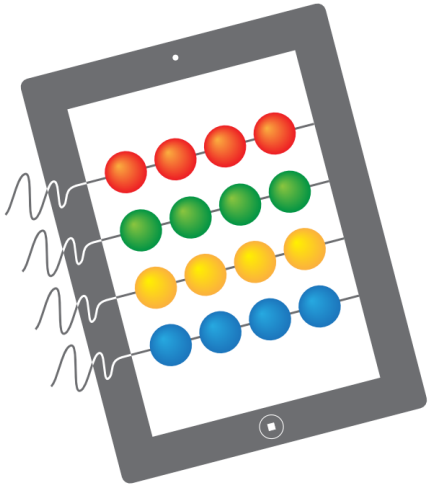
FP7 ICT STREP Project



LEARN PAD

# Deliverable D4.2

# Quality Assessment Strategies for Contents

http://www.learnpad.eu

LATEX template v. 0.5

CC BY

| | | |
|---|---|---|
| **Project Number** | : | FP7-619583 |
| **Project Title** | : | Learn PAd<br>Model-Based Social Learning for Public Administrations |

| | | |
|---|---|---|
| **Deliverable Number** | : | D4.2 |
| **Title of Deliverable** | : | Quality Assessment Strategies for Contents |
| **Nature of Deliverable** | : | Report |
| **Dissemination level** | : | Public |
| **Licence** | : | Creative Commons Attribution 3.0 License |
| **Version** | : | 4.0 |
| **Contractual Delivery Date** | : | October 31, 2015 |
| **Actual Delivery Date** | : | October 30, 2015 |
| **Contributing WP** | : | WP4 |
| **Editor(s)** | : | Alessio Ferrari (CNR), Hans Friedrich Witschel (FHNW), Giorgio O. Spagnolo (CNR), Stefania Gnesi (CNR) |
| **Author(s)** | : | Alessio Ferrari (CNR), Hans Friedrich Witschel (FHNW), Giorgio O. Spagnolo (CNR), Stefania Gnesi (CNR) |
| **Reviewer(s)** | : | Congyu Zhang (FHNW) |


# Abstract

In this deliverable we identify the quality assessment strategies for the natural language content associated to the Business Process Models (BP Models), within the Learn PAd project. The deliverable presents an in-depth domain analysis, including literature review, interviews with public administration (PA) stakeholders, and questionnaires submitted to PA stakeholders. Moreover, it defines a set of guidelines for editing natural language content in Learn PAd, and a quality model with associated rule-based and algorithmic strategies for computing the quality of such content. An experimental evaluation is presented concerning the potential usage of machine-learning techniques as a complementary tool for quality evaluation. The deliverable also introduces some technical details that pave the basis to successively create the content analysis component of the Learn PAd platform.



# Keyword List

BPMN, Business Process, Natural Language Content, Verification, Quality

CC BY

## Document History

| Version | Changes | Author(s) |
|---|---|---|
| 1.0 | First Draft - Table of Content. | Alessio Ferrari, Stefania Gnesi, Giorgio O. Spagnolo, Hans Friedrich Witschel |
| 1.1 | First draft including Chapter 2,3,4 and 5. | Alessio Ferrari |
| 1.2 | Second draft including Chapter 6. | Hans Friedrich Witschel |
| 1.3 | Final draft including Chapter 1,7,8. | Alessio Ferrari, Stefania Gnesi, Giorgio O. Spagnolo, Hans Friedrich Witschel |
| 2.0 | Internal release. | Alessio Ferrari, Stefania Gnesi, Giorgio O. Spagnolo, Hans Friedrich Witschel |
| 3.0 | Reviewer comments have been addressed. | Alessio Ferrari, Stefania Gnesi, Giorgio O. Spagnolo, Hans Friedrich Witschel |
| 4.0 | Final quality check of Candidate Release version. | Antonia Bertolino |

## Document Reviews

| Release | Date | Ver. | Reviewers | Comments |
|---|---|---|---|---|
| **ToC** | 1 Sept 2015 | 1.0 | | |
| **Draft** | | 1.3 | | |
| **Internal** | | 2.0 | Congyu Zhang (FHNW) | Details are available in the individual review report. |
| **Candidate Final** | 24 October 2015 | 3.0 | | |

CC BY

# Glossary, acronyms & abbreviations

| Item | Description |
|---|---|
| BP | Business Process |
| BPMN | Business Process Model and Notation |
| PA | Public Administration |
| NLP | Natural Language Processing |
| NL | Natural Language |


CC BY

CC BY

# Table Of Contents

CC BY

CC BY

# List Of Tables

CC BY

# List Of Figures

# 1 Introduction

## 1.1. Deliverable Purpose

The current deliverable is part of the **Models and Contents Quality Assessment** work-package (WP4) of Learn PAd. Among the other objectives, WP4 aims to analyse the natural language (NL) content that is associated to the business process (BP) models in the wiki pages of the Learn PAd platform. The goal is to evaluate the *quality* of such content, and provide appropriate feedback to the editor of the content, so that she/he can apply corrections, and, in turn, improve the content quality.

This deliverable presents the first results of WP4 related to **Task 4.2: Linguistic Quality Evaluation**, and **Task 4.3: Feedback-based Quality Evaluation**. The first task is concerned with the evaluation of the linguistic quality of the Learn PAd NL content by means of rule-based or algorithmic approaches. The second task is oriented to evaluate the linguistic quality of the NL content by means of machine-learning techniques.

Both tasks have been prepared with an in-depth **domain analysis**, oriented to understand which are the typical quality defects of current NL documents in the Public Administration (PA), and, in particular, the defects of procedure descriptions. Indeed, the Learn PAd wiki pages will be oriented to describe procedures in the PA, and we expect them to be similar in terms of language and content to the current PA procedure descriptions. On the other hand, we do not want wiki pages to exhibit the quality defects that occur in current PA documents. Hence, we have first performed a literature review on recommendations on writing styles for PA documents, and performed a set of interviews and questionnaire with civil servants, to understand which are the typical defects of PA documents. From this domain analysis, we have defined a set of guidelines to be used by the contributors of the Learn PAd content. Moreover, we have defined a quality model as reference for developing a set of rule-based and algorithmic strategies to identify linguistic defects of the Learn PAd content.

**Task 4.2** is focused on the definition of such strategies. The strategies have been defined according to six main quality attributes, namely simplicity, non-ambiguity, content clarity, presentation clarity, completeness and correctness. With respect to the initial work plan, we did not consider the consistency quality attribute, oriented to establish the degree of consistency between BP models and NL content in wiki pages. Indeed, according to Deliverable D5.1, Sect. 5.2, the wiki pages are generated directly from BP models, and, hence, consistency is ensured by construction. We have defined 16 measurable indicators in total for the different quality attributes. Moreover, we have performed a preliminary implementation and evaluation of the rule-based and algorithmic strategies to measure such indicators. The strategies defined within this task will be part of the Content Analysis component of the Learn PAd platform.

During **Task 4.3**, part of the linguistic defects identified during the domain analysis have been experimentally evaluated with machine-learning approaches. To this end, a data-set was defined using 23 real-world PA procedure descriptions, since, at the stage of preparing this deliverable, the Learn PAd content is not available yet. We have experimented with Naive Bayes and Decision Trees, and we have seen that, though results are encouraging, research is still needed to include machine learning techniques in the automated quality evaluation mechanism of Learn PAd.

## 1.2. Related Deliverables

The deliverable has been organized according to the first results of the Learn PAd project, according to the following deliverables.

- **D1.1. Requirements Report**. The deliverable identifies Learn PAd as a socio-technical ecosystem that is based on fundamentals of process-oriented learning and consists of a set of software components, the so-called Learn PAd platform.
- **D2.1 Platform Architectural Description**. The deliverable describes how the different components of the Learn PAd platform, including the Content Analysis component, interact.
- **D4.1 Formal Verification of Business Processes**. The deliverable lists the quality assessment strategies for BP models.
- **D5.1 Models for Setting the Wiki**. The deliverable outlines the planned interactions among BP models, wiki pages and Learn PAd users.

## 1.3. Deliverable Structure

The deliverable is organized as follows.

- Chapter 2 introduces Learn PAd roles and quality assessment strategies for natural language content, together with an outline of the research approach followed.
- Chapter 3 describes the domain analysis performed in the context of quality of natural language content. The domain analysis is composed of a literature review, a set of interviews with civil servants, a a questionnaire distributed to civil servants.
- Chapter 4 describes a set of guidelines that we have derived for editing natural language content in the context of Learn PAd. Moreover, the chapter describes a quality model that we have developed to define measurable indicators of quality defects.
- Chapter 5 describes the automated quality assessment strategies included in Learn PAd, according to the quality model defined in Chapter 4.
- Chapter 6 describes the experiments performed to evaluate to which extent machine learning can be applied in Learn PAd to automatically check the quality of Wiki pages.
- Chapter 7 introduces the software quality assessment mechanisms included in the Learn PAd platform.
- Chapter 8 reports some conclusions and future development.

# 2 Quality Assessment Strategy for NL Contents

This chapter describes the process envisioned for the quality assessment of natural language content – referred in the following as NL content or simply, content. Overall, the process can be partitioned into two complementary quality assessment strategies: an **automated** quality assessment strategy, and a **crowd-based** quality assessment strategy. The former, more software-intensive, employs automatic assessment strategies – as described in Chapter 5. The latter, more human-intensive, employs the feedback of the learners to improve the quality of the NL content, and, in the long term, to provide additional guidelines to plug in the Learn PAd platform. Moreover, the chapter also outlines the research approach followed, which can be regarded as a reference map for the rest of the deliverable.

## 2.1. Pages Structure

To understand the two strategies it is first useful to outline the Wiki structure foreseen for Learn PAd. Before looking at the picture in Fig. 2.1, let us first give some history of this structure, to understand its rationale. As described in Deliverable 5.1, Sect. 5.2., in Learn PAd each BP model that describes a procedure is associated with a set of Wiki pages, one for the overall model and one for each component (e.g., task, gateway, *etc.*) of the model. These Wiki pages are automatically generated from the structure of the BP model. This direct mapping between components and pages ensures *consistency* between the BP models and the associated content. However, to ensure consistency, the content of such Wiki pages should not be editable outside of the modelling platform. Unfortunately, this approach does not allow learners to contribute to the learning content with their knowledge. Therefore, in agreement with all the partners of Learn PAd, we have decided to introduce an additional Wiki page, which will include collaborative content.

Fig. 2.1 gives an overview of the structure that we have agreed. For each model, and for each component we will have *two* pages: a **Static Wiki Page** and a **Collaborative Wiki Page**. The former will include static content, which provides a *general* description of the associated model or component, and can be edited solely through the modelling platform. The latter, which will be accessible through a *link* from the static page, will include collaborative content. This can be edited solely through the Learn PAd platform and will include *details* about the model or component described. As explained, this split into two pages is guided by the need to ensure consistency between models and Wiki pages, and, at the same time, allow Wiki pages to be extended with more detailed content coming from the learners. Examples of Static and Collaborative pages are provided at the top of Fig. 2.4. We will later refer to such figure to discuss an example of automated quality evaluation.

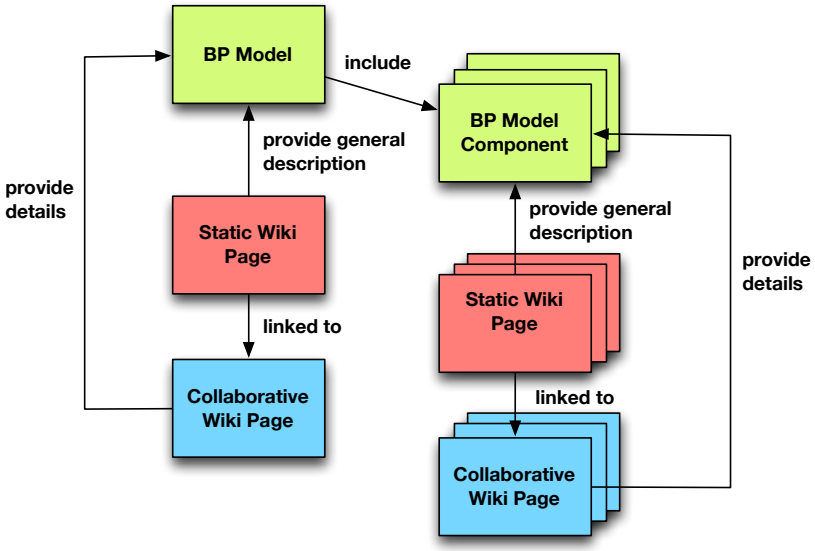



**Figure 2.1: Structure of the Wiki with respect to the BP models.**

## 2.2. Roles and Tasks

The two envisioned strategies for quality assessment jointly operate to improve the quality of the content. It is therefore useful to present them as part of a single quality assessment process. Let us first consider the roles involved within the quality assessment process, and the expected tasks of each role, by looking at Fig. 2.2. Circles represent roles, while boxes represent tasks to perform. An arrow connects a role to a task, in case the role is supposed to perform such task.

Let us give a brief overview of roles and tasks. The **Content Manager** first edits the static content within the modelling platform. Such content will be part of the Static Wiki Pages. After the Wiki pages are generated, she/he provides some initial details in the Collaborative Wiki Page. Then she/he validates the linguistic quality of the Static and Collaborative pages. The **Learners** can provide feedback on the descriptions reported in the Wiki pages, and can edit the Collaborative page. Meanwhile, the **Content Manager** monitors such feedback, and she/he provides modification to both Static and Collaborative pages following the feedback. Moreover, the **Content Manager** monitors the content added by the Learners in the Collaborative page and she/he re-validates such content.

When the Content Manager sees that some common linguistic defects could be addressed through automated quality assessment, she/he contacts the **Guidelines Manager** who will take care of updating the Learn PAd platform with novel guidelines. In the following section, we give the details of each role and each task, taking Fig. 2.2 as reference.

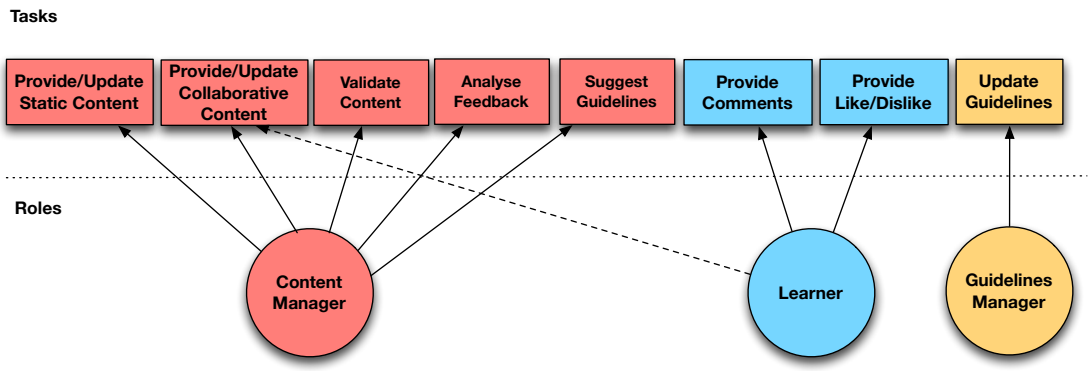



**Figure 2.2: Roles and tasks associated to the quality assessment strategy for NL Content.**

The roles involved in the quality assessment process, together with their tasks, are the following.

- **Content Manager.** This role is played by a person who is expert in the specific process described by the BP model. A Content Manager is associated to one or more BP models of an organization. In principle, the role of Content Manager can be played by the same person who covers the role of Modeler (i.e., the role who edits the models, see Deliverable 4.1). The Content Manager takes care of the *maintenance* of the learning content, which includes both the BP Models, and the Natural Language (NL) Content associated to the models, and is included in the Wiki pages. Since the quality of the BP Models is the topic of Deliverable 4.1, we will not stress the role of the Content Manager on the quality of the models, but we will focus solely on his tasks in the quality assessment of NL Content. In this sense, the tasks of this role are the following.

  1) **Provide/Update Static Content:** the Content Manager provides a first general description of the BP model, and of its components, in the Static Wiki Page. The Content Manager can also update the Static Wiki Page, whenever the feedback coming from the Learners suggests him/her that the page needs improvement.
  2) **Provide/Update Collaborative Content:** after Wiki pages are generated and loaded in the Learn PAd platform, the Content Manager provides a detailed description of the BP model, and its components, in the Collaborative Wiki Page. The Content Manager can also update the Collaborative Wiki Page, whenever the feedback coming from the Learners suggests him/her that such page needs improvement.
  3) **Validate Content:** when all the Wiki pages are loaded and edited, the Content Manager can validate them. A button will be provided in each Wiki page. If the linguistic content of the page is found to be defective, the Content Manager provides appropriate corrections, until the content is considered valid.
  4) **Analyze Feedback:** the Content Manager monitors the comments provided by the Learners for the Wiki pages of the models she/he is in charge of. By reading such comments, she/he is able to evaluate the required improvements on the content. Moreover, by monitoring the number of Like/Dislike on a specific Wiki page, she/he can understand which are the Wiki pages that require major improvements according to the community, and she/he can prioritize updates on such pages.
  5) **Suggest Guidelines:** after monitoring a set of Wiki pages, the Content Manager understands that *specific* guidelines can be provided to address some common negative feedback coming from the Learners. For example, if many Learners encounter difficulties in understanding a specific jargon adopted in the descriptions, the Content Manager will recommend to avoid such jargon. The recommendations given by the Content Manager will be collected by the Guidelines Manager. After working with the platform, the Content Manager can also suggest to tune the parameters used by the current quality assessment strategies for Wiki pages. For example, she/he can suggest to increase the maximum length admitted for sentences, if she/he finds the constraints of the platform too restrictive for the context of the organisation. This opportunity of tuning is also the reason why we have decided to allow only the Content Manager to validate the content: by being the only person performing validation, she/he can have a clear view of what can be improved in the quality assessment.

- **Learner.** This role is played by a civil servant of the organization for which the NL Content has been written. Like the Content Manager, the Learners are allowed to perform the task named **Provide/Update Collaborative Content**. However, all the contributions will be validated by the Content Manager, to ensure the validity and the coherence of such contributions. The additional tasks of the Learner, in the context of NL content quality assessment, are reported below.

  1) **Provide Comments:** as described in Deliverable D5.1, Learn PAd gives Learners the possibility to provide comments to improve the NL Content. Such comments might be recom-

mended corrections/change requests on the content, suggested according to the Learner's daily experience and practice. Moreover, such comments might be requests for clarification of the content, in case the learner does not understand the descriptions provided (e.g., caused by the usage of technical language).

2) **Provide Like/Dislike:** by means of Like/Dislike buttons, as typical of social networks, the Learner can provide an easy feedback on the quality of the content.

In the following, we will refer to Comments and Like/Dislike with the term Feedback.

- **Guidelines Manager:** this role is covered by a person who is in charge of maintaining the Learn PAd platform. The Guidelines Manager is associated to multiple Content Managers, possibly belonging to different organizations, who will refer to him as the collector of guidelines recommendations. The main task of this role is following reported.

  1) **Update Guidelines:** after receiving guidelines recommendations from the Content Managers, she/he will decide the guidelines that to plug in the Learn PAd platform for providing automated quality assessment, or additional modifications that can help improving the usability of the platform taking into account the opinion of the users.

## 2.3. Quality Evaluation Process

Let us now put all the roles and the tasks together to see how the quality evaluation process operates. To this end, we will refer to Fig. 2.3. In this figure, the tasks have incoming and out-coming arrows. An arrow goes from a role to a task when such role is expected to perform such task. An arrow goes from a task to a role when the product of the task is used by the role.

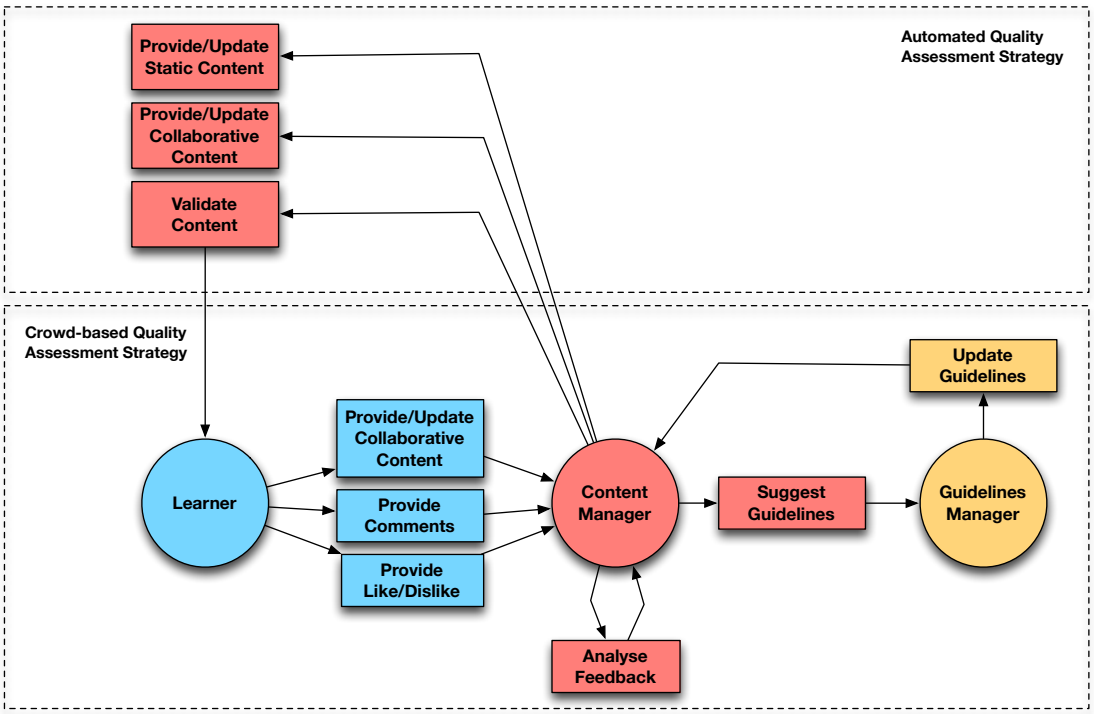



**Figure 2.3: The two quality assessment strategies for NL Content depicted as components of an overall quality assessment process.**

### 2.3.1. Automated Quality Assessment Strategy

The Automated Quality Assessment Strategy involves only one role, namely the Content Manager, whom we will refer to as the "user" in the following. This role edits the NL Content of both the Static and Collaborative pages and then she/he asks the platform to automatically validate the produced content (Validate Content). In case the system raises warning concerning the quality of the content, she/he will update the content according to the recommendations on possible improvements provided by the platform. she/he will iterate the validation and update the content, until the content is free of warnings, or the warnings provided by the system are considered negligible. The same process is performed anytime a Learner provides some textual contribution to the Collaborative page. The details of the strategies that are implemented in Learn PAd for automatically assessing the content of Wiki pages are given in Chapter 5.

