and development*. Boston, McGraw Hill.

Umeda Y., Ishii M., Yoshioka M., Shimomura Y., and Tomiyama T. (1996). Supporting conceptual design based on the function-behavior-state modeler. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 10, no. 4, pp. 275-288.

Umeda Y., Takeda H., Tomiyama T., and Yoshikawa H. (1990). Function, behaviour, and structure, in *Applications of artificial intelligence in engineering V*, *Proceedings of the Fifth International Conference*, Boston, USA, July 1990, vol. 1, J. S. Gero, Ed. Computational Mechanics Publications Southampton Boston Co-published with Springer-Verlag Berlin Heidelberg New York London Paris Tokyo, 1990, pp. 177-194.

Umeda, Y., Tomiyama, T. and Yoshikawa, H. (1995). FBS Modeling: Modeling scheme of function for conceptual design, *Proceedings of the 9th International Workshop on Qualitative Reasoning*. Amsterdam, Netherlands, 11-19 May 1995: 271-8.

Vintar, S (2004). Comparative Evaluation of C-value in the Treatment of Nested Terms, in Proceedings of "Memura 2004 - Methodologies and Evaluation of Multi-word Units in Real-World Applications", (LREC 2004 Workshop), pp. 54-57.

Watson, R. (2006). Part-of-speech tagging models for parsing. In: Proceedings of the 9th Annual CLUK Colloquium. Open University, Milton Keynes, UK.

Winkler, J., & Vogelsang, A. (2016). Automatic Classification of Requirements Based on Convolutional Neural Networks. 2016 IEEE 24th International Requirements Engineering Conference Workshops (REW).

Yoshida, K., Tsuruoka, Y., Miyao, Y., Tsujii, J. (2007). Ambiguous part-of-speech tagging for improving accuracy and domain portability of syntactic parsers. In: Proceedings of the 20th international joint conference on Artificial intelligence (IJCAI'07). Hyderabad, India.