To have a practical view of how the crowd-based strategy will work, it is useful to refer to the mock-ups of Fig. 2.4, which extends the case already presented in Deliverable 4.1. In the picture, we see a sample Wiki page – the **Static Wiki Page** – that shows a process for getting the reimbursement for expenses in a generic organization. The page has a link to the **Collaborative Wiki Page**, where more details are provided, following the template that we will outline in Sect. 5.6 – some of the fields are omitted for the sake of visibility of the picture. A button "VALIDATE" is placed at the bottom of the page. The same button will occur in the Static page, as well as in all the pages associated to the single tasks, and components of the process. The user presses the button "VALIDATE", and the system processes the text in the page according to the quality assessment strategies described in Chapter 5. Then, the user is re-directed to a **Quality Evaluation Page**, which for each of the quality attributes that we have defined (see Sect. 4.2), shows a human understandable quality evaluation score (e.g., GOOD, BAD, *etc.*), and a simple recommendation. For each attribute, the Content Manager can press the button "INSPECT", which will re-direct him/her to the **Inspection Page**. This is a non-editable page, where each defect found in the text is underlined (e.g., the vague expressions "proper" and "as soon as possible"). When the user moves the mouse over the defect, the user will see a tool-tip where explanations are provided for the defect. Then, the user can press the button "MODIFY", to update the page, and can re-execute the validation process.

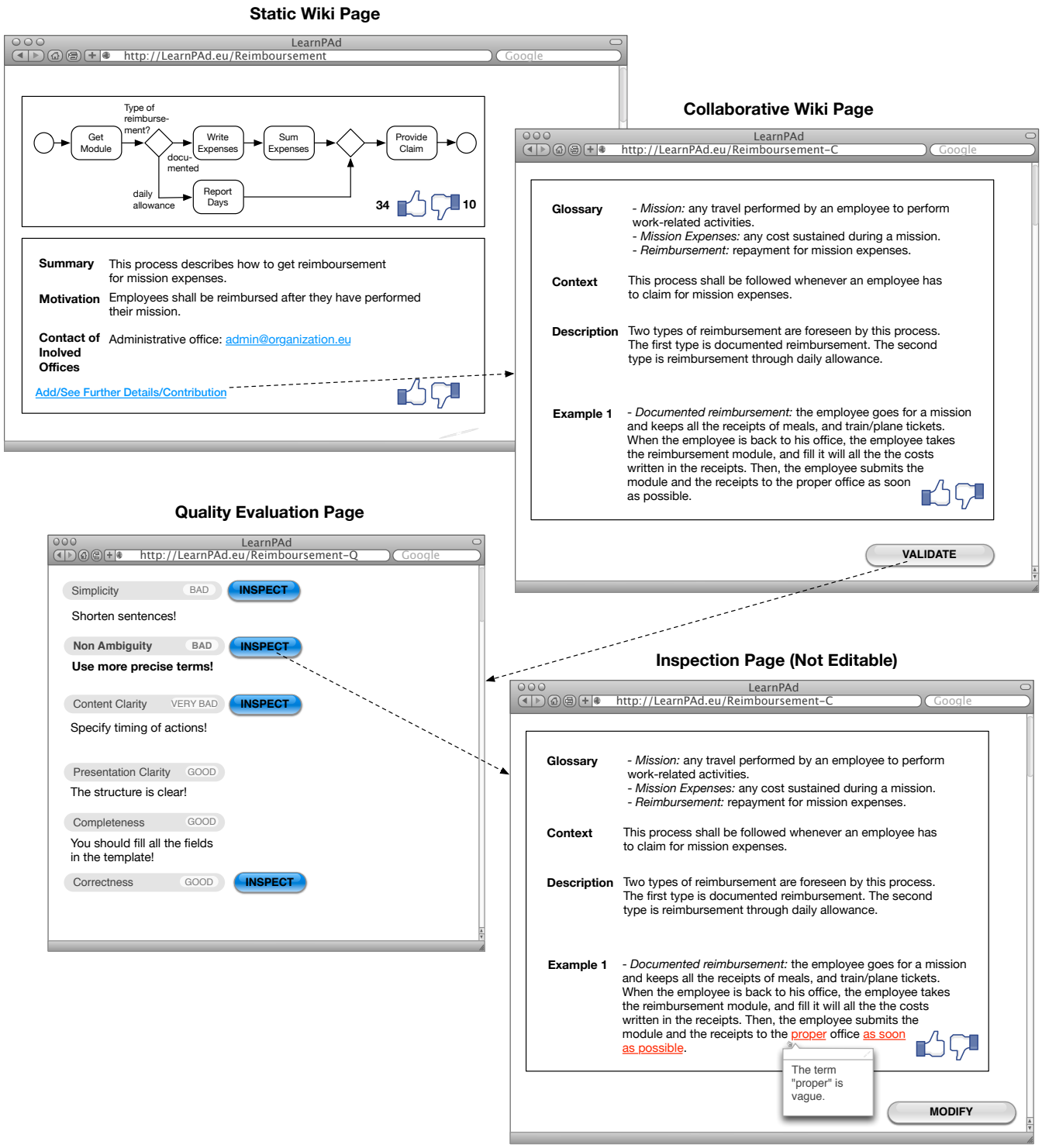



**Figure 2.4: Example of automated quality assessment of the NL Content.**

### 2.3.2. Crowd-based Quality Assessment Strategy

The Crowd-based Quality Assessment Strategy involves three roles, namely the Learner, the Content Manager, and the Guidelines Manager. The Learner provides feedback on the content, by means of comments (Provide Comments) and like/dislike (Provide Like/Dislike) buttons. The Content Manager will monitor the contributions of the Learners, and will evaluate all these feedback (Analyze Feedback) to understand and prioritize the required modifications on the content. Then, she/he will perform such modifications, and will repeat the Automated Quality Assessment Strategy, as described in Sec. 2.3.1.

In the long term, the Content Manager will be able to identify typical weaknesses of the content for which she/he is in charge, according to the feedback of the users. To address these common weaknesses, she/he will recommend content validation guidelines to plug into the Learn PAd platform (Suggest Guidelines). The Guidelines Manager will collect guidelines recommendations from multiple Content Managers, and will define techniques to automatically assess such guidelines (Update Guidelines). These iterations will enable a refinement of the Automated Quality Assessment Strategy. Moreover, as previously explained, the Content Manager will also be able to recommend appropriate *tuning* of the parameters employed by the already available quality assessment strategies.

To have a practical view of how the crowd-based strategy will work, it is useful to refer to the mock-ups of the following figures, which extends the case already presented in the previous section. As shown in the "Description" field of the Collaborative page, two types of reimbursement are foreseen, namely documented reimbursement and daily allowance reimbursement. Both types require to get a reimbursement module, which is not referred in the description. One of the Learners provides a comment, asking about such module. The Content Manager replies to the comment, to help the Learner, and will update the page with an appropriate link to the module. Note that, in the Static Wiki page, we have a link named "Add/See Further Details/Contribution". The link brings to the Collaborative Wiki page, where further details are given (an excerpt is provided in the figure), and can be further added by Learners and by the Content Manager.

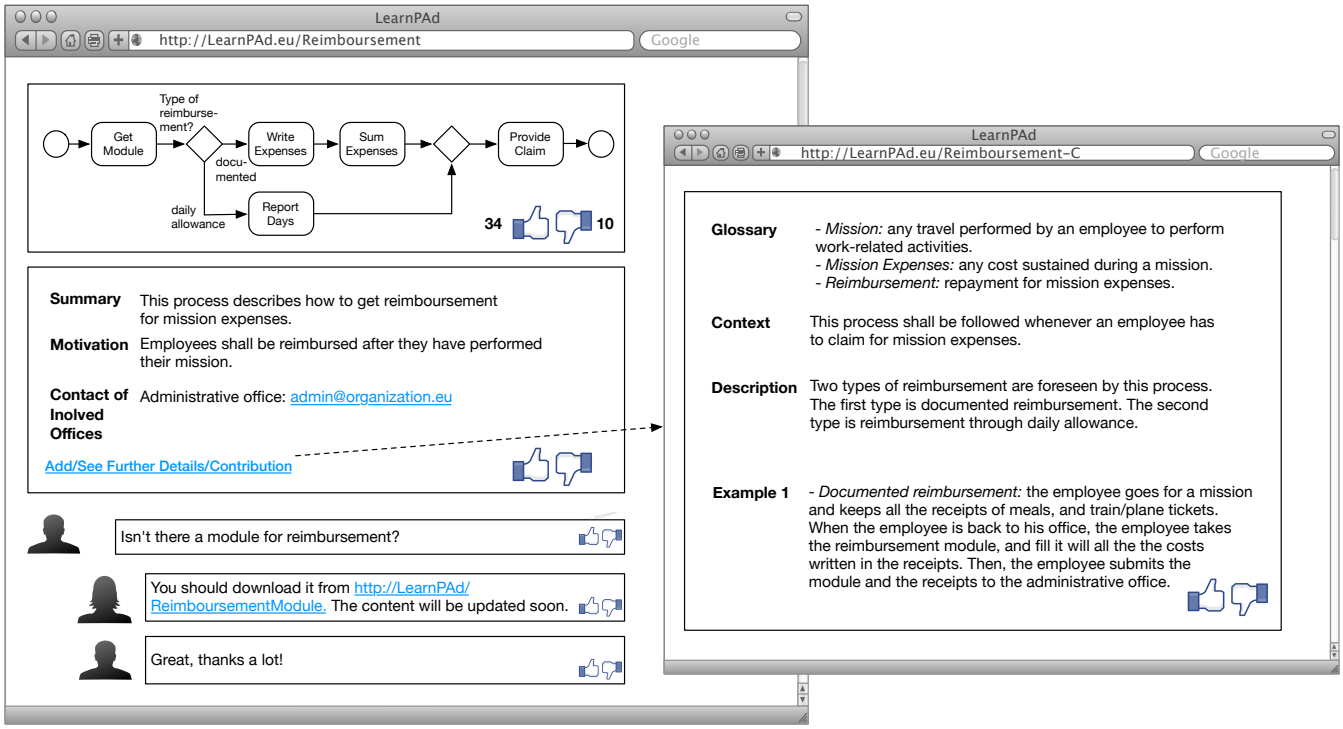



**Figure 2.5: Example of crowd-based quality assessment of the NL Content. The Learner highlights a missing document, and the Content Manager provides her/him an early feedback. Then, she/he will update the content of the page.**

The Content Manager updates the content as in Fig. 2.6 (for simplicity, in the figure we do not report the model). Note that the Content Manager has updated the Collaborative page, and not the Static page. This choice is driven by the fact that changes in the Static page shall be minimised. Indeed, the content of such page can be modified only within the modelling environment. This would imply re-generating the pages and re-loading them in Learn PAd, which would require longer time.

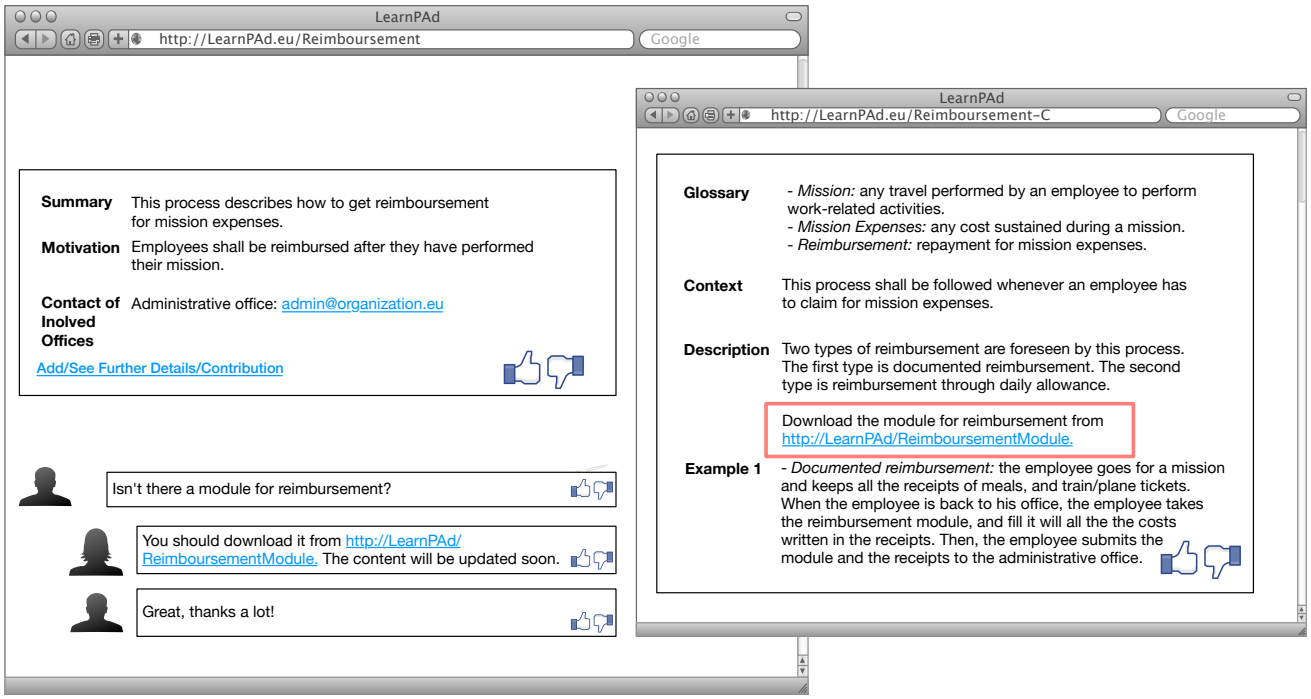



**Figure 2.6: Example of crowd-based quality assessment of the models. The Content Manager updates the content to address the need of the Learner.**

In the long term, the Content Manager understands that, probably, a field called “Modules” might be actually needed in the Static page, so that modules for procedures are available when needed, and the editor does not forget to add such modules. Therefore, she/he contacts the Guidelines Manager to suggest to add a field “Modules” to the template of the Static page. The Guidelines Manager, who takes care of the maintenance of Learn PAd, will add such field to the Static page[1].

It is worth highlighting that the crowd-based quality assessment strategy described in this section is highly human-intensive, and relies on the capabilities of the Learn PAd collaborative environment, as specified in Deliverable D5.1. In a sense, this section is a guide for Learn PAd users on *how* the crowd-based quality assessment can be put into place, once the Learn PAd platform is deployed in a specific administration.

[1] As described in Sect. 5.6, the field is currently part of the “Input Documents” field foreseen for the Static Page

## 2.4. Research Approach

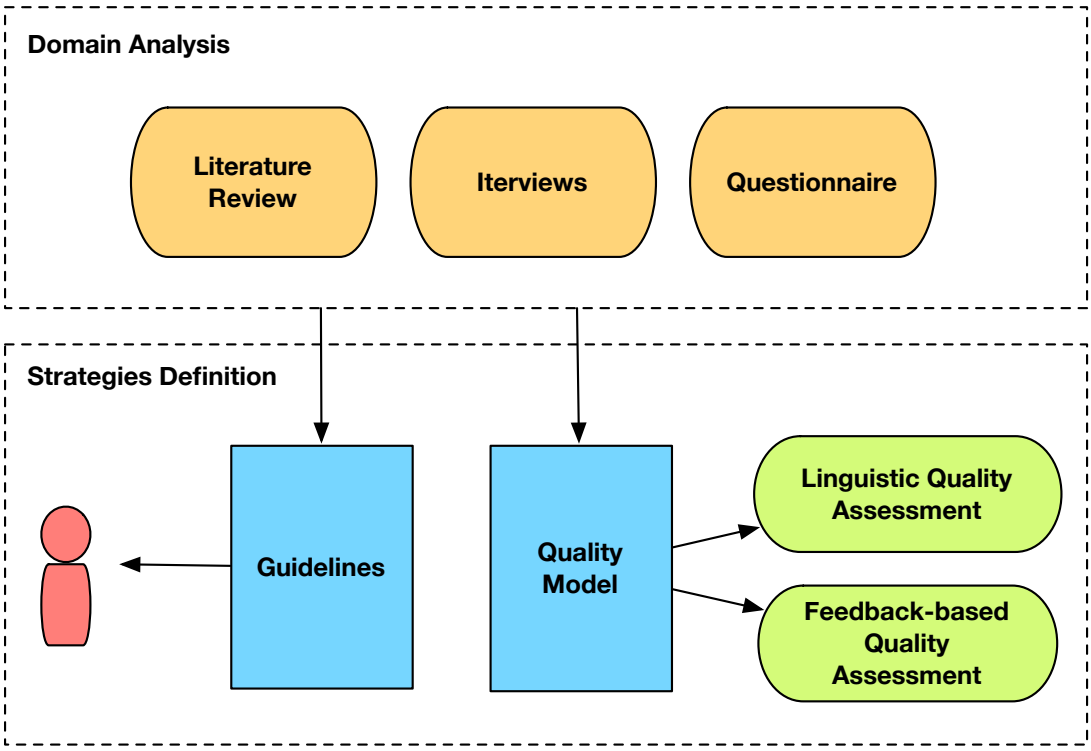



**Figure 2.7: The research approach followed for the definition of quality assessment strategies for NL content.**

In this section, we outline the research approach that we have followed to define our strategies for the quality assessment of NL content included in the Wiki pages of Learn PAd. This description will also work as an outline to understand the logical structure of the current deliverable. Fig. 2.7 depicts the research approach followed. Our research started with a **Domain Analysis** (Sect. 3) based on a **Literature Review** on quality of NL documents in general, and on quality of PA documents in particular. To complement this research with some direct feedback from people from the PA, we have also performed a set of **Interviews** with civil servants of the PA, oriented to understand which were the main problems with the quality of their current procedure descriptions. After the interviews, we have prepared a **Questionnaire** for PA civil servants, to understand the relevance of the different quality defects that they encounter in documents describing procedures that they have to apply in their daily life. The idea was that Wiki documents in Learn PAd shall not exhibit the defects shown by the current PA procedure descriptions, and shall follow the guidelines available for editing clear documents.

Given such domain analysis, we have been able start working on our quality assessment strategies (Strategies Definition). We have used the result of the analysis to define a set of **Guidelines** for editing Wiki documents in the Learn PAd platform. Such guidelines do not necessarily define measurable indicators of quality, but are oriented to be easily read and applied by PA civil servants. Moreover, still based on the domain analysis performed, we have also defined a **Quality Model** for Wiki documents. Both the Guidelines and the Quality Model are described in Chapter 4.

The Quality Model has been the basis to define our strategies for quality assessment. The quality model is made by quality attributes (e.g., Clarity, Non-Ambiguity, *etc.*). Each attribute is associated to *measurable* indicators of potential defects. For each indicator, we have identified strategies to be implemented as part of the quality assessment component of the Learn PAd platform (**Linguistic Quality Assessment**). All such strategies are reported in Chapter 5. Moreover, we have selected a sub-set of defects to be checked with machine learning approaches (**Feedback-based Quality Assessment**).

The idea was to understand to which extent machine learning could be practically used within Learn PAd, and how the feedback of learners could be automatically used to improve the quality of Wiki descriptions. Research on this topic is reported in Chapter 6 .


CC BY

# 3 Domain Analysis

Domain analysis is the process of understanding the characteristics of a specific domain, and it is paramount when defining requirements for a software system. In the Learn PAd context, the domain is the public administration (PA), and, in the context of this deliverable, we are interested in the procedures of the PA, and more precisely in the NL defects of the documents that describe the procedures of the PA. To scope this domain, we have performed three subsequent activities. First, a literature review was performed to understand the work previously done on NL defects in general and on NL defects of PA documents in particular (Sect. 3.1). This was a preliminary step to define and perform a set of interviews with PA civil servants, which gave us more insight to tune our strategies for NL defect identification (Sect. 3.2). Finally, the results of the interview drove the definition of a questionnaire submitted to PA civil servants (Sect. 3.3). It is worth noting that the utility of this domain analysis is not limited to the definition of the strategies for quality assessment of NL content in Learn PAd. Indeed, this domain analysis can be used by other researchers who will have to deal with operational problems associated to PA procedures, and with NL defects associated to the description of such procedures.

## 3.1. Literature Review

In this section we introduce a state-of-the-art, which represents the starting point for the definition of quality assessment strategies for Natural Language (NL) Content. We first give an overview of books, associations and tools that are oriented to improve the quality of NL text (Sect. 3.1.1). Then, we review a set of public guidelines for writing PA documents (Sect. 3.1.2), since the NL Content of XWiki pages in Learn PAd is mainly expected to describe procedures of the PA. Moreover, we outline the current research on two quality attributes that are desirable in textual documents, namely non-ambiguity (Sect. 3.1.3) and readability (Sect. 3.1.4), and we highlight whether and how existing techniques for detecting ambiguity and readability defects can be applied in the Learn PAd context. Finally, we outline the contribution of the current work with respect to previous research and tools that address the issue of document quality (Sect. 3.1.5).

### 3.1.1. Books, Associations and Tools

The quality of textual document is considered a paramount issue in several context. Education material, technical manuals, software requirements, contracts, journals and public administration documents are only a limited set of cases where the degree of quality of written text is crucial to achieve effective communication. In general, textual quality is associated with the concepts of clarity, conciseness and the absence of technical jargon, and several books have been edited with practical recommendations on how to write using the so called "*plain* English", a language that shall be easily understandable by the target audience. Among such books, it is worth citing *The Plain English Guide* by Martin Cutts [14]. Within a set of 25 guidelines on using easy words, reducing cross references, and planning before writing, the guide tells that the average sentence length should be 15-20 words, and sentences should not exceed 40 words. Another reference book is *Style: Toward Clarity and Grace* by Joseph M. Williams [69]. This book is not structured with a set of guidelines as [14], but is more an education textbook, with theoretical reflections and practical examples to be studied rather than to be consulted as a practical manual.

A peculiarity of the book is the analysis of content-related aspects such as emphasis and elegance, which are not covered in other texts. The dissemination of plain English is also the objective of the Plain English Foundation[1], an Australian organisation that provides training and tools that help improving the quality of written text, and of the Plain English Campaign [2], providing analogous services and certifying documents written in plain English. Similarly, the Plain Language Association International [3] aims to promote plain language in any language. Also some tools are available that helps checking the quality of text in terms of plain English. Among them, the *Emingway Editor*[4], which is free for access and download, marks with different colors sentences that are hard to read, terms with simpler alternatives, adverbs, passive voice expressions and other quality defects. Other tools such as the plugin for Microsoft Word 2010[5] released by the University of Surrey are more focused on readability checks (see Sect. 3.1.4 for an insight on readability). Concerning tools specific for Italian, it is worth mentioning PADocs [6], specifically oriented to the redaction of PA documents. However, the tool is focused on providing a common structural standard for PA documents, rather than checking their quality in terms of language. Moreover, this is a commercial tool distributed by Tecnodiritto s.r.l. [7], and, to our knowledge, no free tool exist that helps editing PA documents, neither for Italian, nor for English.

### 3.1.2. Quality Guidelines in Public Administrations

The listed books, associations and tools are not specifically targeted to public administrations (PA) – with the exception of PADocs –, but to the general quality of a text. However, a text adopted in a PA – a regulation, a procedure, or, as in the case of Learn PAd, a wiki document describing a procedure – needs to exhibit certain quality attributes that are *specific* for PAs. Therefore, several PAs in the world have defined a set of guidelines to be adopted by the civil servants while writing PA-related documents, such as regulations, procedures, press releases and even speeches. General guidelines from the European Union (EU) are available in the document *How to Write Clearly* [8], and its extension named *Clair's Clear Writing Tips*, and dated Nov, 2014 [9]. Besides providing recommendations on reasoning on the expected reader of the document, such guidelines tend to emphasize the need to use short and simple expressions (e.g., "evaluate" instead of "carry out an evaluation of", "as" instead of "in view of the fact that"), avoid jargon and passive voice, and limit acronyms. Interestingly, these guidelines highlight a typical issue of PA documents, namely copy-paste errors. Often, documents are written by recycling the content of other documents, and some errors often occur in the resulting documents because the recycled text is not properly adapted. A detailed *Style Guide* for the use of English, directed to editors and translators working for the European Commission (EC), is available at `http://goo.gl/O5xp10` (last update: 13 Aug, 2015). The guide specifies a set of linguistic conventions that cover punctuation, capitalization, abbreviations, numbers and other elements of style that an editor or translator is expected to adopt when writing/translating official EC documents.

Also the government of the United States (US), within the *Plain Language* initiative [10], provides a set of guidelines for writing PA texts directed to the public (*Federal Plain Language Guidelines* [11]). These guidelines are similar in spirit to those provided by the EU in *How to Write Clearly*, but are more detailed in term of content – especially concerning recommendations for structuring a document –, and each guideline includes a set of examples and useful references. Similarly, also the Canadian government

[1] `https://www.plainenglishfoundation.com`
[2] `http://www.plainenglish.co.uk`
[3] `http://www.plainlanguagenetwork.org`
[4] `http://www.hemingwayapp.com`
[5] `http://goo.gl/Wx1Tef`
[6] `http://www.padocs.it/demo.php`
[7] `http://www.tecnodiritto.it`
[8] `http://goo.gl/iz557W`
[9] `http://goo.gl/ylRjIT`
[10] `http://www.plainlanguage.gov`
[11] `http://goo.gl/kn1CWY`, last update March, 2011

provides a *Successful Communication Toolkit*[12], designed for Canadian's Government communicators and managers, with the purpose of assisting them in communicating policies, programs and services to citizens. The peculiarity of this guide is the presence of case studies, which report the successful application of the plain writing guidelines in real PA contexts. The government of United Kingdom (UK) also provides style guidelines in its website [13], with a list of concise and clear recommendations. The peculiarity here is the focus on writing PA documents that are published on the Web. Specific guidelines are also given concerning aspects that are relevant within the Learn PAd project, namely guidelines that specify how to organize a procedure description. Stemming from the work of Redish [57], these guidelines recommend to describe procedures through numbered lists. Moreover, they recommend to keep lists short (5 to 10 items) for unfamiliar items, and suggest that, if lists are longer, they should be grouped into shorter lists each with its own sub-heading. Interestingly, these guidelines also suggest to avoid juridical jargon even in legal documents directed to larger audience. These latter guidelines are based the empirical study of Trudeau [63], which shows that 80% of the people involved prefers plain English instead of legal jargon, and that the more specialist the knowledge of the reader, the higher the preference for plain English. It is worth highlighting that the guidelines recommended by the UK government are also *applied* in practice – a fact that is not straightforward, as we show later – as one can easily see by navigating the GOV.UK website [14].

Concerning Italian Government, a specific norm for the simplification of the PA language was published on May, 8th 2002 [56]. The norm also includes recommendations and examples. In line with such norm, Cortellazzo and Pellegrino [13] defined a more detailed, although quite general, set of 30 guidelines for editing PA documents [15]. Overall, these guidelines are not so different from those found in the style guides of other countries, with the exception of the language specific examples provided, and the suggestions concerning the usage of the subjunctive verbal form, to be avoided in favor of the easier present tense. The debate concerning the language of PA is still lively in Italy, and the AQAA association (Associazione per la Qualità degli atti amministrativi [16]) published in 2011 a guide for editing administrative acts (*Guida alla redazione degli atti amministrativi. Regole e suggerimenti* [38] [17]). Although the general guidelines are in line with [56] and [13], the guide appears as the most comprehensive reference for editing PA documents in Italian. A peculiarity of the guide is the room given to the structure of a public act, the relevance of motivation and preamble, and the constraints for citing norms and other acts. Nevertheless, as reported in the recent study of Libertini *et al.* [43], such guide is still not well known within the Italian PA. As a result many of today's PA documents are adapted from older PA documents, with style-related modifications that highly depend on the motivation and good will of the editors.

### 3.1.3. Research on Non-Ambiguity

Ambiguity of terms and sentences is a relevant quality defect in any document. In general, ambiguity occurs whenever the meaning intended by the information producer (i.e., the writer) differs from the meaning understood by the information consumer (i.e., the reader) [30]. Ambiguity of *terms* is an open problem in the computational linguistic community, and is traditionally associated to the so-called word-sense disambiguation (WSD) task [50, 36, 61]. Techniques for WSD aim at identifying the intended meaning of a polysemous term – i.e., a term with multiple meaning such as "bass" (which can indicate "bass guitar" or a kind of fish) –, depending on its linguistic context. Several approaches exist that address this problem, which use unsupervised [1, 68], supervised [42, 53] and knowledge-based approaches [4, 51]. However, such techniques are mainly aimed to support information retrieval and machine translation, and are not oriented to detect ambiguity as a quality defect of a text.

[12] http://goo.gl/vCg1zr, May, 2003
[13] https://goo.gl/5UIoHi
[14] https://www.gov.uk
[15] http://www.maldura.unipd.it/buro/
[16] http://www.aquaa.it
[17] http://goo.gl/1s1kfP

Ambiguity as a quality defect has been largely studied in the field of requirements engineering, a particular field of software engineering that study all the activities for eliciting, defining, maintaining and validating requirements. In software engineering, a requirement expresses a need or constraint to be satisfied by a system. Requirements are normally written in natural language before the system is developed. Such requirements need to be understood by different stakeholders involved in the development of the system – e.g., customers, developers, verification and validation team– and they should be as less ambiguous as possible, to avoid misunderstanding among the stakeholders, which normally have different background and skills. Therefore, several studies have been performed to categorise and detect ambiguities in NL requirements. Part of the works are focused on the identification of typical ambiguous terms and constructions [7, 6, 33, 71, 32]. Other works address the ambiguities by translating the requirements into formal languages or models [23, 10, 3, 41]. Finally, some works focus on the usage of natural language understanding methodologies [49, 40].

A seminal work on ambiguity in requirements is the one of Berry *et al.* [6], where ambiguities are partitioned into four classes: *lexical* (i.e., the terms used have several meanings), *syntactic* (i.e., the requirement sentence has more than one syntax tree, each one with a different meaning), *semantic* (i.e., the predicate logic expression equivalent to the sentence has more than one interpretation) and *pragmatic* (i.e., the meaning of the sentence depends on the *context* in which it is used). According to this study, Gnesi *et al.* developed QuARS [33], a tool that detects ambiguities according to keyword-based linguistic indicators. A similar approach is followed by ARM [71]. Both these works are mainly focused on detecting *lexical* ambiguities, which depend on vague, weak or subjective expressions (e.g., “as soon as possible”, “reasonably”), and are also frequent source of ambiguities in PA documents.

Other interesting tools, not solely focused on requirements disambiguation, are LOLITA [48] and Circe-Cico [3]. The first is a general purpose NL analysis framework that employ a large hierarchical semantic network as a knowledge base. Here, ambiguity detection is performed by integrating the text into the network to perform automatic interpretation. The tool has been employed also to generate object-oriented models. The second tool, Circe-Cico, allows the progressive transformation of NL requirements into formal models. Still oriented to formal model generation are the notable works of Kof (see, e.g., [41]). These tools solve issues associated to *lexical*, *syntactic* and *semantic* ambiguities. However, they require to use constrained natural languages, which are unlikely to be found both in requirements, and in PA documents.

Valuable work has also been performed on *syntactic* ambiguities, and in particular on *anaphoric* (e.g., [72]) and *coordination* ambiguities (e.g., [9]) in requirements. Anaphoric ambiguities are referred to the contextual relationships given by pronouns. Coordination ambiguities are referred to the relationships among phrases given by conjunctions such as “and” or “or”. These works, which use machine-learning techniques, highlight the distinction between innocuous and nocuous ambiguities. The former are those ambiguities for which different readers tend to have the same interpretation. The latter are ambiguities that lead to different interpretations in practice. Though these types of syntactic ambiguities are relevant also in PA documents, the referred techniques are specifically targeted to requirements, and their use for PA documents needs further experimentation.

Finally, *pragmatic* ambiguities are explicitly addressed in the work of Gleich *et al.* [32], where pragmatic ambiguities are always resolved at syntactic or semantic level. Still on pragmatic ambiguities, recent works of Ferrari *et al.* [29, 27], aim to detect ambiguities that depend on the knowledge of a reader in the domain of the requirements. To this end, they use graph-based representations to model potential backgrounds of different readers. Moreover, an algorithm has been developed that takes the concepts expressed in the requirements and searches for corresponding “concept paths” within each graph. The paths resulting from the traversal of each graph are compared and, if their overall similarity score is lower than a given threshold, the requirements sentence is considered ambiguous from the pragmatic point of view. These techniques can in principle be adapted also to identify ambiguities in PA documents. However, it has to be highlighted that they are at an experimental stage, and, according to our recent evaluation [29], strategies are required to increase the performance of these approaches, so that they can be plugged in a *fast* quality checker, as the one foreseen within Learn PAd.

### 3.1.4. Research on Readability

A large amount of research has been devoted to the quality of text in terms of its *readability*. According to the definition of Dale & Chall [15], the readability of a text involves (a) typographical aspects – in this case it is common to speak about *legibility* –, (b) degree of interest that the content raises on the reader, and (c) stylistic aspects. Research on automated approaches to assess text readability has been mainly focused on the stylistic dimension. Therefore, in the following, we will discuss the most relevant and recent works that specifically address the issue of *stylistic readability*, defined as the capability of a text to be easily read "in terms of vocabulary, sentence structure and other expressional elements by a certain group of readers" [15]. For simplicity, and compliance with the discussed literature, in the following we will refer to "stylistic readability" simply as "readability".

Early works on automatic readability assessment have been mainly focused on defining *formulas* that could associate a degree of readability to entire documents. In general, such readability formulas take into account raw textual features, such as the length of sentences and words, the number of syllables in each word, and the number of words in each sentence. They normally assume that words with more syllables, and sentences with more words are less readable than shorter words and sentences. Among such readability formulas, the most widely adopted are the Flesch-Kincaid Grade Level [39], the Gunning-Fog Score [34], the Coleman-Liau Index [11], the SMOG Index [47], and the Automated Readability Index [59]. Given a text, such formulas compute a number that represent a grade level. The grade level is based on the USA education system, and is equivalent to the number of years of education that are required to read the given text. A grade level around 10-12 is the reading level achieved after completion of high-school. Normally, when a text is directed to the general public, the grade level should not be higher than 8. Web-based implementations of such formulas are also largely available online (see, e.g., `https://readability-score.com`). Language-specific readability formulas, still based on raw textual features, have also been defined for Italian. Among them, it is worth mentioning the Flesch-Vacca formula [31] (adaptation of the Flesch-Kincaid Grade Level [39]), and the GulpEase index [44].

More recent studies on readability have stressed the limits of traditional readability formulas. Indeed, since readability formulas are based on raw textual features – i.e., they do not take into account the actual content and structure of sentences – they may fail in reliably assessing the degree of readability of a text. Let us think, for example, to public administration texts, where a large number of acronyms is often found. An acronym is normally a short term of one syllable. Such formulas will assume that the term is easy to read, and in turn, a text containing several acronyms will be considered easy to read. However, to be understood, an acronym may require background knowledge that the reader does not have, and this impairs the readability of the text. Similar issues occur also when considering sentences. Indeed, a longer sentence is not necessarily less readable than a shorter one that use more complex syntactic constructions. Given these observations, and given experimental studies that showed the limits of readability formulas (see, e.g., [60, 24]), the research community focused on improving readability evaluation, by considering also the degree of difficulty of the *vocabulary*, and the complexity of the *syntax* adopted in the documents.

The difficulty of the vocabulary is considered, for example, in the Dale-Chall formula [8], which measures the readability of a text by taking into account the percentage of words in the text not included in a list of $3,000$ words considered easy-to-read. Currently, a ranked set of the $5,000$ most common American English terms is available at `http://www.wordfrequency.info/free.asp`, while a list of the $7,000$ most common Italian terms has been edited by De Mauro [17] in his *Basic Italian Vocabulary*. The list of these terms is maintained at `http://www.sensocomune.it`. Measures specific for Italian, and inspired to the Dale-Chall formula, have been defined in recent works of Dell'Orletta *et al.* [18, 19], and are based on the Basic Italian Vocabulary.

The complexity of the syntax is taken into account in more recent works that use machine-learning approaches. Such works can be partitioned into two groups, namely works oriented to establish a readability class for a document [54, 2, 25, 52], and works oriented to associate a document to a

degree of ranking [37, 45, 62]. It is worth noting that, with the exception of Dell'Orletta *et al.* [19], all the cited works perform readability assessment at the document level. Therefore, such approaches can provide little guidance in our context, where the editor of the PA document, or XWiki content, needs to know which specific part of text needs improvement in terms of readability.

### 3.1.5. Contribution

With the current work, we aim to develop a quality checker for wiki documents that provide descriptions of PA procedures represented through BP Models. The quality checker is aimed to (1) identify quality defects at the level of sentences, words and presentation, (2) support the writer in understanding what part of the text is poor in terms of quality, and (3) suggest what can the writer do to improve the text.

In this chapter, we have outlined the currently available knowledge resources that are concerned with the quality of a text. For different reasons, such resources covers only partially the needs for the quality checker envisioned within Learn PAd. Indeed, books, associations and available tools aim to address the challenge of plain English and plain language in general, but are not specifically oriented to solve quality issues of PA procedures. Quality guidelines for PA are oriented to improve the quality of PA documents, but are currently not supported by any tool, and, moreover, no specific guidelines is given for *procedures* of the PA. Research on non-ambiguity and readability address only one aspect of the quality of a text, and, again, available techniques need to be tailored for PA procedures. Indeed, as noted in [19], the notion of readability is strictly genre-dependent. This implies that specific strategies have to be defined to check the readability of a text in the context of PA procedures. Similarly, as noted in [9] and [46], also non-ambiguity is a reader's dependent concept, and strategies have to be defined that consider the *context* of the reader of the procedures.

The contribution of this work, with respect to the cited literature, is therefore manifold. (a) We provide a quality model that is *specific* for descriptions of PA procedures described in the Learn PAd platform. The quality model outlines quality attributes that are desirable for procedure descriptions in Learn PAd, namely Simplicity (i.e., a concept of readability specific for PA documents), Non-Ambiguity, Content Clarity, Presentation Clarity, Correctness and Completeness. (b) We define and implement multiple strategies to automatically check the different quality attributes at the level of words, sentences and presentation. (c) We provide specific recommendations to the writer for improving the quality of the text. (d) Finally, an additional research contribution is the experimental evaluation of ML techniques for checking the quality of PA procedures (see Sect. 6). Our experiments show that conceptual work is still needed, before ML can be successfully applied to automatically check the quality of PA procedure descriptions.

## 3.2. Interviews

After the analysis of the literature, we have performed a set of four in-depth interviews with civil servants belonging to different PAs. The goal of these interviews was understanding the typical characteristics of the potential environments in which Learn PAd is going to be deployed, and identifying which are the categories of problems in NL procedures that the civil servants encounter in their daily life. Moreover, observations coming from such interviews have been used to define the questionnaire to be distributed to civil servants.

We have chosen different offices to have a complete view of potential environments and problems. In particular, we have performed two interviews with the administrative staffs of CNR-ISTI[18]. Each interview involved a two civil servants that were working together in the same office, so four people in total were interviewed. Moreover, we have performed one interview with an EU Project Officer, belonging to the the "2 Mears Seas Zeeën Programme"[19]. Finally, we have performed one interview

[18] http://www.isti.cnr.it
[19] http://www.interreg4a-2mers.eu/en/

involving two front-desk employees working at SUAP (Sportello Unico Attivit Produttive) offices of the Marche Region [20]. Interviews at CNR-ISTI have been performed as face-to-face meetings, while the others have been performed through Skype calls.

Each interview lasted two to three hours. To have a uniform view of the gathered information, the interviewer was always the same person, a software engineering researcher with previous experience in interviews. All the interviews were semi-structured [20]: a list of questions was preliminary prepared by the interviewer as a support to give a direction to the meeting; then, when the answers given by the civil servants were opening new interesting directions, new questions were raised to explore such directions. During the interviews the interviewer took notes, and, at the end, he edited a detailed report to highlight the insights acquired. Here, we provide a brief summary of each interview, and an overall discussion on the observations that we have made, which are useful for understanding the rationale that we have adopted when defining our questionnaire.

### 3.2.1. Interview 1 - CNR-ISTI Administrative Staff (Personnel Management)

**Participants** Two people of the administrative staff participated to the interview. They manage the whole process associated to personnel selection and management. One is an expert civil servant with several years of service at the institution, the other has less years of service, and, though autonomous, he has less experience than the other.

**Types of procedures** The procedures that they have to follow (e.g., recruiting personnel, organising commissions, *etc.*) are not documented in a specific document. More often, they are a combination of basic process blocks that they have to reconstruct, based on a series of circulars sent by the central authority of the institution. Sometimes, they prepare a list of steps that define the overall procedure, and they use these steps as notes. Moreover, exchanging these notes is also a current practice among civil servants.

**Operational problems** They tell that the most difficult aspect is not following the procedures - which are well known by elder personnel and can be easily taught - but: (a) editing the acts associated to the procedures, since they have to take care of all the consequences of the documents that they edit, and, in this sense, having a global view of the process of personnel selection and management is fundamental; (b) understanding how a novel circular or regulation impacts on the current procedures.

They tell that an operational manual is not really needed for their processes, since the overall restructuring/variations occur with a frequency of years. However, there might be circulars/regulations that indirectly create constraints to the current procedures, but the actual change is not stated anywhere and is left to the interpretation of the civil servant. In this sense, their work require critical sense and deep knowledge of the procedures.

They state that a novel civil servant have to work by sub-tasks, he cannot be associated to the management of a whole set of procedures. Normally, when reading a circular, the most relevant aspect to that impact the understanding is the background of the reader, since the content may impact several aspect of the current procedures, also belonging to other offices. In this sense, it is also considered important to read the circulars that impact other offices, since each office is linked to the others.

**Linguistic problems** One of the most relevant problems encountered by novel civil servants is understanding of the meaning of specific/legal terms that belong to the domain of a specific procedure. For example, the term “Prestare Servizio” (“to Serve”) implies having a contract as researcher or employee, and not an agreement for research fellows. Therefore, when using/not using such term in an official document might change the meaning of the document. They tell that knowing these technical terms is relevant, and a *glossary* associated to documents might be useful for novel civil servants.

[20] http://www.impresa.marche.it/SportelloUnicoAttivitaProduttiveSUAP.aspx

The other the main defects of the circulars - which are the main official documents that impact the work of this office - are:

- absence of an overview that explain the context of the circular
- use of terminology and concepts that belong to other offices/processes and that are not always easy to understand
- too many references to other regulation that are not essential, and therefore they can make the reader loose the specific focus
- absence of explicit motivations for changes to the procedure
- it is not evident what is really relevant in the text
- absence of examples
- absence of clarifications about what to do if something goes wrong
- too long and complex sentences
- ambiguity apparently used on purpose. Sometimes it appears that juridical jargon is used to create ambiguity and leave freedom of interpretation to the civil servants.

### 3.2.2. Interview 2 - CNR-ISTI Administrative Staff (Projects and Contracts)

**Participants** Two employees from the administrative staff of CNR-ISTI participated to the interview. They take care of contracts and projects. They are both expert employees.

**Types of procedures** Procedures of this office mainly involve the redaction of official documents. The official documents of CNR-ISTI are several, from circulars, to contracts, to agreement with companies, *etc*. However, regardless of the documents involved, the of this office procedures are not defined, except for special cases, and depend on internal and EU regulations. As for the other administrative staff, when a new circular that impact the office is received, the civil servants interpret the circular and adapt their current procedures.

**Operational problems** The main operational problems highlighted are as follows. First, the absence of clear procedures, since the procedures are defined through regulations and circulars that *change* procedures that are not clearly defined anywhere. Second, they do not know who is the person to contact to explain the content of a circular, if such content is not understandable. Third, there is not enough teaching, and there is no time to learn by yourself by reading the laws and existing documents.

**Linguistic problems** In general, the content of EU regulations that impact the work of this office is considered rather clear from the linguistic point of view. As in interview 1, the main linguistic problems occur with the circulars, which are the main documents that impact the procedures of this office. The problems are the following:

- absence of a uniform structure for the circulars
- cross references and overlapping with other laws and circulars. These two problems lead to the inability to understand what shall be done, because relations are chaotic.
- absence of categories for the circulars. Semantic tagging, or clear categorical titles shall be provided (e.g., Benefits, Contracts) to make easier for the receiver to understand if the circular impacts hishe/her work, and what is the topic of the circular.

- absence of a clear recipient of the circular (who is this circular for?)
- problems related to copy/paste of the content of other modules and circulars
- problems related to copy/paste of e-mail addresses and contacts (often the e-mail refer to contact persons that have been moved to other offices)
- contradictory information: sometimes in the same document a part contradicts the other
- redundancy in the information
- procedures for which the sequence of steps has to be re-built
- abuse of technical language (i.e., juridical or office-specific jargon)

### 3.2.3. Interview 3 - EU Project Officer (Projects Review and Approval)

**Participants** Project Officer of an EU secretariat for approval of inter-regional projects. The officer has a six years experience in the field.

**Types of procedures** The duty of the secretariat is to define procedures for applicants for EU grants under the INTERREG - "2 Mears Seas Zeeën Programme". Every 7 years a new program is issued and they have to define all the procedures and forms for application, approval and review of the projects submitted. The EU regulations state how the program has to be organised. Then, implemented acts and delegated acts from the EU Commission specify in more detail the regulations. From these documents, they have to define the Operation Manual, which is edited according to the implemented/delegated acts templates. The Operation Manual is made of procedures, rules, application forms. The program specifies the procedures to be followed by the applicants. Then, they define internal procedures to ensure coherence in the day-by-day internal implementation of the program. These are mainly written, but in some obvious cases some freedom is left to the employees. There is also an internal very detailed manual that specifies every single task of the employees. This is updated by an officer who visits the different offices regularly and ask them about their activities. Though there is an internal manual, a novel civil servant is always coupled with an expert who teaches him/her the procedures.

**Operational problems** When a modification occurs in the EU regulation, the head of the office calls the officers and they discuss the impact of the new regulation to their job (e.g., changing the procedure, changing the templates). Regulations are high-level, and they require interpretation. They rarely tell you what to change in the daily activity. Moreover, since a change in the regulation changes also the rules of the program, they have to deliver guidelines to the applicants in accordance with the changes in the regulations. Somehow, they translate in a easy language the regulations for the applicants. Therefore, a large amount of domain knowledge is required. However, operational problems are not frequent for this office, also thanks to the structured and detailed chain of procedures that is clearly documented.

**Linguistic problems** The officer deals with three main regulations: common provision regulation, FESR regulation, and regulation on the EU territorial cooperation. In general, these include many sentences that are understandable for a wide audience. However, the main problems are:

- ambiguous content - both lexically and syntactically - are in general due to politic compromises: they are often written on purpose in an ambiguous way. Moreover, they are often due to inter-linguistic problems: some terms have different meaning when translated in another language, and the EU has 23 different languages and they are all official languages.

- absence of the motivation for a specific rule. This is mostly due to the gap between who makes the laws and who is in charge of making them practical (i.e., their office). However, in general EU regulations are rather clear and practical.
- redundancy in the content. In many cases, the procedure has to be reconstructed. Moreover, it often contains too many contextual elements, things are recalled too many times instead of referencing other documents. The risk is to summarise another document that should be referred, but without the required degree of detail.

### 3.2.4. Interview 4 - SUAP Offices of the Marche Region (Front Desk)

**Participants** Two officers of the SUAP (Sportello Unico Attivit Produttive) of two different offices of the Marche Region. Both officers ere experienced employees.

**Types of procedures** SUAP is a public entity that provides a single interface to industries and small business enterprises for managing al the interactions with the PA. SUAPs resides in single municipalities, and they have to conform to national, regional and province laws and regulation. Normally, when new norms are introduced, these are discussed in meetings at the regional level with different representatives coming from the different SUAPs, to understand issues associated to the new norms. Then, the decisions pass to the offices, but are not explicitly formalised in terms of documents, even if, sometimes, guidelines are defined after the regional meetings. However, what is defined during these meetings have to be formalised through deliberation of the single municipalities. In the end, most of the process is handled through experience and daily practice. However, SUAP is supported through a website, and this makes the process clearer.

**Operational problems** problems occur when a new norm contradicts other norms, or it is unclear the possible conflict with other norms, or it is not clear why a certain norm is cited because the relation is often too generic. Other operational problems are related to the fact that the region might promulgate a law that substitute a preceding law, but the associated practical regulation (edited by the technical staff) takes a year to be promulgated. In the meantime, it is unclear which is the practical regulation that has to be followed.

**Linguistic problems** The main linguistic problems found in norms and regulations used by SUAP offices are:

- absence of motivation for norms or regulations. The intention of the legislator is often unstated. Therefore, norms are kept and followed even if they do not make sense. Moreover, the norms are not abrogated because the motivation of the norm is not clear.
- reference to norms and regulations without a specific motivation

### 3.2.5. Observations on the Interviews

**Heterogeneity** The most evident aspect that we have perceived in performing the interviews is the *heterogeneity* of procedures, documents, and terminology. In the case of interview 1 the procedures involve organizing committes, recruiting personnel, and managing internal practices of the organization. In this case, most of the procedures depend on the circulars, which express variations on existing procedures that are not formally described anywhere, and are procedural knowledge of the employees. In the case of interview 2, procedures mostly involve the redaction of official acts related to contracts and projects, which depend on internal regulations and on national and European ones. Also in this case, a large part of the procedures depend on internal circulars. In the case of interview 3, a complex pyramid of EU laws and regulations are used to come to a structured, and internal process, which,

in turn, is oriented to defined procedures for EU citizens. Similarly, in interview 4, employees have to employ EU laws and regulations, but they have also to *integrate* them with national and local laws, again to define procedures for Italian citizens.

Given this heterogeneity of environments, we argue that a general questionnaire for civil servants should identify also the differences in the types of procedures, and in the types of procedure management approaches, and not solely the defects of the official documents. Moreover, the questionnaire should use the generic term “official documents”, to refer to the heterogeneous acts (laws, regulations, circulars, operational manuals, etc.) that civil servants have to read.

**Operational problems** One common aspect of interviews 1,2 and 4 is the absence of documents that clearly specify the procedures to be followed, and the consequent relevance given to the experience, and critical sense, of the employees. Although in the case of interview 3, the officer tells that they have an internal manual that is continuously updated, also in this case experience plays a fundamental role. Indeed, novel civil servants are always coupled with more experienced ones to learn their procedures. However, apparently, the presence of clear manuals and procedures highly reduces operational problems, which are more frequent in the offices of the other interviews. Operational problems in all the cases are mainly related to *changes* in the procedures. Nevertheless, given the more rigorous structure and documentation of internal procedures in interview 3, such changes are handled in a more effective way.

A part of the questionnaire should be dedicated to understand which are the typical operational problems of the procedures, since we have seen that the clarity of documents associated to procedures is only a dimension of the more complex problem of putting a procedure into practice in a dynamic context.

**Linguistic problems** Besides, the peculiar linguistic problems that we have listed in each interview, the most common problem of the different offices is the absence of *motivation* for rules, procedures, and official documents in general. Currently, all such problems are normally resolved by referring to other offices, although, again with the exception of interview 3, it is not always clear who is the *recipient* of the relevant information. Another common problem is *redundancy* of the information, which, in different terms, is highlighted in all the interviews, and especially in the interview 3, where the abundance of documents also lead do potential redundancy. Technical or *juridical jargon* is also perceived as a relevant problem. However, according to interview 3, such jargon is often needed to clearly specify concepts, and – also according to interview 1 – the problem is more to understand and use such language appropriately. It is interesting also how interview 1 and 3 speak about *ambiguity*. Ambiguity often appears as used on purpose by regulators, especially in the case of EU regulations, which are often the consequence of compromise. This is an important aspect to be considered: laws and regulations tend to leave aspects open to interpretation, and the weight of the decision is often left to the experience and competence of the civil servants.

The linguistic problems identified belong to different categories. Some are purely related to the poor clarity of the language (e.g., juridical jargon, long sentences), some to clarity of the entire document (e.g., relevant content not emphasised), some to the synthesis (e.g., redundancy), some to the external or internal coherence of the document (e.g., overlapping with other rules, internal contradiction). Hence, a general questionnaire should consider these different categories separately, and identify potential defects of the documents in such categories.

## 3.3. Questionnaire

The literature review and the interviews have been used as a basis to define a questionnaire to be submitted to a larger group of civil servants, to understand the relevance of the operational and linguistic problems identified. The questionnaire has been submitted in Italian to two distinct groups of people,

namely part of the administrative staff of CNR-ISTI (17 people), and a set of employees of SUAP offices of the Marche Region (5 people). Here, we first list the questions of the questionnaire, the results, and then we give overall observations on the analysis of the results. We have preferred to keep the results separated between the two groups, since we have obtained heterogeneous answers, and we considered more useful to highlight the differences. Moreover, given the limited number of people in the second group, adding the results of the groups would have hidden the contribution coming from the employees of SUAP offices.

### 3.3.1. Questionnaire Planning and Delivery

Defining a general questionnaire that might be suitable to different recipient implies adopting a vocabulary that is generic, and, at the same time, clear and unambiguous. Therefore, the meaning of relevant concepts (e.g., procedure, operational manual, official document) was clearly specified every time the questions were involving such concepts. Moreover, practical examples were provided when the definitions could to lead to potential misunderstanding.

Given the general public that was going to read the questionnaire, we could not ask questions about *all* the potential defects that are listed in the literature. Many of them are too technical, and we could not ask the readers to express the impact of, e.g., double negative expressions, or passive forms, on the clarity of a procedure. At the same time, we wanted to have a flavor of which are the most relevant defects in PA procedures, as perceived by civil servants. Therefore, our questions have been designed to consider only those defects that emerged from the interviews, and that we could expect to be understood by the general public of civil servants. It is also worth noting that the questionnaire does not solely focus on linguistic defects, but takes into account also other practical/operational problems that the civil servants find when performing a procedure.

The first part of the questionnaire was dedicated to identify the types of profiles answering the questionnaire, and the types of operational problems (i.e., practical problems) found when performing procedures. The second part was focused on the problems of the documents that describe the procedures. The third and last part was dedicated to understanding how the civil servants normally solve the problems of interpretation of such documents. The questionnaire was delivered by means of Google Forms, and was submitted by e-mail to the potential participants of CNR-ISTI and SUAP. The questionnaire requires about 10 minutes to be filled, and the participants had seven days to fill the questions. All data have been treated in anonymous form.

### 3.3.2. Questions

*Experience, Type of Work and Operational Problems*

**Experience**

1) How many years of experience do you have in the public administration? [Less than 1; From 1 to 3; From 3 to 10; More than 10]
2) How many years of experience do you have in the performance of your current duties? [Less than 1; From 1 to 3; From 3 to 10; More than 10]
3) What is your role? [Director; Head of Office; Expert Employee; Employee; Other]

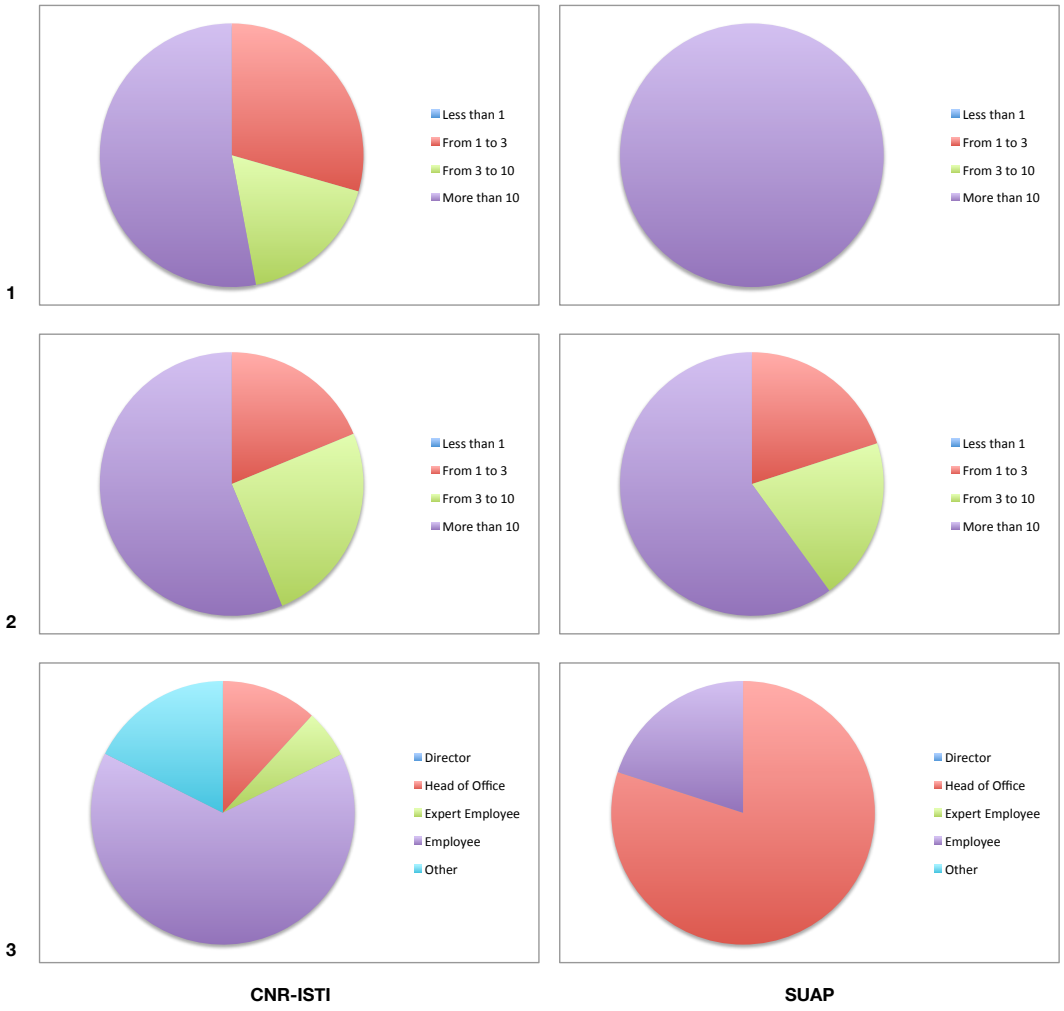



**Figure 3.1: Answers on questions 1, 2 and 3 concerning Experience.**

## Type of Work

1) How often does your work require the interpretation of official documents? *(examples: laws, regulations, circulars, directives)* [Never; Rarely; Sometimes; Often; Always]
2) How often does your work require the preparation of official documents? *(examples: announcements, communications, regulations, circulars, directives)* [Never; Rarely; Sometimes; Often; Always]
3) How often does your work require data entry activities? *(through Web applications, or through software programs such as Microsoft Excel)* [Never; Rarely; Sometimes; Often; Always]

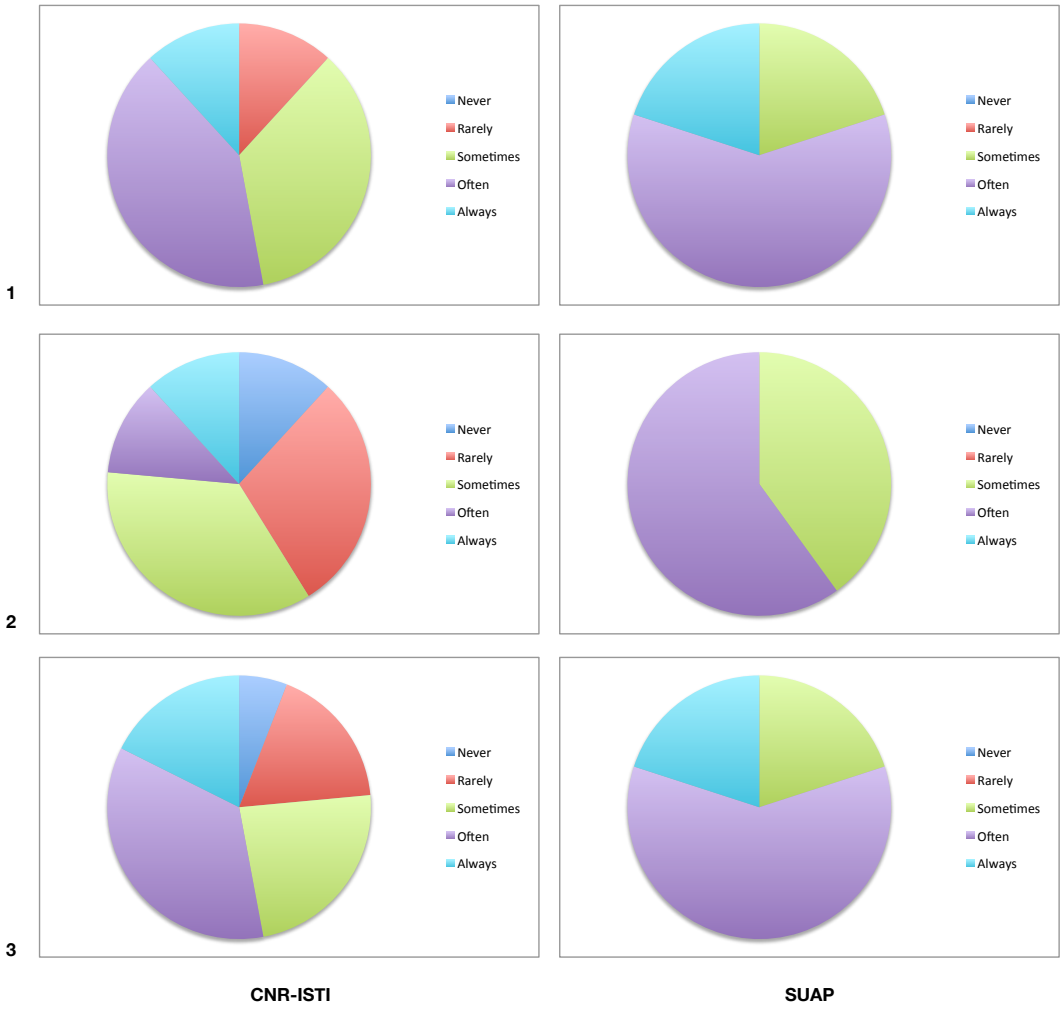



**Figure 3.2: Answers on questions 1, 2 and 3 concerning Type of Work.**

**Type of Procedures** *In this context, a PROCEDURE is any sequence of actions to be performed respecting some rules. A procedure may include the preparation of official documents, performing administrative practices, organizing commissions, performing data entry, etc.*

1) How often does your work require to perform PROCEDURES? [Never; Rarely; Sometimes; Often; Always]
2) The procedures that you perform are:
    1) Described in operational manuals (detailed descriptions of the actions that you have to perform) [Never; Rarely; Sometimes; Often; Always]
    2) Inferred from regulations and internally formalised into operational manuals [Never; Rarely; Sometimes; Often; Always]
    3) Inferred from regulations and not formalised [Never; Rarely; Sometimes; Often; Always]
    4) Inferred from daily practice [Never; Rarely; Sometimes; Often; Always]

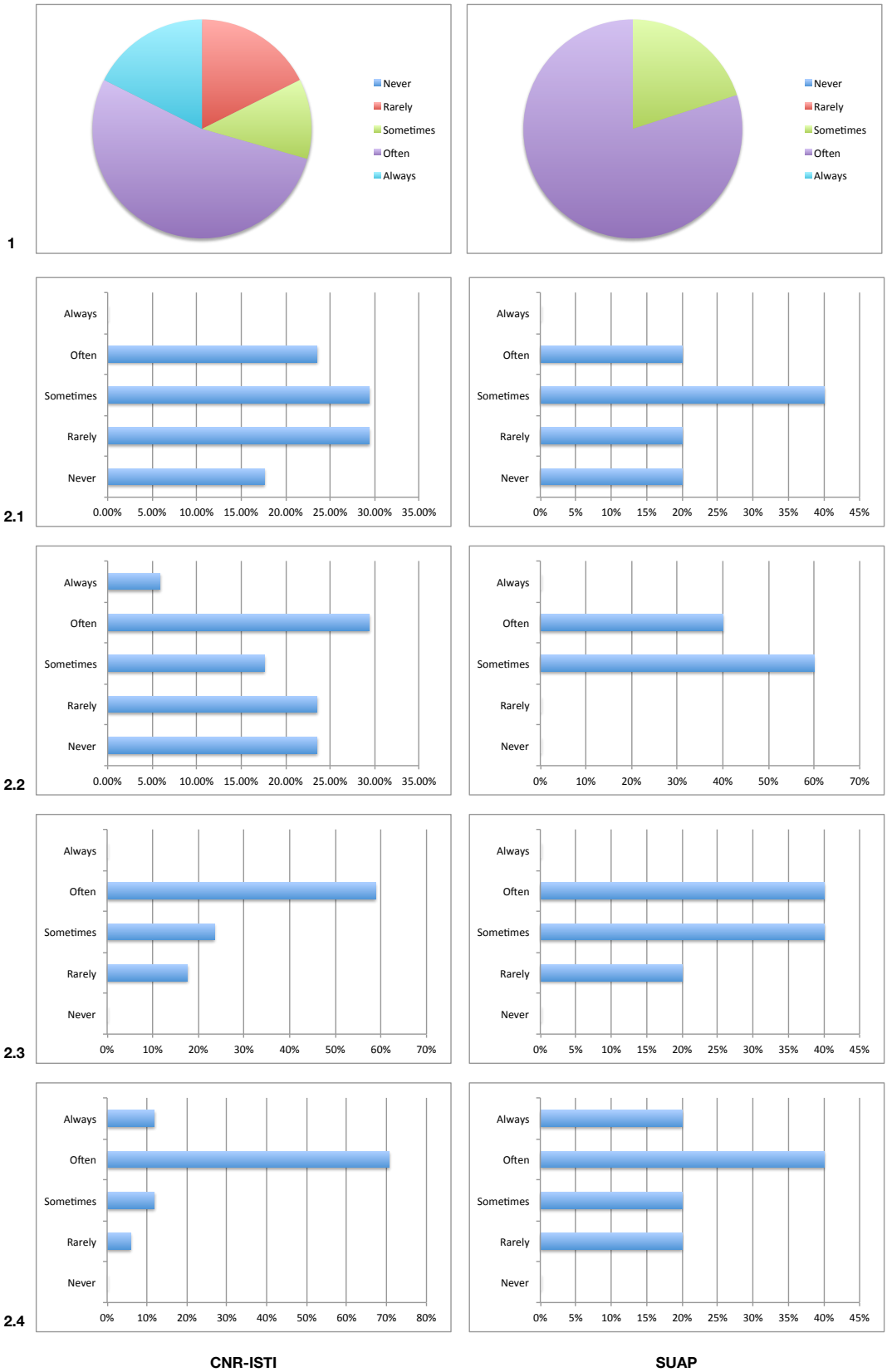



**Figure 3.3: Answers on questions 1 and 2 and 3 concerning Type of Procedures.**

## Operational Problems of Procedures

1) From the operational point of view, what are the most common problems you encounter when performing a procedure? (Select 4 answers maximum)

1) Regulations or manuals associated with the procedure are not known
2) When a new regulation/law amends the procedure, it is not clear how to apply the regulation/law
3) The content of laws, regulations and manuals is not clear
4) The sequence of steps associated with the procedure is not clear
5) The actual procedure is a combination of those described in official documents and must be inferred
6) Your role in the higher level procedure is not clear
7) You did not receive appropriate training
8) You do not have time to dedicate to self-training
9) Regulations have been repealed by new laws for which there is still no implementing regulation

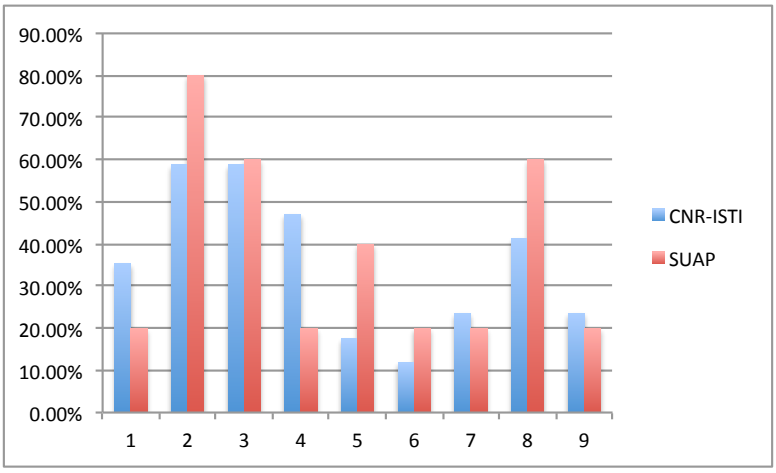



**Figure 3.4: Answers on question 1 concerning Operational Problems of Procedures.**

*Problems with Documents associated to Procedures*

*Consider the official documents that you use to perform your procedures (laws, regulations, circulars, guidelines, manuals , etc.)*

1) In general, is it difficult to interpret the content of these documents? [Never; Rarely; Sometimes; Often; Always]
2) When you have to interpret these documents, which are the most significant CLARITY defects that make the procedure more difficult to put into practice? (Select 3 answers maximum)
    1) The document does not have an explicit argument/topic
    2) The document does not have an explicit motivation/function
    3) The document or the sections do not have a clear title
    4) The document is not divided into sections
    5) The document does not include practical examples
    6) The document does not explain the motivation of specific rules or instructions
    7) There is no glossary
    8) There is no reference to the software tools that shall be used
3) When you have to interpret the SENTENCES of these documents, which are the most significant CLARITY defects that make the procedure more difficult to put into practice? (Select 6 answers maximum)
    1) Sentences are difficult to understand
    2) Sentences are too long
    3) Sentences include too many different concepts

4) If the sentences express rules/instructions, these are difficult to understand
5) If the sentences express rules/instructions, these are difficult to put into practice
6) Relevant terms do not have a clear definition
7) Sentences have an ambiguous structure
8) Sentences use ambiguous terms
9) Sentences use terms that are typical of other offices
10) Sentences use too many synonyms
11) Sentences contain grammatical errors
12) Sentences contain juridical jargon
13) Terms have specific meanings but they are used inappropriately
14) Acronyms and abbreviations are not defined

4) When you have to interpret these documents, which are the most significant SYNTHESIS defects that make the procedure more difficult to put into practice? (Select 3 answers maximum)

1) The document contains too many references to laws and regulations
2) The document contains repetitions
3) The document does not make clear what is important and what is not
4) The document contains lists of steps that are too long
5) The document is too long
6) The document is too detailed
7) The document contains obvious information
8) The document refers irrelevant information

5) When you have to interpret these documents, which are the most significant defects of INTERNAL COHERENCE that make the procedure more difficult to put into practice? (Select 5 answers maximum)

1) While reading the document you realize that relevant information is missing
2) The document does not make clear what are the institutions/offices involved
3) The document does not explain who are the subjects involved
4) The recipient of the document are are unclear
5) The document describes a procedure, but there are no explicit sequence of steps to be carried out
6) The document describes a procedure with an explicit sequence of steps, but some steps are missing
7) The document describes a procedure with an explicit sequence of steps, but the sequence is illogical
8) The document does not explain what to do/who to contact if a problem occurs
9) The document contains parts that contradict each other
10) The document defines constraints that are too strict
11) The document defines constraints that are illogical according to common sense
12) The document leaves too much room for individual choices
13) The structure of the document is not consistent

6) When you have to interpret these documents, which are the most significant defects of COHERENCE WITH OTHER DOCUMENTS that make the procedure more difficult to put into practice? (Select 5 answers maximum)

1) The document does not include enough context information
2) The document combines instruction with context information
3) The document does not have an explicit category
4) Documents in the same category do not have a uniform structure
5) There is inconsistency between the category of the document and its contents

6) The document defines rules or procedures that overlap with other documents
7) The document contradicts other documents
8) The document contains parts of other documents that have been inappropriately copy-pasted
9) The document refers to other documents without justifying the reference
10) The document does not mention other important documents

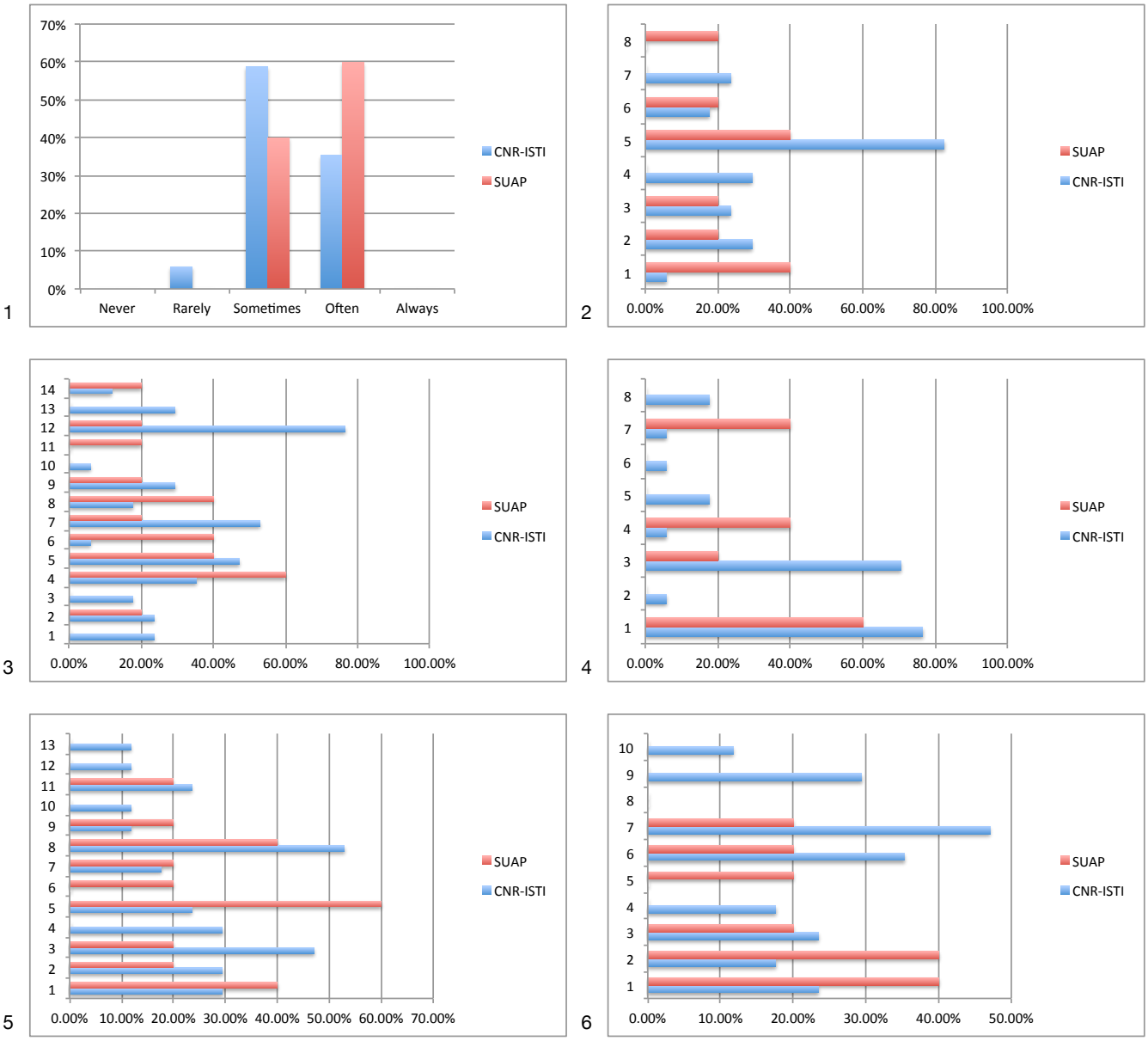



**Figure 3.5: Answers on questions concerning Problems with Documents associated to Procedures.**

*Resolution of the Problems*

1) In general, how do you solve the problems of interpretation of the documents? (Select 2 answers maximum)
   - I search other documents
   - I ask an expert in my office
   - I contact other offices by phone
   - I contact other offices by e-mail

- I try to apply the procedure in practice

2) In general, do you notify the problems of interpretation of the documents? [YES; NO]
3) Would you consider useful a software that allows you to notify the problems of interpretation in a simple way? [YES; NO]
4) Do you keep notes on how to put the procedures into practice? [YES; NO]
5) If you keep notes, these notes:
    - Describe the practical detail of the procedures? [YES; NO]
    - Summarise the basic steps of the procedures? [YES; NO]
    - They serve to summarize regulations and manuals that are too complex? [YES; NO]
    - Do you believe they would be useful to new employees? [YES; NO]

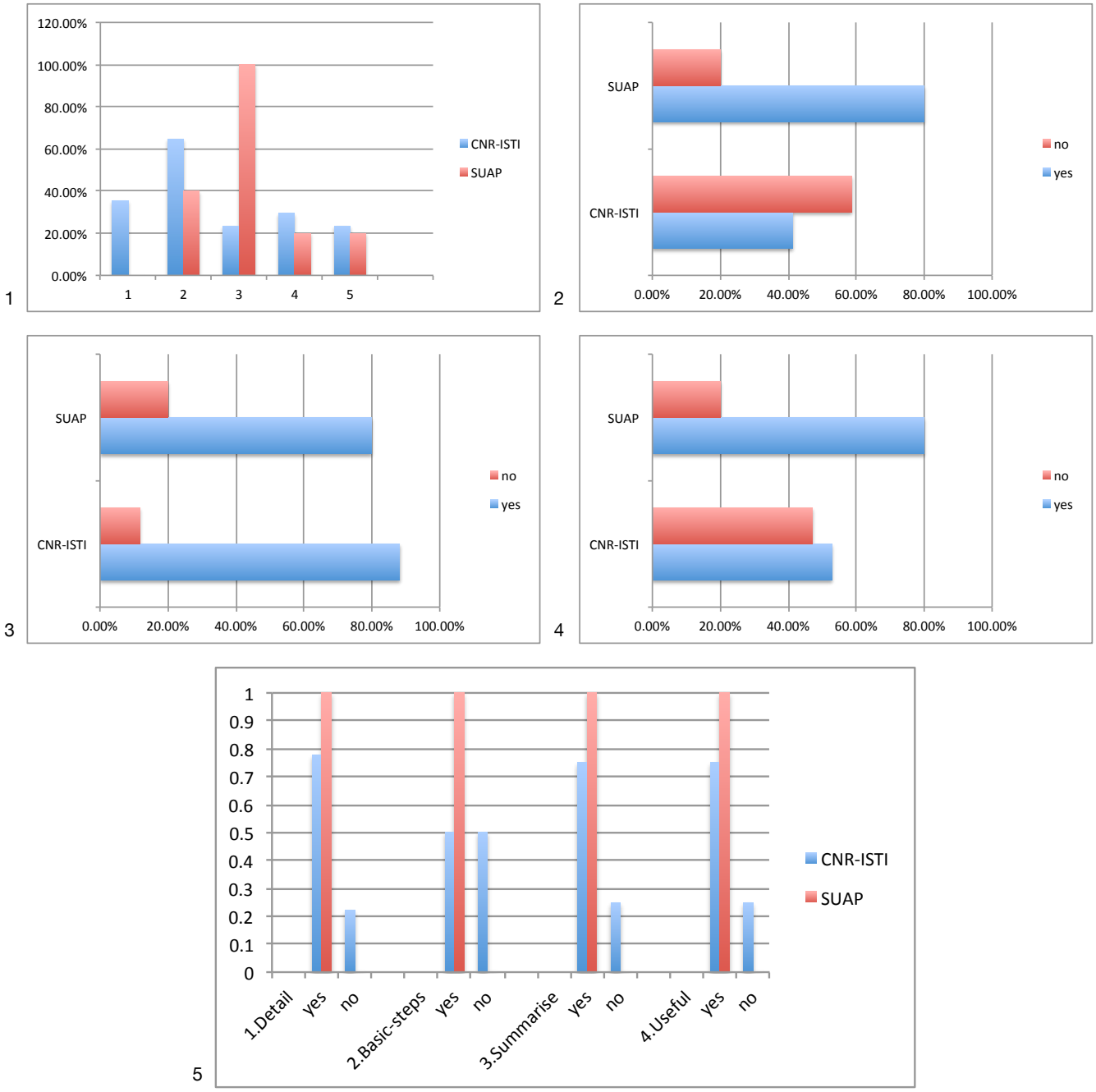



**Figure 3.6: Answers on questions concerning Resolution of the Problems.**

### 3.3.3. Analysis of the Results and Observations

**Experience and Types of Work** Concerning the different characteristics of ISTI-CNR and SUAP personnel, we see that the heterogeneity of the former is higher, both in terms of years of experience in the PA – all SUAP personnel has more than 10 years of experience, while for CNR-ISTI age varies –, in terms of roles covered – SUAP personnel are mainly head of offices, while CNR-ISTI personnel varies from employees to heads – and in terms of characteristics of the work – e.g., in CNR some people never edit official documents, while all SUAP personnel edits official documents from sometimes to often. However, people of both groups are frequently required to *interpret* official documents of the PA: this makes them suitable to answer our questions concerning the quality of such documents.

Although some heterogeneity remains in the answers of CNR-ISTI, also due to the larger group of people interviewed, we see that both CNR-ISTI and SUAP people have often to perform procedures in their job. Operational manuals exist in both groups, but rather often procedures depends non-formalised interpretations of regulations, and on experience coming from the daily practice. The domain knowledge of the employees in making the PA machine work is therefore paramount, and a system like Learn PAd that allows civil servants to contribute with their knowledge to the training of other employees is potentially a key asset.

**Operational Problems** From the operational point of view, the dominant problems of procedures in both groups are (1) modifications to the regulations (i.e., when a regulation impacts on a procedure, it is unclear how such procedure shall be changed); (2) clarity of laws, regulations and manuals associated to procedures; (3) lack of time that employees can dedicate to an autonomous training. Interestingly, a relevant part of CNR-ISTI personnel also complains about the absence of a clear sequencing of steps associated to the procedures. The three main problems identified by both groups are actually the main issue that Learn PAd is planning to address. Moreover, it is worth noting that the second problem, namely the clarity of documents is the topic of the current deliverable.

**Linguistic Problems** Concerning the problems with documents associated to procedures, both groups say that sometimes to often they find difficulties in interpreting such documents. Concerning defects of *clarity* of such documents, answers from SUAP are heterogeneous, with a slight dominance given to the absence of practical *examples*. Such dominance is evident in the answers of CNR-ISTI, where 82% of the people tell that the lack of practical examples is the main problem. Other relevant problems are the absence of a motivation/function for the document and the absence of section partitioning. Looking at the linguistic problems, we see that for CNR-ISTI the main difficulties are associated to the abuse of *juridical jargon* (76.5%), followed by the ambiguous structure of the sentences (52.9%) and by the unclear applicability of the rules or instructions (47.1%). Answers from SUAP in this group are less uniform, and they equally cover almost all of the linguistic defects.

The answers of CNR-ISTI are clear also concerning the defects in terms of *synthesis* that they find in the documents: the document contains too many references to other rules and regulations (76.5%) and the document does not clarify what is relevant and what is not (70.6%). The former is also the main problem registered by SUAP (60%).

Concerning *internal coherence* of the documents, answers from CNR-ISTI are slightly more heterogeneous compared to the previous ones. The main problems are: the document does not explain what to do/who shall be contacted in case of problems (52.9%); the document does not clarify who are the subjects involved (47.1%), the offices involved (29.4%) or the recipients of the document (29.4%). These latter three problems can be summarised with the need to have clear *actors* defined in the documents. Instead, the absence of an explicit sequence of steps is considered the dominant coherence problem in documents used by SUAP.

Contradiction with other documents is the main issue for CNR-ISTI for what concerns *external coherence* of documents (47.1%), followed by overlapping of rules or procedures with other documents (35.3%). Instead, for SUAP, contextual information is critical: 40% of the people tell that documents do

not include contextual information, and that contextual information is often mixed with instructions.

**Resolution of the Problems** To solve problems of interpretation, people from CNR-ISTI use to ask clarifications to other people in their office (64.7%), while people from SUAP tend to contact other offices by phone (100%). This difference is mainly due to the difference between SUAP and CNR-ISTI profiles. While CNR-ISTI profiles are rather various – as witnessed by the first questions –, SUAP profiles are mainly head of offices. Moreover, in small towns, they are often the only employee of such offices, and cannot consult other colleagues directly.

Another relevant difference between CNR-ISTI and SUAP is the notification of problems. While the latter tend to notify problems of interpretation (80%), the former do not (58.8%). However, both groups tell that they would like an easy-to-use software to notify their problems (CNR-ISTI – 88.2%, SUAP – 80%). Finally, notes kept by civil servants of both groups on how to put into practice the procedures might have different forms, from summaries to description of practical details. Moreover, almost all of the answers tell that these notes might be useful to other civil servants. Therefore, again, two other objectives of Learn PAd– notify problems and sharing knowledge – are again confirmed to be relevant for civil servants.

**Overall Observations** The CNR-ISTI group is rather representative of different potential situations of PA offices. CNR is a large research institution, and the administrative staff includes people with plural expertise that have to perform different procedures at different degrees of formalization, as shown by the first answers to the questionnaire. Therefore, we can consider it as sufficiently representative of different, large and *multi-functional*, PA realities. On the other hand, SUAP is a good example of a highly *specialised* reality, with single civil servants acting as heads of office, with front-desk duties. From the answers, we see that, although some problems of the procedure descriptions are common to both realities (e.g., absence of practical examples, excessive number of references), there is a visible heterogeneity in the answers concerning linguistic defects. We conjecture that this implies that one single approach that focuses on a specific linguistic defect will not be able to address all the problems that are relevant for different PA realities. In other terms, one size does not fit all, and, hence, a generic quality checker shall be in principle able to address multiple defects at the same time. Moreover, such system shall be *customisable*, since the general solutions that we can deploy might not fit perfectly with the needs of different PAs. These conclusions have been employed while defining our approach for automated linguistic quality assessment. Moreover, the answers to the questionnaire have been used to *prioritise* our work on specific quality defects. Of course, not all defects raised by the civil servants can be addressed in the limited context of our system. However, the answers to the current questionnaire provide useful hints for further research on document quality and consistency in the PA.

CC BY

# 4 Guidelines and Quality Model

## 4.1. Guidelines

A set of guidelines have been defined that are directed to the contributors to the content of Learn PAd, namely **Content Managers**, and **Learners**. The objective of the guidelines is to let the contributors be aware of the expected quality of their content. Such guidelines are the result of the domain analysis described in Chapter 3, and take inspiration from both public guidelines for defining PA documents, in particular EU Guidelines [22], UK Guidelines [64], US Guidelines [67], from the Plain English Guide [14] and from our interviews. The cited manuals are sufficiently extensive, and provide a large number of examples of good and bad writing styles. Therefore, we did not consider useful to repeat additional examples in these guidelines. Instead, the guidelines have been designed to be concise and clear, and to act as an easily accessible checklist that the contributors can read to verify that their content has the appropriate degree of quality. The list of guidelines is reported in Table 4.2, together with the main source of the guideline. When the source of the guideline is one of our interviews, we use the I1, I2, I3, or I4 identifier, to specify which interview gave birth to such guideline. When the guideline cannot be traced to a specific source, but has been defined to address specific defects that we consider relevant (as, e.g., grammatical errors, see guideline 4.4), or specific formatting issues (as, e.g., labels for steps, see guideline 3.3), the source is referred with the identifier D. Guidelines are partitioned into five groups, namely **General** (i.e., guidelines that impact the whole procedure description), **Fields** (i.e., guidelines that specify the fields needed in a description – these guidelines are enforced by the Learn PAd template described in Sect. 5.6), **Steps** (i.e., guidelines associated to the partitioning of the procedure into steps), **Sentences** (i.e., guidelines associated to the writing style and the clarity of the text), and **Warnings** (i.e., guidelines associated to the specification of exceptional situations).

Moreover, guidelines associated to defects that are identified as more relevant according to our questionnaire are collected in Table 4.1. Such table can be used as a minimal reference for **Content Managers**, and **Learners** to avoid the most relevant mistakes when editing their content.

| ID | Guideline |
|---|---|
| 1.1 | Divide the procedure into steps |
| 1.3 | Motivate the procedure and the steps |
| 2.5 | Specify the intended reader of the procedure |
| 2.6 | Specify the subjects involved in the procedure |
| 2.7 | Partition the content into sections |
| 4.22 | Do not use juridical jargon |
| 4.5 | Avoid linguistic ambiguities in words and sentences |
| 4.2 | Highlight keywords and relevant content |
| 2.9 | Do not reference too many resources/procedures that are not strictly relevant |
| 1.8 | Your procedure shall not contradict/overlap with other procedures |
| 1.7 | Provide examples |
| 1.6 | Put it into practice what you wrote to check its applicability |
| 5.4-5 | Specify people to contact in case of problems |

**Table 4.1: List of most relevant guidelines, according to our questionnaire.**

| ID | Guideline | Source |
|---|---|---|
| **General** | | |
| 1.1 | Be clear, concise and coherent | [22] |
| 1.2 | Divide the procedure into steps | [67] |
| 1.3 | Motivate the procedure and the steps | I4 |
| 1.4 | Leave space for individual choices | I1 |
| 1.5 | Do not describe obvious / common-sense issues | [22] |
| 1.6 | Put it into practice what you wrote to check its applicability | D |
| 1.7 | Provide examples | I1 |
| 1.8 | Your procedure shall not contradict/overlap with other procedures | I2 |
| **Fields** | | |
| 2.1 | Provide a glossary | [22] |
| 2.2 | Define an overview of the procedure | I1 |
| 2.3 | Specify the topic of the procedure | D |
| 2.4 | Specify the scope/context of the procedure | I1 |
| 2.5 | Specify the intended reader of the procedure | I2 |
| 2.6 | Specify the subjects involved in the procedure | I2 |
| 2.7 | Partition the content into sections | D |
| 2.8 | Specify the tools needed to perform the procedure (web link, documents, etc.) | D |
| 2.9 | Do not reference too many resources / procedures that are not strictly relevant | I1 |
| 2.10 | Reference other relevant procedures / documents instead of repeating their content | I3 |
| **Steps** | | |
| 3.1 | Divide a procedure in logically linked steps | [67] |
| 3.2 | Separate the steps with new lines | D |
| 3.3 | Define a label for each step | D |
| 3.4 | Use bullet points or numbered lists to identify the steps | [67] |
| 3.5 | If the chronological order of the steps is important, use a numbered list | [67] |
| 3.6 | If the order of the step is not important (steps can be performed in parallel), use bullet points | [67] |
| 3.7 | Use action verbs in steps (Do, Make, Fill-out, etc.) | [14] |
| 3.8 | Use the imperative action verb at the beginning of each step | [14] |
| 3.9 | If conditions apply to the action, include them before the action verb | [14] |
| 3.10 | Do not mix instructions in steps with contextual information. Give the necessary contextual information before the instruction. | D |
| 3.11 | Use 7 to 10 steps maximum for each procedure | [64] |
| 3.12 | If more than 7-10 steps are needed, partition the procedure into sub-tasks | [64] |
| 3.13 | Give clear headings to each sub-task | [64] |
| 3.14 | Use clear and informative headings throughout, make headings verbose | [22] |
| 3.15 | Use a uniform structure for all parts of the procedure | I2 |
| **Sentences** | | |
| 4.1 | Clarify acronyms and abbreviations | [64] |
| 4.2 | Highlight keywords and relevant content | I1 |
| 4.3 | Delete redundancies | I2 |
| 4.4 | Avoid grammatical errors | D |
| 4.5 | Avoid linguistic ambiguities in words and sentences | I1 |
| 4.6 | Use connectives (hence, therefore, etc.) between sentences | [67] |
| 4.7 | Use short sentences (max 25 words) | [64] |
| 4.8 | Use short paragraphs (max 5 sentences) | [64] |

| | | |
|---|---|---|
| 4.9 | Cover only one topic per sentence / paragraph | [67] |
| 4.10 | Avoid double negations | [67] |
| 4.11 | Keep subject, verb and object close together | [67] |
| 4.12 | Use the word “must” for obligations | [67] |
| 4.13 | Use verbs instead of nouns (“evaluate” instead of “carry out an evaluation of”) | [22] |
| 4.14 | Do not use synonyms for important terms | D |
| 4.15 | Do not use passive voice and name the subject who performs the action | [22] |
| 4.16 | Use adverbs only rarely | D |
| 4.17 | Avoid inconsistent use of terminology | I2 |
| 4.18 | Avoid inconsistent / contradictory content | I2 |
| 4.19 | Adapt the terminology to the target audience | [22] |
| 4.20 | When recycling text (copy/paste), make sure to properly adapt it | [22] |
| 4.21 | Do not use difficult terms | [64] |
| 4.22 | Do not use juridical jargon | [64] |
| **Warnings** | | |
| 5.1 | Define warnings at the beginning, or before the step causing the warning | [22] |
| 5.2 | Tell the reader what to do if he/she makes a mistake | I1 |
| 5.3 | Include questions that you imagine the reader might have, and answer them | [22] |
| 5.4 | Specify people to contact in case of problems with the understanding of the procedure | I1 |
| 5.5 | Specify people to contact in case of problems with the practical implementation of the procedure | I1 |

**Table 4.2: List of guidelines for contributors of the Learn PAd content.**

## 4.2. Quality Model for LearnPAd

To develop a system that is able to check that a document that describes a PA procedure exhibits a certain degree of quality, a quality model has to be defined. A quality model is a reference model against which a certain artifact – a PA procedure expressed in natural language in our case – can be evaluated [33]. A quality model is defined by means of a set of quality attributes, which are high-level quality properties that the PA procedure shall exhibit. Each quality attribute is associated to a set of indicators, possibly partitioned into sub-categories. An indicator is a measurable characteristic of the PA procedure that provides information about a quality attribute.

According to the domain analysis performed, we have first defined a general quality model for PA procedures. From such model, we have selected a subset of indicators to be automatically checked by means of rule-based strategies. The selection has been driven by the results of our questionnaire, which helped us selecting the most relevant quality indicators to be checked. Part of the indicators have also been checked by means of machine learning techniques.

The general quality model for PA procedures is reported here in two parts for the sake of visualisation (see Fig. 4.1 and 4.2). Seven general **quality attributes** have been defined, namely:

- **Clarity:** this attribute tells that the PA procedure is understandable, both in terms of content, in terms of presentation, and in terms of practical applicability.
- **Non-ambiguity:** this attribute tells that the content of the PA procedure has only one interpretation, independently of the reader. The attribute considers the non-ambiguity of terms, and the non-ambiguity of the syntax used in the sentences of the PA procedure.
- **Simplicity:** this attribute tells that the content of a PA procedure is easy to read. The attribute considers both the difficulty of the terms and the difficulty of the syntax.

- **Completeness:** this attribute tells that all the required fields of a given template for PA procedures are filled with content. The attribute requires a reference template to be defined.
- **Conciseness:** this attribute tells that the PA procedure is sufficiently synthetic, and does not have any irrelevant detail or repetition.
- **Correctness:** this attribute tells that the content of the PA procedure is correct in terms of grammar, and does not include copy-paste errors.
- **Coherence:** this attribute tells that the content of the PA procedure is not contradictory or illogical. The attribute takes into account the internal coherence, the external coherence (i.e., the coherence with other documents), and the coherence with respect to the real world (referred as applicability incoherence).

For each quality attribute, a set of indicators have been defined, in some cases partitioned into subcategories. The indicators are the leafs of the mind-maps shown in Fig. 4.1 and 4.2. Each indicator is associated to a "scope", expressed in squared brackets in the figures. The scope tells at which level of granularity the indicator can be potentially checked: TERM means that the indicator can be checked at the level of single or multi-word terms of the document; SENT means that the indicator is associated to the sentences; PART means that the indicator is associated to a group of sentences; DOC means that the indicator impacts the whole document. Here, we will not describe each single indicator, since we argue that their meaning can be easily understood from the figures. Moreover, for those indicators that will be checked by means of rule-based strategies, we will provide accurate descriptions in Chapter 5.

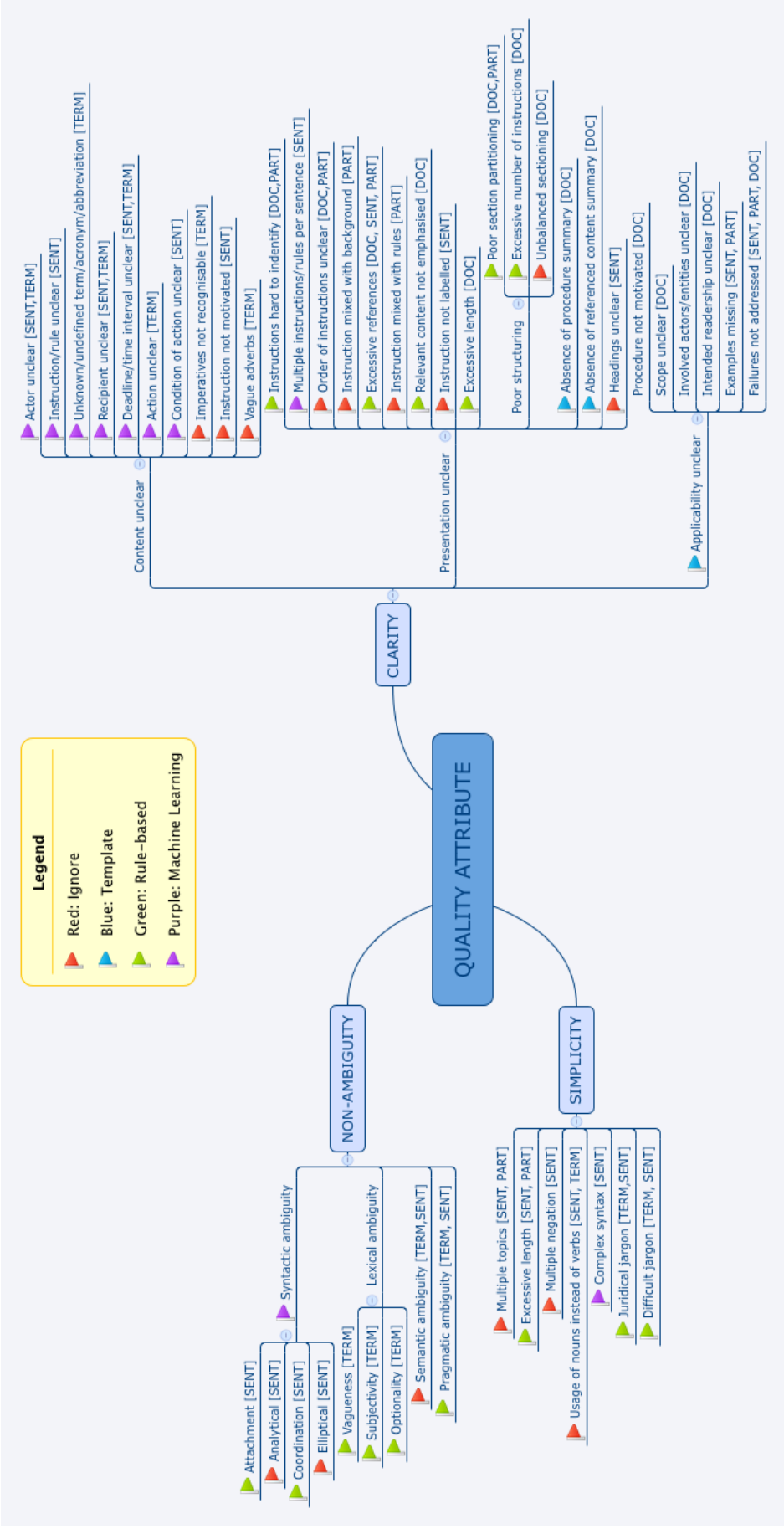



**Figure 4.1: Quality Model for Public Administration procedures - Part 1.**

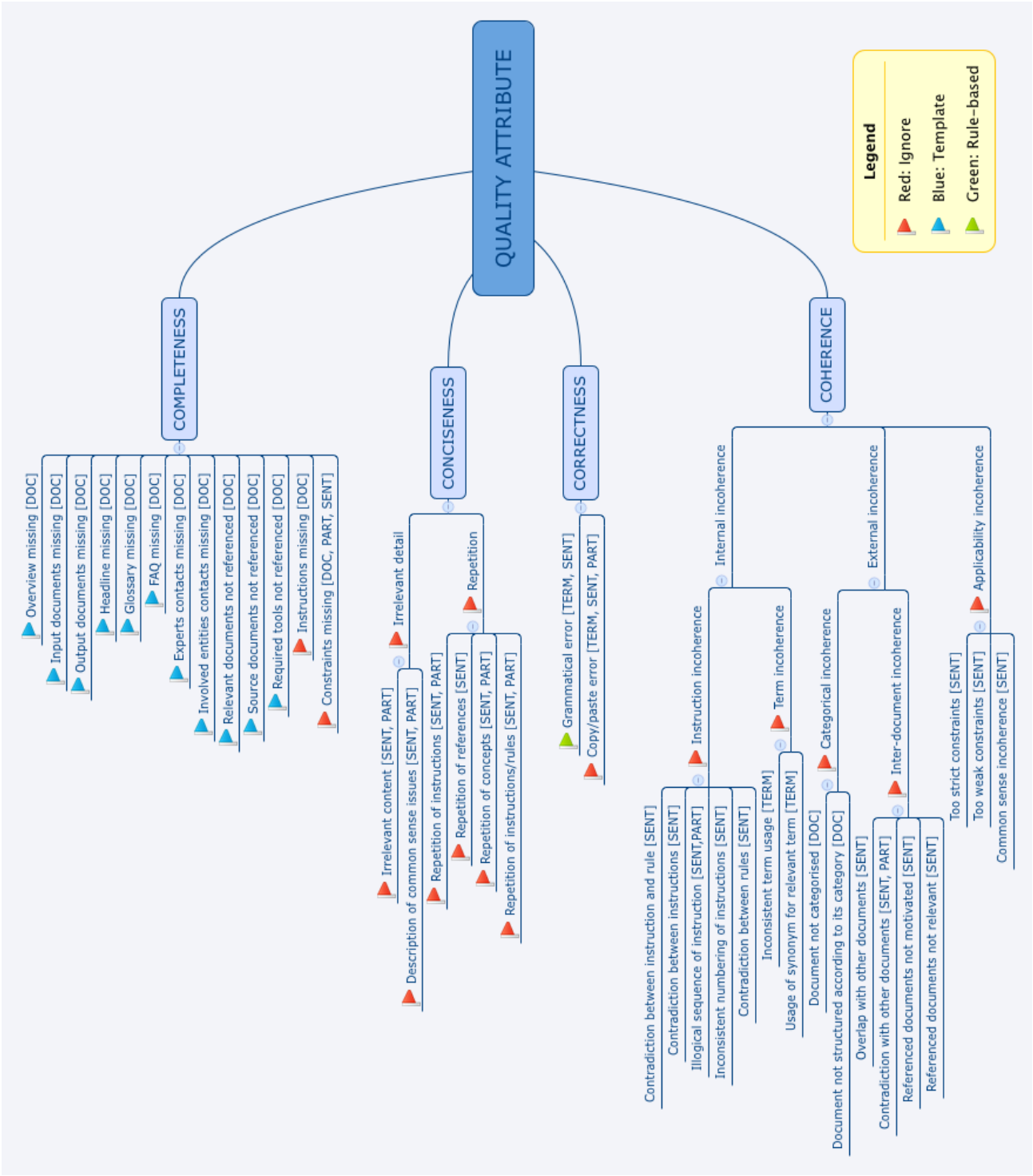



**Figure 4.2: Quality Model for Public Administration procedures - Part 2.**

The quality model defined in Fig. 4.1 and 4.2 has been used to define the quality model for Learn PAd. To this end, for each indicator in the original quality model, we have established a means to compute it in the context of Learn PAd. Some indicators can be computed by means of rule-based approaches (Green flag), some by means of machine-learning approaches (Purple flag), and some by means of a template (Blue flag) – i.e., the PA procedure is required to conform to a given template. It is worth noting that some indicators associated to the clarity attribute (e.g., all indicators under "Applicability unclear", and two indicators under "Presentation unclear") can be addressed by introducing specific fields in a template – such template is defined in Sect. 5.6. In this way, *clarity* defects can be addressed

by transforming them into *completeness* defects.

Part of the indicators, and in particular those associated to Coherence and Conciseness, have been ignored (Red flag). Indeed, such indicators can in principle be addressed by means of rule-based, or, more likely, machine learning techniques. However, to define specific strategies for each of these indicators further research is still needed.

The resulting quality model is depicted in Fig. 4.3, and lists all the indicators that will be actually checked in Learn PAd, and that are discussed in Chapter 5.

The original quality attribute named “Clarity” has been partitioned into “Content Clarity” and “Presentation Clarity”. Quality defects associated to unclear applicability have been resolved by introducing appropriate fields in the template. The quality model does not include the indicators that we have experimentally checked by means of machine learning, since this activity, presented in Sect. 6 did not lead to results that can be directly employed in the Learn PAd platform. However, for part of such indicators, we have defined corresponding rule-based approaches. Hence, the reader will notice that some of the indicators with the Purple flag in Fig. 4.1 – namely “unclear acronym” and “actor unclear” – appear also in Fig. 4.3 under the Content Clarity quality attribute.

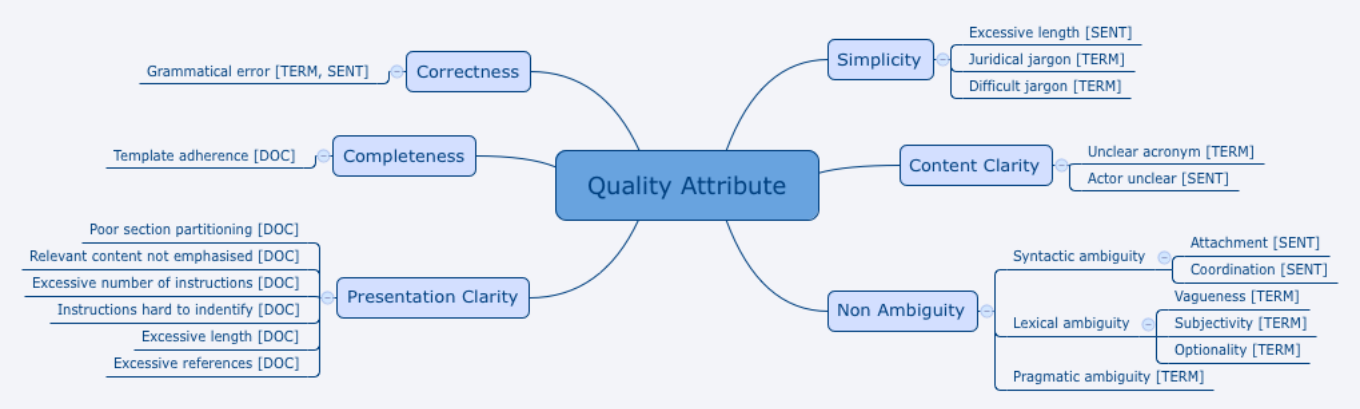



**Figure 4.3: Quality Model for Learn PAd.**

For each quality attribute, a quality measure has been defined, as a number expressed between 0 and 1, with lower values indicating poor quality and higher values indicating good quality. More formally, given a PA procedure description $D$, the quality measure $Q$ is a function $Q : D \rightarrow [0, 1]$. The measure is defined for each quality attribute in Chapter 5. Hence, we have six quality measures in total, namely $Q_{SIM}$ for Simplicity, $Q_{AMB}$ for Non-Ambiguity, $Q_{CC}$ for Content Clarity, $Q_{PC}$ for Presentation Clarity, $Q_{COM}$ for Completeness, and $Q_{COR}$ for Correctness. Each value is associated with a percentage. For specific intervals, the numeric percentage is mapped to human-readable values, shown in the Quality Evaluation Page (see Sect. 2.4). The mapping is reported below:

- 0% - 25%: VERY BAD - Quality is very poor, errors shall be corrected;
- 25% - 50%: BAD - Quality is poor, errors shall be corrected;
- 50% - 75%: GOOD - Quality is acceptable, but there are still some errors;
- 75% - 99%: VERY GOOD - Few errors;
- 100%: EXCELLENT - No errors found.

At this stage, these quality measures will not take into account the *severity* of the different indicators. i.e., the fact that one indicator might have a higher impact than another on the quality. Appropriate tuning in this sense is foreseen in future works. Moreover, we do not provide an overall quality measure. Indeed, the quality measures are computed in different ways – i.e., based on sentences, or based on

the whole document – and an aggregate value of these measures would not give an appropriate flavor of the actual quality of the PA procedure under evaluation.

As evidence that the indicators of the Learn PAd quality model have been selected to address the most relevant defects, Table 4.3 relates the indicators – and the sections where they are described – with the most relevant guidelines, already listed in Table 4.1.

| ID | Guideline | Indicator | Sect. |
|---|---|---|---|
| 1.1 | Divide the procedure into steps | Instructions hard to identify | 5.5.3 |
| 1.3 | Motivate the procedure and the steps | Template adherence | 5.6 |
| 2.5 | Specify the intended reader of the procedure | Template adherence | 5.6 |
| 2.6 | Specify the subjects involved in the procedure | Template adherence | 5.6 |
| 2.7 | Partition the content into sections | Poor section partitioning | 5.5.1 |
| 4.22 | Do not use juridical jargon | Juridical jargon | 5.2.2 |
| 4.5 | Avoid linguistic ambiguities in words and sentences | Syntactic, Lexical, Pragmatic Ambiguity | 5.3 |
| 4.2 | Highlight keywords and relevant content | Relevant content not emphasised | 5.5.2 |
| 2.9 | Do not reference too many resources/procedures that are not strictly relevant | Excessive references | 5.5.6 |
| 1.8 | Your procedure shall not contradict/overlap with other procedures | - | - |
| 1.7 | Provide examples | Template adherence | 5.6 |
| 1.6 | Put it into practice what you wrote to check its applicability | - | |
| 5.4-5 | Specify people to contact in case of problems | Template adherence | 5.6 |

**Table 4.3: Mapping between most relevant guidelines and associated indicators.**

Guidelines 1.6 and 1.8 do not have corresponding indicators in Learn PAd. Indeed, the former is a generic guideline that is associated to applicability coherence, while the latter is related to external coherence. As previously specified, coherence is not considered in the context of this study.

The reader will also notice that part of the indicators are not associated to highly relevant guidelines (e.g., excessive number of instructions, difficult jargon, actor unclear, etc.). The decision to check also such additional indicators has been driven by the observation, given in Sect. 3.3.3, that one single approach that focuses on a limited number of linguistic defects will not be able to address all the problems that are relevant for different PA realities, and hence, an additional set of indicator is desirable.

# 5 Linguistic Quality Assessment Strategies

## 5.1. Overview

This chapter describes the strategies that we have defined to assess the quality of the linguistic content of Learn PAd. Such strategies stem from the quality model that we have described in Sect. 4.2. Indeed, for each quality attribute, we have identified a set of *indicators*, which can be automatically calculated or detected and provide information about a particular quality attribute of a document [33]. In a sense, indicators can be regarded as *defects* seen from the perspective of the automated quality checker.

For each indicator, here we define strategies that aim to compute or detect the presence of that indicator in the text. The strategies that we define are *rule-based* or *algorithmic*. By *rule-based* we intend strategies that use rules to check the presence of an indicator in the text. Here, rules are regular expressions that might involve characters or more complex linguistic constructs, such as words, and phrases. To express simple rules we generally use an intuitive semi-formal notation that use natural language and symbols. To express more complex rules (as, e.g., actor unclear – Sect. 5.4.1, syntactic ambiguity – Sect. 5.3.2), we use a notation inspired to the JAPE grammar, which is the one employed by the tool GATE [66]. This is a rather intuitive grammar, which we considered suitable to express complex rules in a clear, but sufficiently formal, way. Other indicators (as, e.g., unclear acronym – Sect. 5.4.2, pragmatic ambiguity – Sect. 5.3.3) are computed by means of *algorithmic* strategies. In these cases, we textually describe the algorithm and refer external resources for more detailed descriptions. Each strategy has been designed to identify the majority of potential defects. The idea, borrowed from the requirements engineering domain [5], is that the system raises the possibility of a defect in the text, and that the user considers whether such defect is an actual defect, or can be ignored.

For each indicator, we also provide *recommendations*, which the Content Analysis component of Learn PAd will associate to the defective part of the text as shown in the Inspection Page of Fig. 2.4. Indicators associated to the Presentation Clarity and Completeness attributes are not associated to specific part of the text. Therefore, for these indicators, the recommendations will be presented in the Quality Evaluation Page of Fig. 2.4.

It is worth noting that our indicators are defined for the English language. Our choice fell on this language, since this is a language with more linguistic resources, and it is also the official language adopted in the Learn PAd project. However, for each indicator, we considered useful to highlight its degree of language dependency, since some of the indicator can be equally applied independently from the language.

Each of the following sections is dedicated to a quality attribute, and to the associated indicators. Moreover, Sect. 5.8 provides a preliminary evaluation of part of the strategies that have been defined.

## 5.2. Quality Attribute: Simplicity

The *simplicity* quality attribute defines how easy is to read a natural language description in Learn PAd. It is a quality attribute that, in a sense, shall give an overall degree of readability of each sentence, and compute an aggregate value of readability. Such quality attribute takes into account the difficulty of the terms. The difficulty associated to the syntax – a topic that is still a matter of research, see. e.g., [18]

– instead is considered by simply evaluating the length of the sentences. We use the term “simplicity” and not “readability”, since readability in the literature is a more domain-generic concept, which, as discussed in Sect. 3.1.4, involves also typographical aspects and degree of interest that a text raises. Here, we wish to highlight that the defects that we address are those that makes *difficult* the understanding of PA procedure descriptions, such as, e.g., juridical jargon and difficult jargon. Therefore, we have considered the term simplicity to be more appropriate.

To compute a quality measure for this quality attribute we define the following formula:

**Quality measure.** $Q_{SIM}$ = 1 - (Number of defective sentences / Total number of sentences).

Each indicator defined for this quality attribute is designed to tell if a sentence is defective or not. If a sentence is considered defective by *at least* one indicator, such sentence is counted in the “Number of defective sentences” in the formula. In other terms, sentences that have only one defect will count like sentences that have more defects.

Here, a **sentence** is defined as any string of text comprised between a sentence-start marker, and a sentence-end marker. Sentence-start markers are the beginning of the file, or any sentence-end marker. Sentence-end markers are full stops (“.”), question/exclamation marks (“?”, ”!”), and newline. Specific treatments allow to identify usage of full stops in file extensions (e.g., *.exe), IP addresses (e.g., 127.0.0.1), common abbreviations (e.g., Prof., Ph.D.), and acronyms (e.g., F.A.O., N.A.T.O.), which are discarded from the computation of sentence-end markers. If a string of text is comprised between quotes, between “–” symbols, or between parentheses is not considered as a separate sentence. This definition of sentence is applied throughout the rest of the document.

The following sections describe the indicators that we consider for this attribute.

### 5.2.1. Indicator: Excessive length

This indicator tells that a sentence is too long. The length of a sentence is a rather intuitive indicator of its complexity. Normally a long sentence includes multiple concepts that have to be processes by the reader, and is more likely to include complex syntactic constructions that require higher reading effort. An example of long sentence is provided below:

- **Long Sentence:** *Further distribution of vote sheets within the staff is permissible upon issuance of the vote, but distribution outside the agency is permissible only after the final collegial decision is recorded by the Secretary in an SRM to the action office and the votes have been released to the public.* This sentence is 49 words, and 293 characters, and it requires multiple readings to be understood.

This indicator can be easily checked with this basic rule:

- **RULE 1:** $N$ = number of words in a sentence, $N < \tau$.

The *The Plain English Guide* by Martin Cutts [14] state that sentences should be 15-20 words in average, and should not exceed 40 words. Moreover, the style guidelines of the English government [64] recommends sentences to not exceed 25 words. Therefore, in the context of Learn PAd, we take the threshold $\tau$ of 26 words as basic rule to check whether a sentence is too long. Such threshold can be increased or decreased by the Guidelines Manager of Learn PAd.

| |
|---|
| **Recommendation:** Shorten the sentence. A sentence should not exceed 25 words.<br>**Language Dependency:** this indicator is independent from the language. |

### 5.2.2. Indicator: Juridical jargon

Juridical jargon is the usage of terms and constructions that belong to the juridical domain. This domain has defined a specific jargon that is understood by domain experts, and in a sense, is oriented to

establish clear concepts and to avoid ambiguity. Nevertheless, studies as [63] have shown that even technical experts prefer text that use plain English instead of legal jargon, and that the more specialist the knowledge of the reader, the higher the preference for plain English. These studies have been used also by the UK government to define their guidelines for editing the content of their Web pages [64], where they recommend to minimize the usage of juridical jargon, and latin terms, which are typical in legal writing. Moreover, our interviews and questionnaires show that the presence of juridical jargon is one of the main linguistic problems found in their current procedure descriptions.

To address this problem, we define the current indicator – i.e., juridical jargon – which aims to identify juridical words and expressions in the Learn PAd content. It is worth mentioning that the term "jargon" includes not only words and expressions, but also the syntax. Here, we focus solely on the terms (i.e, words and expressions), since other indicators are defined in Learn PAd that address problem with ambiguous syntax (see Sect. 5.3.2), a typical problem of juridical jargon.

Let $J$ be a set of juridical terms, let $S$ be a sentence and let $T(S)$ be the set of any ordered sequence of words in a sentence (i.e., any potential single or multi-word term). The following rule checks the presence of juridical terms.

**RULE 1:** $\forall j \in J, \forall t \in T(S)$, if $t == j$, mark $t$ as juridical jargon.

The set $J$ of juridical terms used in Learn PAd is composed of 877 terms in total. To compose this set, we have merged comprehensive glossaries selected from the Web. In particular, we have merged juridical terms from (a) the glossary provided by NY-COURTS.GOV, the New York State Unified Court System[1], (b) the glossary provided by the Judicial Branch of the State of Connecticut [2], and (c) the list of legal Latin terms in Wikipedia [3].

**Recommendation:** The term $t$ is juridical jargon. Substitute $t$ with a more common term.
**Language Dependency:** this indicator is dependent from the language. However, lists specific for each language can be defined.

### 5.2.3. Indicator: Difficult jargon

This indicator quantifies the amount of sentences using terms (single and multi-words) that are considered difficult, either because they are rare, or because they are overly complex expressions that can be substituted with simpler ones. The Dale-Chall formula [8] measures the readability of a text by taking into account the percentage of words in the text not included in a list of $3,000$ words considered easy-to-read. Such formula has two primary defects in our context: (1) It gives only an index and does not tell the editor which term is defective, i.e., hard to read; (2) the set of $3,000$ words is too restricted and risks to raise too many warnings. Indeed, a 5-6 years old child normally already uses $2,500$-$5,000$ common words [64], and by age 9, people normally build the set of words that they use every day. This set is normally composed of two sub-sets, a primary set (around $5,000$ terms), and a secondary set (around $10,000$ terms). Though also the secondary set includes terms that are used in every day life, such set includes also terms that are less common, and, hence, more difficult. Therefore, to identify the usage of difficult jargon, we define a rule that, for each sentence, checks that each term is contained in the primary set. More formally, let $S$ be a sentence, and let $W(S)$ be any word in the sentence. Moreover, let $E$ be the set of $5,000$ terms that belong to the primary set of easy-terms. The following rule checks the presence of difficult jargon:

**RULE 1** $\forall w \in W(S)$, if $w \notin E$, mark $w$ as difficult jargon.

If a sentence has at least one word that is detected to be difficult, according to the previous rule, such sentence will be marked as defective. As set $E$, we have used the set of top-5000 most common terms available at [16].

The previous rule checks that terms used in a sentence are easy-to-read for a general public, and

[1] http://www.nycourts.gov/lawlibraries/glossary.shtml
[2] http://www.jud.ct.gov/legalterms.htm
[3] https://en.wikipedia.org/wiki/List_of_legal_Latin_terms

it is domain independent. Indeed, the list of common words is based on the selection of the most frequent words in genre-balanced corpus [16]. To detect difficult expressions that are *specific* of PA documents, we resort to use the list of pompous terms that litter official writing [55]. Such list of terms has been edited by the Plain English Campaign[4], with the objective of making official writing easier to read. While the list of easy words include only single-word terms, this list includes also multi-word terms (e.g., "acquaint yourself with", "despite the fact that", *etc.*). Therefore, we define a rule to check the presence of difficult jargon according to such list. Let $D$ be the set of difficult terms. Let $S$ be a sentence, and let $T(S)$ be any sequence of words in the sentence. The rule is as follows:

**RULE 2:** $\forall d \in D, \forall t \in T(S)$, if $t == d$, mark $t$ as difficult jargon.

If a sentence has at least one term that is detected to be difficult according to one of the previous rules, such sentence is marked as defective. As set $D$, we have used the mentioned set of $407$ difficult terms listed in [55].

**Recommendation:** The term $t$ is difficult. Substitute $t$ with a simpler term.
**Language Dependency:** this indicator is dependent from the language. However, lists specific for each language can be defined.

## 5.3. Quality Attribute: Non-Ambiguity

The *non-ambiguity* quality attribute defines the degree of non-ambiguity of a NL description in Learn PAd. Such quality attribute considers both the ambiguity of the terms (at lexical and pragmatic level) and the ambiguity of the syntax. As for the *simplicity* quality attribute, we define the following quality measure:

**Quality measure.** $Q_{AMB}$ = 1 - (Number of defective sentences / Total number of sentences).

Each indicator defined for this quality attribute is designed to tell if a sentence is defective or not. If a sentence is considered defective by *at least* one indicator, such sentence is counted in the "Number of defective sentences" in the formula. In other terms, sentences that have only one defect will count like sentences that have more defects. The following sections describe the indicators that we consider for this attribute.

### 5.3.1. Indicator: Lexical ambiguity

In general, a lexical ambiguity occurs whenever a term can have different meaning (e.g., the word "bank" can be the bank of a river, or the bank as "establishment for custody, loan, exchange, or issue of money") [6]. However, in this context, we will not refer to this definition of lexical ambiguity – cases as the one exemplified will be treated as *pragmatic* ambiguity, since the interpretation of "bank" depends on the context. Instead, we will refer to the model defined by Gnesi *et al.* [33], for checking the quality of NL requirements specification. According to such model, lexical ambiguity occurs whenever a sentence includes an adverb, adjective or conjunction, possibly combined with prepositions, that might lead to different interpretations of the sentence. In practice, the considered model does not take into account *names* or *verbs* with potentially different interpretations, but solely typical expressions that are commonly source of potential misunderstandings. Four categories of lexical ambiguity are defined in [33], namely vagueness, subjectivity, optionality and weakness. The first category includes the usage of vague expressions, with a non uniquely quantifiable meaning, such as "accurate", "suitable", "appropriate", "clearly", etc. The second category includes expressions that refers to personal opinions or feelings, such as "better", "accordingly", "depending on", etc. The third category includes expressions that reveal the presence of an optional part in the sentence, such as "if necessary", "if needed", "and/or". The fourth category include cases when a weak main verb, such as "can", "may", etc., is used. Examples for the first three categories are provided below:

[4] http://www.plainenglish.co.uk

- **Vagueness:** *The field office will forward the application to the **appropriate** official for a final decision.* Here, the term "appropriate" is vague, and the editor shall specify which is the specific official that is in charge of taking the final decision.
- **Subjectivity:** *Support staff may be called in from other teams **depending on** the extent of the scene.* Here, the expression "depending on" leaves the reader with the freedom to personally evaluate the extent of the scene.
- **Optionality:** *The director of the group must transfer 10% of the funded loans to the institute **and/or** to the department.* Here the expression "and/or" leaves the freedom of sending the funded loans to just one organisation.

In the context of Learn PAd, we do not consider cases of of "weakness", since this indicator was specifically designed for NL requirements specifications, and appeared less suitable for PA documents. Indeed, in the context of PA procedure descriptions, we have found that it is rather frequent to find verbs such as "can" or "may" (e.g., 63 cases of"can", and 124 cases of "may" are found in our data-set), and these are normally acceptable (as, e.g., in the following example *'"Ensure you can meet the deadlines"*).

To check the presence of vagueness, subjectivity or optionality in a sentence, we define three rules. Let $V$, $U$ and $O$ be sets of vague, subjective, or optional terms. Let $S$ be a sentence, and let $T(S)$ be any sequence of words in the sentence. The rules are the following:

**RULE 1:** $\forall v \in V, \forall t \in T(S)$, if $t == v$, mark $t$ as vague.
**RULE 2:** $\forall u \in U, \forall t \in T(S)$, if $t == u$, mark $t$ as subjective.
**RULE 3:** $\forall o \in O, \forall t \in T(S)$, if $t == o$, mark $t$ as optional.

If a sentence has at least one term that is detected to be vague, subjective of optional according to the at least one of the previous rules, such sentence is marked as defective. In Learn PAd, we employ the dictionaries used by QuARS [33], to check the three categories of lexical ambiguity exemplified above. Therefore, the sets $V$ (446 terms), $S$ (19 terms) and $O$ (11 terms) are composed of all the terms used by QuARS.

**Recommendation:** The term $< t >$ is <vague|subjective|optional>. Remove $t$ or substitute it with a more unequivocal term.
**Language Dependency:** this indicator is dependent from the language. However, lists specific for each language can be defined.

### 5.3.2. Indicator: Syntactic ambiguity

Syntactic ambiguity manifest itself whenever the sentence can have more than one grammatical structure, each one with a different meaning. Four types of syntactic ambiguity are defined in the literature [6], namely *analytical* (i.e., a complex noun group with modifiers [35]), *attachment* (i.e., a prepositional phrase can be attached to two parts of the sentence), *coordination* (i.e., when more than one conjunction "or", or "and" is used in a sentence), *elliptical* (i.e., when words are omitted because they are expected to be deduced from the context), and *anaphoric/referential* (i.e., when pronouns or other words refer to other elements, but there is more than one possibility). This latter type of ambiguity may involve different sentences, and the literature often categorise it as pragmatic ambiguity. However, given its strong relation with the syntax, and its similarity with, e.g., attachment ambiguity, we consider more reasonable to include it among the syntactic ambiguities.

Examples of each category are provided below:

- **Analytical:** *The Italian office director.* Here, "Italian" can be referred to the office or to the director.
- **Attachment:** *The officer edits a resumee with a template for the final assessment.* Here "for" can be referred to the "template", or to the "resumee" or can specify a deadline (i.e., before the final assessment).

- **Coordination:** *The employee met the council **and** the head of office **and** the secretary assessed his presence.* Here, the sentence can have several parses. For example, it is unclear whether both the head of office and the secretary assessed the presence of the employee, or just the secretary.
- **Elliptical:** *The successful candidate receives the letter on Sept. 12, and the unsuccessful doesn't.* Here, the ambiguity is whether the unsuccessful candidate receives a notification in another date, or does not receive any notification.
- **Anaphoric:** *The delegate assesses the presence of the candidate, and **he** provides his signature.* Here "he" can be referred to both the delegate or the candidate.

In principles, all such types of ambiguity could be detected by having a parser that outputs the possible syntax trees for a sentence, each one with a degree of probability, and by checking whether two or more parse exist that have similar probability. Statistical parsers exist that output multiple syntax trees for a sentence. For example, we have performed experiments with the Stanford Parser[5], which, for each sentence, produces a set of syntax trees – the number can be chosen by the user –, scored according to their likelihood to be correct parse of the sentence. The likelihood is expressed as a log probability. However, we have seen that, given any sentence, ambiguous or not, the probability of the syntax trees are really close – in general, less than 1% –, and therefore we cannot leverage these indexes for syntactic ambiguity checking. For example, consider the following two sentences:

- **Ambiguous:** *The employee shall send the minute meeting with e-mail attachments;*
- **Non Ambiguous:** *The employee shall send the minute meeting as attachment to the e-mail*.

The first sentence is ambiguous (i.e., it has an attachment ambiguity), since the minute meeting could be part of the e-mail attachment, or they could be written in the e-mail, and associated with some attachments. The second sentence is instead clear: the minute meeting are attachments. However, the parse trees for the first sentence have -80 and -80.3 probability, while for the second sentence the parse trees have -89.76 and -89.77. Though the second sentence is not ambiguous, its parse trees are much closer in terms of probability. Similar situations occurred with other sentences in the "Monti Azzurri - Titolo Unico" document from Deliverable 8.1 that we have used for our feasibility tests (115 sentences). Also the probability number can be of little help for syntactic ambiguity checking. Indeed, the probability strongly depends on factors like the length of the sentence, the rarity of the words in the sentence, and whether word dependencies in the sentence are known to the parser or not. In other terms, we have found that using state-of-the-art statistical resources for parsing is of little help for syntactic ambiguity detection.

Therefore, we have decided to focus on a sub-set of the syntactic ambiguity categories and to provide rule-based approaches for them. The chosen categories are coordination and anaphoric ambiguities. The choice has fallen on these categories since they are more clearly defined in the literature, and can be in principle associated to the presence of specific keywords (e.g., "and", "or" for coordination ambiguities, and pronouns for anaphoric ambiguities). The other types of syntactic ambiguities are more likely to be identifiable with machine learning approaches.

**Coordination Ambiguities** Potential coordination ambiguities may occur when we have more than one coordinating conjunction in the form "or" or "and" in the same sentence, as in the example provided in the previous page. Moreover, they may occur when a conjunction is used with a modifier, as e.g., in the phrase *"**Novel employees and directors** are required to provide summaries of their work at the end of the year"* (is "novel" referred to employees only, or to both employees and directors?). To detect these types of ambiguity, two rules, one for each type, can be provided.

[5] http://nlp.stanford.edu/software/lex-parser.shtml

- **RULE 1:** Any sentence including the following pattern *P* = (Token)* (and | or) (Token.kind != ”punctuation”)* (and | or) (Token)*
- **RULE 2:** Any sentence including the following pattern *P* = (JJ) (NN | NNS) (and | or) (NN | NNS).

The first rule searches for at least two occurrences of “and” or “or”, not separated by punctuation (e.g., commas, semicolons, separator such as “-”, etc.). As reported in [6], commas, and other types of punctuation may clarify the syntactic structure. Coordination ambiguity may occur also in presence of punctuation. However, we have evaluated these cases are sufficiently rare to be negligible. The second rule matches cases where an adjective (JJ) precedes a couple of singular (NN) or plural nouns (NNS), joined by “and” or “or”.

**Anaphoric Ambiguities** Anaphora occurs in a text whenever a linguistic expression (e.g., personal pronouns such as “he/she/it”, possessive pronouns as “her/his”, relative pronouns such as “that”, “which”, demonstrative pronouns such as “this”, “who”, etc.) refer to a previous part of the text. The referred part of the text is normally called *antecedent*. An anaphoric ambiguity occurs if the text offers one or more antecedent options, either in the same sentence or in previous sentences [72]. Here, we focus on anaphoric ambiguities that involve third personal subject/object pronouns and possessive pronouns, of the three genders, namely male (“he”,“his”, “him”, “himself”), female (“she”, “her”, “hers”, “herself”), and neuter (“it”, “its”, “itself”, “they”, “their”, “theirs”, “them”, “themselves”). We do not focus on first and second person pronouns, since these are less frequent in PA documents.

The potential antecedents for these pronouns are noun phrases (NP) [72]. Therefore, we define the following two rules to identify potential cases of anaphoric ambiguities.

**RULE 3.** Any sentence including the following pattern $P$ = (NounChunk) (NounChunk)+ (Pronoun)

**RULE 4.** Match any part of text including the following pattern $P$ = (NounChunk) (NounChunk)+ (Split) (Pronoun)

The first rule matches any single sentence with a pronoun and two or more potential antecedents. The second rule searches for potential antecedents in the previous sentence (the notation “Split” indicates the sentence separator).

A sentence is considered defective whenever it is matched by one of the rules defined for coordination or anaphoric ambiguities. Depending on the type of syntactic ambiguity, different recommendations are issued by Learn PAd.

**Recommendation (Coordination Ambiguity):** The sentence is ambiguous because you are using complex combinations of“and” or “or”. Clarify the sentence by introducing some commas, or by splitting it into two sentences.
**Recommendation (Anaphoric Ambiguity):** The sentence is ambiguous because you are pronouns instead of names. Clarify the sentence by replacing the pronouns with names.
**Language Dependency:** this indicator is dependent from the language. Rules have to be re-defined to be applicable to the syntax of other languages.

### 5.3.3. Indicator: Pragmatic ambiguity

Pragmatic ambiguity occurs whenever the interpretation of a term or sentence depend on the context [6, 28]. The context is composed of all the factors that might influence the interpretation of the term/sentence. In a textual document, these factors are the terms/sentences that occurs previously or subsequently in the document, the domain of the document, and the background of the reader. Let us consider the following example:

- **Pragmatic Ambiguity:** The **director** shall be able to start the **operation** before the end of July 2013.

The meaning of the term“operation” might be in principle clarified by the previous sentences of the document (i.e., the document might specify the *type* of operation). However, if the previous sentences do not clarify the type of operation considered, or specify multiple types of operation, the reader might give a potentially wrong interpretation to the sentence. A similar problem might occur with the term “director”. If the document is directed to a specific PA office, then the director might be understood as the director of the office. If the document is directed to a group of offices in an organization, with multiple layers of management (e.g., department director, office director, institution director), such term might cause misunderstandings. Therefore, the interpretation of the sentence also depends on the background of the reader of the sentence.

Detecting ambiguities of this type is a complex task. We have developed an approach that models the background knowledge of potential readers of a document, and that compares the interpretations of a sentence given by these readers. The details of the approach can be found in [28]. Along the Learn PAd project, we have experimented such approach considering a document with 114 sentences. The obtained results, reported in [29], have been considered promising, but, according to our experiments, the processing time required by the approach (i.e., 250 seconds for each sentence, 8 hours in total) makes it not suitable for a quality checker such as the one envisioned in Learn PAd. Indeed, we expect the quality checking to be performed in reasonable time (i.e., in terms of seconds), and we cannot expect a document editor to wait hours before the analysis is terminated.

Therefore, in the context of Learn PAd, we decided to develop a novel approach to detect pragmatic ambiguity cases. The idea of the approach stems from the observation that a sentence such as the one above can be better clarified if the names “director” and “operation” would be associated with a specifier. In other terms, the previous sentence would not be ambiguous from the pragmatic point of view – or at least its potential degree of ambiguity might be reduced – if rephrased as follows:

- **Non-ambiguous:** The **director of the office** shall be able to start the **assessment operation** before the end of July 2013.

In principle, pragmatic ambiguity might be associated to name, verbs, adjectives, *etc.*. Here, we focus only on names. However, the idea of the approach can be in principle extended to other part-of-speech. The approach consist of two steps. First, we detect all the names that are not coupled with adjectives, and that are not followed by a specifier such as “of” (**Step 1**). Then, for such names, we search for their definition in Wikipedia. If Wikipedia leads to a disambiguation page, the name is considered ambiguous (**Step 2**). The idea of using Wikipedia is driven by the rationale that Wikipedia contains domain specific definitions in a large variety of domains. Moreover, the disambiguation page for a word in Wikipedia often contains different definitions, each one specific for a domain. For example, if we search the term “director” in Wikipedia, we are redirected to a disambiguation page that lists the possible different meaning of director in the Arts (e.g., film director, music director), Business (e.g., managing director, executive director), and other domains. Hence, our algorithm works as follows.

**Step 1.** We first use the following regular expressions among part-of-speech to select singular and plural names (NN and NNS, respectively) with associated adjectives (JJ) or other names (as, e.g., “managing director”), either before the name (first rule) or after the name (second rule). Moreover, we also search for names followed by the the preposition “of” (third rule). The notation Token indicates any type of part-of-speech.

**Find Specified Names 1:** (JJ | NN | NNS) (NN | NNS)

**Find Specified Names 2:** (NN | NNS) (JJ | NN | NNS)

**Find Specified Names 3:** (NN | NNS) (“of”) (Token)

All the names matched by the previous rules are discarded from the set of potential ambiguity.

**Step 2.** All the names that are not matched by the previous rules are searched in Wikpedia, e.g., by means of the Wikipedia API for Python[6]. If the search gives a disambiguation error – i.e., – the term is marked as "`ambiguous`", otherwise is marked as "`not ambiguous`". Any sentence that contains a term marked as "`ambiguous`" is considered defective.

## 5.4. Quality Attribute: Content Clarity

The *content clarity* quality attribute defines the degree of clarity of a NL description in Learn PAd. Clarity of content is associated to specific aspects of sentences that make them more *understandable* from the procedural point of view. In other terms, this attribute focuses on aspects associated to the applicability of a procedure, such as the presence of well-defined actors in a sentence, and the presence of clear time constraints.

To compute a quality measure for this quality attribute we define the following formula:

**Quality measure.** $Q_{CC}$ = 1 - (Number of defective sentences / Total number of sentences).

Each indicator defined for this quality attribute is designed to tell if a sentence is defective or not. If a sentence is considered defective by *at least* one indicator, such sentence is counted in the "Number of defective sentences" in the formula. In other terms, sentences that have only one defect will count like sentences that have more defects. The following sections describe the indicators that we consider for this attribute.

### 5.4.1. Indicator: Actor unclear

This indicator tells that the actor of an action is unclear. This might occur in different cases, as e.g., in the following examples:

- *The **officer** shall send the review form within 5 days from the reception of the review request.*
- *The procedure shall be carried out before the end of March 2015.*

In the first case, it is unclear which officer is in charge of sending the review form. This situation might be resolved though the other sentences of the documents – where the concept of officer might be defined –, and can be apportioned to the cases of potential *pragmatic ambiguities*, discussed in Sect. 5.3.3. The second case, instead, is using the passive voice, and this is a typical case where the subject of the action, i.e., the actor, is not specified in the sentence, and he/she is therefore unclear. However, a simple "by" could help specifying the actor, as in the following rephrasing:

- *The procedure shall be carried out by the certification authority before the end of March 2015.*

In this section, we will define rule to identify cases similar to the one shown in the second example. The rule below has been defined such cases:

**RULE 1.** Any sentence containing the following pattern $P$ = (Auxiliary) (RegularPP | IrregularPP)+ ($\neg$ "by")

The rule matches any case where we have a term that indicates the presence of at least an auxiliary verb (i.e., "am", "are", "were", "being", "is", "been", "was", "be") followed by one or more past participle in regular form (i.e., any term terminating with "-ed") or irregular form (e.g., "written", "spent", "proven", etc. – a list of 175 irregular verbs have been used). Moreover, the rule checks the presence of the preposition "by" following the verbs, as indicator of the potential specification of an actor. A sentence is considered defective whenever it is matched by rule above.

[6] https://pypi.python.org/pypi/wikipedia/

**Recommendation:** The sentence does not specify the subject. Please include who is performing the action.
**Language Dependency:** this indicator is dependent from the language. Rules have to be re-defined to be applicable to the syntax of other languages.

### 5.4.2. Indicator: Unclear acronym

An acronym is word made from the initial letters or parts of other words, generally used to identify organisations (e.g., NATO, NASA, etc.) or domain specific concepts (e.g., BPMN, SQL, etc.). An acronym is normally composed of capital letters, which can be separated by full stops (e.g., F.A.O.), or not (e.g., FAO). This indicator checks for acronyms that are never expressed in their extended form (e.g., North Atlantic Treaty Organization for NATO). Though from our interviews does not emerge a peculiar relevance given to undefined acronyms, we have seen that this is instead a relevant problem in the data-set used for feedback-based quality evaluation (see Sect. 6). Indeed, such procedure descriptions include a large amount of sentences with acronyms, and in most of the cases the meaning of such acronyms is not defined in any part of the text. Though some acronyms are commonly used, many acronyms found are domain specific, or even procedure specific and need to be defined to clarify their meaning.

In Learn PAd, we define an algorithm that makes use of regular expressions to check the presence of unclear acronyms in a document. The algorithm first searches for potential acronyms (**Step 1**). Then scans the document to search for sentences where the potential acronym occurs together with its definition; if no sentence is found, the acronym is marked as unclear (**Step 2**).

**Step 1** The following regular expression is used to find potential acronyms:

**Find Acronyms:** $[A-Z|\backslash.]\{2,\}$

The expression matches any string of text with capital letters or full stops, if it is composed of at least two characters. This expression includes cases of sequences of full stops, and terms written in capital letters (e.g., “PROTOCOL” in a capitalized title). After the execution of the regular expression, these cases are discarded from the list of potential acronyms. In practice, all potential acronyms made of full stops are discarded, as well as sequence of capital letters longer than 5 character.

**Step 2** In each sentence where the acronym appears, the algorithm checks if a sequence of words exist that express the acronym in its extended version. The following regular expression is used to find the presence of a potential extended version of an acronym of length “len” in a sentence. The value of “len” is computed without counting the full stops (CNR and C.N.R. have both len = 3).

**Find Acronym Definition:** $([A-Z]+\backslash w+([\ \ ]|)))\{\text{len}\}$

The regular expression searches for sequences of length “len”. The sequences are required to be composed of one or more capital letters, followed by any word character ($\backslash w$), followed by a space ([ ]), or not (to detect final words). Finally, the algorithm checks that each capital letter in the matched string matches the capital letters found in the candidate acronym.

If the extended version of an acronym is found in at least one sentence in the document, the acronym is marked as `''clear''`, and no defect will be raised if the acronym appears in the rest of the document without its extended version. If no sentence exist where the acronym appears together with its extended version, such acronym is marked as `''unclear''` in each sentence where the acronym appears. In turn, each sentence including an `''unclear''` acronym will be marked as defective.

## 5.5. Quality Attribute: Presentation Clarity

This quality attribute defines the degree of clarity of the presentation of the NL content in Learn PAd. Such quality attribute considers the clarity of the presentation format (i.e., bullet list, enumerations, bold

characters, etc.), and not the clarity of the content. To define the indicators associated to the presentation clarity we have taken inspiration from general guidelines for Web accessibility (as, e.g., [12]), as well from the more specific UK Government guidelines [64, 65]. Often, these resources do not provide numeric parameters for checking the presentation quality of a text that shall appear in a computer screen, as in the case of Learn PAd. Therefore, in such cases, we resort to define arbitrary, but reasonable numeric parameters (e.g., maximum lenght of a page, amount of emphasised text, *etc.*), which can be tuned by the Guidelines Manager of Learn PAd.

To compute a quality measure for this quality attribute we define the following formula:

**Quality measure.** $Q_{PC}$ = 1 - (Number of defective indicators / Total number of indicators).

Each indicator for this quality attribute is associated to a binary decision: Defective/Not Defective. The decision, in some cases, depends on a threshold to be specified for each indicator. The reader might notice that, for this attribute, the computation of the quality measure focuses on indicators and not on sentences. The choice has been driven by the fact that, in general, presentation aspects involve defects that affect the whole document (e.g., excessive length, poor section partitioning, etc.), and not single sentences. It is worth noting that, while other quality attributes do not need information about the format of the text (e.g., font style, list, *etc.*), this quality attribute specifically focuses on such information. Since XWiki provides the possibility to export the HTML version of the content, the indicators for this attribute will be defined over HTML tags, when required. The following sections describe the indicators that we consider for the presentation clarity attribute.

### 5.5.1. Indicator: Poor section partitioning

This indicator tells that a document is not properly partitioned into sections. This implies that no sectioning is provided, and that paragraphs (i.e., groups of sentences separated by a blank line) are too long. According to our questionnaires, the absence of section partitioning is one of the main problems in understanding the current procedure descriptions. In the context or Learn PAd, as for HTML pages, a section is normally identified by a header. Therefore, a rule shall be defined that checks the presence of headers in the page. Moreover, a rule shall be provided to check the partitioning of the text into paragraphs. Two rules are therefore defined to check this indicator:

**RULE 1:** $N$ = number of <h* > tags, $N > 1$.

**RULE 2:** $L$ = number of sentences between <p>, $L < \tau$.

The first rule checks that there is at least two <h* > tags, which identify headers in HTML. The "*" notation indicates that we check the presence of h1, h2, ..., h6, which are tags that identify headers in HTML. The threshold $\tau$ for the second rule is set according to the recommendation of the UK government [64], which recommends to use less than 5 sentences for each paragraph, therefore $\tau = 5$. Paragraph are identified with the HTML tag <p>, which is the proper tag to identify paragraphs. If one of the rules is violated once, this indicator will be set to Defective.

**Recommendation (RULE 1):** Partition your document into sections.
**Recommendation (RULE 2):** Split your paragraphs. Each paragraph shall be less than 5 sentences.
**Language Dependency:** this indicator is independent from the language.

### 5.5.2. Indicator: Relevant content not emphasised

This indicator tells that the relevant content is not properly emphasised with respect to the rest of the text. The weak emphasis given to relevant content is one of the most important problems encountered by civil servants according to our questionnaire. Emphasis in textual documents is normally given by providing **bold** or *italic* text, by using capital letters, by increasing the size of the relevant text with respect to the rest of the content, or by introducing visual frames around the relevant text. However, to our knowledge, no specific and quantitative guideline is provided in the literature that specify how much

of a text should be emphasised. We argue that this lack of guidelines also depends of the variability of the amount of relevant content in a text, e.g., a document might contain a large amount of relevant text, and another document might contain very few useful information. Moreover, relevance is also a *situational* concept [70], and depends on the perceived utility of the reader, and of the task that the reader has to perform. In the context of Learn PAd, we will not discuss how to identify what is relevant and what is irrelevant in a text. Instead, we will check that at least a certain amount of text is emphasised in a Learn PAd page. The underlying assumption is that, if someone provides some content, she/he is also expected to highlight what is relevant in that content according to hishe/her point of view. Our indicator will specifically focus on the amount of **bold** terms, with respect to the rest of the text. Indeed, in Web content, italic is often used for foreign words, capital letters are normally discouraged [65], font size might vary in headings and other structural elements, and frames are used also for tables and other visual items. In other terms, bold is the most common way for giving emphasis in a text. The following rule will check that the amount of bold text is at least $X\%$ of the rest of the text.

**RULE 1:** $n$ = number of terms within  and <b> tags, $N$ = total number of terms, $n/N \cdot 100\% > X\%$

The  and <b> HTML tags are the tags commonly visualised as bold by the browsers. At this stage, we set $X = 10$. As discussed, no guideline exist to define the amount of text that should be emphasised. Therefore, the choice of this number is currently arbitrary and can be changed by the Guidelines Manager of Learn PAd. If the previous rule is violated, this indicator will be set to Defective.

**Recommendation:** Highlight in bold the relevant sentences and keywords of your text.
**Language Dependency:** this indicator is independent from the language.

### 5.5.3. Indicator: Instructions hard to identify

This indicator is oriented to identify cases where instructions are mixed with contextual information, and it is therefore hard to identify them. Normally, a procedure shall specify instructions in the form of bullet point or numbered lists, as also stated in our guidelines. Therefore, in Learn PAd, we will check that such lists exist in the text. The following rule assess the presence of lists in the text:

**RULE 1:** $N$ = number of <ol> or <ul> tags, $N > \tau$

The <ol> and <ul> HTML tags specify the presence of ordered (i.e., enumeration) or unordered lists (i.e., bullet points). The rules checks that at least $\tau$ lists appear in the text. Of course, the number of lists needed might depend on the procedure. However, since we wish to check that some list is provided, at this stage we set $\tau = 0$. The number can be changed by the Guidelines Manager, if she/he sees that such control is too strong. If the rule is violated, the indicator will be set to Defective.

**Recommendation:** Provide bullet point lists or numbered lists for your instructions.
**Language Dependency:** this indicator is independent from the language.

### 5.5.4. Indicator: Excessive number of instructions

This indicator tells that a too large number of instructions is used. Normally, instructions come with bullet-point lists or enumeration. Following the study in [57], the UK government recommends that such lists shall be between 5 and 10 items. Therefore, we define the following rule to assess this indicator:

**RULE 1:** $N$ = number of <li> tags between <ol> or <ul> tags, $N < \tau$.

The <ol> and <ul> HTML tags specify the presence of ordered (i.e., enumeration) or unordered lists (i.e., bullet points), while the items in the list are indicated by the tag <li>. Here, for each list, we count the number of items, and we check that is lower than $\tau$. Following the guidelines of the UK government, we set $\tau = 11$. If the rule is violated once, this indicator will be set to Defective.

**Recommendation:** Limit the number of elements in the lists. Each list shall not be longer than 10 items. If needed, split the list into sub-tasks.
**Language Dependency:** this indicator is independent from the language.

### 5.5.5. Indicator: Excessive length of the document

This indicator tells that the document is too long and shall be partitioned into more pages. As for the excessive length indicator, which checks the length of sentences, the following simple rule can be applied.

**RULE 1:** $N =$ number of words in the document, $N < \tau$.

The choice of the threshold $\tau$ depends on the content to be written in the page, and also the UK government does not provide specific recommendations on the length of a page [64]. However, to establish a threshold, we can take inspiration to the study reported by the Media Corporation [7]. Such study is focused on blog posts. The Media Corporation have evaluated the average time spent on on blog posts compared this to its expected time to read. The conclusion of the study is that the optimal minute-length of a post is 7 minutes. In other terms, a post that sufficiently engage the reader shall be around 1600 words long, since a 7 minutes read comes in around 1600 words. In our context, we have to consider that the content of Learn PAd is different from blog content, which, in general, is oriented to information or pleasure, and can include redundant content. However, the content of Learn PAd is also oriented to *learning*, and engagement is a paramount aspect in learning. Therefore, we set our $\tau$ to 1600, and we leave further adjustments to the Guideline Manager. If the rule is violated, the indicator will be set to Defective.

**Recommendation:** The document is too long. A document shall not be longer than $\tau$ words.
**Language Dependency:** this indicator is independent from the language.

### 5.5.6. Indicator: Excessive references

This indicator tells that too many reference to potentially irrelevant external documents are provided in the text, and this might cause confusion, as in the following example:

- *Given rule 673.4 from document R.125 from Nuclear Commission, given decisions in document S.324-1999, given rule 329 from Std-425.126.334 [...]*

Learn PAd documents are different in format with respect to traditional paper documents. Indeed, they are going to be HTML pages with *links* to external documents (i.e., the references in the previous sentence are actually links). Therefore, this indicator in the case of Learn PAd computes the number of external links, and tells whether the overall number of links in a Learn PAd page is greater than a given threshold $\tau$. The the following rule checks the excessive references indicator.

**RULE 1:** $N$ = number of <a> tags, $N < \tau$.

The HTML tag <a> indicates the presence of a link in the page, either to another page or in the same page. At this stage, we have set $\tau = 5$. This is an arbitrarily chosen constraint that can be tailored by the Guidelines Manager of Learn PAd. If the rule is violated, this indicator is set to Defective.

**Recommendation:** Do not refer more than $\tau$ external documents. The reader might be confused. Refer only relevant external documents.
**Language Dependency:** this indicator is independent from the language.

[7] https://goo.gl/F86N3R

## 5.6. Quality Attribute: Completeness

In general, completeness of a document is a vague and hardly measurable concept, which expresses the idea that every content that is needed appears in the document. To our knowledge, no metric has been defined for PA documents that expresses the degree of completeness of a procedure description, as the ones that Learn PAd will host. Instead, in requirements engineering, measures of completeness have been provided, which checks the completeness of a document with respect to the input documents [26] – in this case we speak about *backward* completeness – with respect to a specification [21] – *forward* completeness – and with respect to a given template [73] – *internal* completeness. Here, we will refer to this latter notion of completeness, which we consider applicable also in the context of PA documents. Hence, this quality attribute tells how many of the required fields of a given template are covered.

To compute a quality measure for this quality attribute we define the following formula:

**Quality Measure**. $Q_{COM}$ = Number of fields with content/Total number of fields.

In our case, we refer to the **NL Content Template** described below. It is worth noting that part of the fields defined below will be part of the Static Wiki Page (identified with S), and others will be part of the Collaborative Wiki Page (identified with C). Therefore, the actual computation of this quality measure will vary depending on the type of page that is checked, since fields are different for each page. All the fields have been defined to address specific needs of civil servants that emerged from our questionnaire. Among them, the reader can see the field Examples, the filed FAQ, as well as the list of involved actors and people to contact in case of problems. The template is as follows:

- **Headline [C]:** a short title describing the content. The title can be the name of the BP Model or entity described.
- **Source Documents [S]:** identifiers of norms, regulations or any other document that give prescriptions or define the content from which the BP model, or entity has been derived.
- **Reference Documents [S]:** identifiers of norms, regulations or any other document that might have an impact on the current description.
- **Glossary [C]:** list of definitions that are useful to understand the NL Content. (preferably a link to a central glossary)
- **Context [C]:** a brief overview of the information that might be useful for a reader to understand the current BP model or entity described.
- **Summary [S]:** brief summary of the BP Model or entity described.
- **Motivation [S]:** the higher-level objective or justification of the BP model or entity described.
- **Intended readership [C]:** type of roles that should read this NL content.
- **Involved actors [C]:** actors (e.g., people, offices, authorities, etc.) that are involved in the BP Model or entity described.(should be in the Organisational Model)
- **Input documents [S]:** documents used as input for the current BP Model or entity described, if any. (should be in the Document Model and BPMN)
- **Output documents [S]:** documents produced by the current BP Model or entity described, if any. (should be in the Document Model and BPMN)
- **Required tools [S]:** software or hardware tools to be used to perform the process associate to the current BP Model or entity, if any. (should be covered by the IT system model)

- **Description [C]:** actual description of the BP Model or entity, expressed in terms of instructions or rules to be performed by any of the actors involved in the BP Model.
- **Examples/Experiences [C]:** list of real-world examples to practically describe the BP Model or entity (preferably expressed with links to other wiki pages).
- **What to do in case of failures [C]:** suggestions of possible alternative choices to take if something goes wrong while performing the process associated to the current BP Model or entity (preferably expressed with links to other wiki pages).
- **Contacts of involved offices [S]:** name, phone number and e-mail of the offices involved in the process.
- **Contacts of experts [S]:** name, phone number and e-mail of the BP Model experts to contact to ask for clarifications.
- **FAQ [C]:** list of frequently asked questions associated to the current BP Model or entity (preferably expressed with links to other wiki pages).

Both pages will be designed to include the fields of the template, with some differences. Static pages will embed the template as predefined fields. Therefore, the Content Manager is required to fill all the fields of the template in order to have 100% quality. Instead, Collaborative pages will include the titles of the fields, together with the descriptions provided above, embedded in the text and editable. Therefore, some fields can be deleted or adjusted according to the contributor's needs. The idea is that a contributor (i.e., a Content Manager or a Learner), is recommended to use the fields, but she/he is not constrained. Indeed, we conjecture that, at this stage, the actual content of Learn PAd wiki pages cannot be foreseen, and some flexibility shall be given to the users to encourage them to give contributions. However, also in the case of the Collaborative page, we will check the presence of text alongside each field of the pre-defined template, and compute the quality measure as defined above.

**Recommendation:** The field <field_name> appears to be without content. Please provide additional information.
**Language Dependency:** this indicator is independent from the language.

## 5.7. Quality Attribute: Correctness

The *correctness* quality attribute defines the degree of grammatical correctness of a NL description in Learn PAd. Hence, in this case, the quality attribute maps is equivalent to the indicator. Grammatical correctness is a fluid concept that evolves according to the evolution of a language and its grammar. Therefore, in our context, we have decided to give a more operational definition of correctness (i.e., a text is correct, if a grammar checker does not find any defect). To this end, we use a set of prescriptive rules, which are embedded in a tool, namely Language Tool[8], which has the advantage of embedding grammar checks that can be extended with the contributions of the user community. Therefore, as the grammar of a language evolves, we expect to easily plug additional rules – or remove old ones –, so that the computed degree of correctness of a sentence is up-to-date with the rules of language. As for simplicity, non-ambiguity and content clarity, we define the following quality measure for this quality attribute.

**Quality measure.** $Q_{COR}$ = 1 - (Number of defective sentences / Total number of sentences).

A sentence is considered defective if it has *at least* one grammatical error according to the Language Tool checker. In other terms, sentences that have only one grammatical error will count like sentences that have more than one.

[8] https://www.languagetool.org

**Recommendation:** specific recommendations for each type of error are directly imported from Language Tools.
**Language Dependency:** this indicator is dependent from the language. However, Language Tool offers extension for Italian, German, French and other languages.

## 5.8. Preliminary Evaluation

We have performed a preliminary evaluation of part of the rules and algorithms that we have defined for Learn PAd. The objective of this preliminary evaluation was understanding whether the strategies that we have defined can be considered suitable for the quality checker envisioned in Learn PAd. To understand if such strategies are suitable, we consider if they raise an amount of warnings that can be acceptable for a contributor. Indeed, as explained in Sect. 5.1, the defined strategies are oriented to highlight as much potential defects as possible. However, such potential defects might be perceived as false defects by the contributor. If the amount of such false defects is tolerable, the contributor can go through the warnings and ignore them. In case the amount of defects is overwhelming, the contributor is likely to ignore any output coming from the Content Analysis component, which will be considered too restrictive. Of course, the acceptability of a defect depends both on the type of defect and on the type of strategy adopted to identify it. Therefore, for each strategy, we will provide arguments to tell whether the strategy defined is acceptable or not in the Learn PAd context.

At this stage, we provided prototypical implementations for 14 rules and two algorithms. We have been able to experiment these strategies on a data-set composed of 23 public administration documents in textual format (1234 sentences in total). The data-set was initially defined for feedback-based quality evaluation, and its rationale and history is provided in Sect. 6. At this stage, we did not implement any indicator associated to the Presentation Clarity and Completeness attributes. Indeed, to check the former, we should have had an appropriate data-set with HTML formatting (our data-set is in textual format), and, to check the latter, Wiki documents should be provided in Learn PAd– it would not make sense to check documents that do not conform at all to the given template.

Table 5.1 summarises the results obtained for the different strategies. Appropriate discussions are provided for each quality attribute, and indicator, considered.

### 5.8.1. Simplicity

**Excessive length** *Excessive length* of sentences leads to 27% of defective sentences. This implies that, given a document of 54 sentences – the average length of our documents –, about 16 sentences are too long. We argue that this amount of warnings is tolerable for an editor, also taking into account that the threshold on the length of sentences comes from established official sources, as the UK government [64].

**Juridical jargon** For *juridical jargon*, we have 22% defective sentences, hence, 12 sentences for each document in average. Again, we consider this amount of warnings acceptable, although, by reviewing the output of our prototypical implementation, we see that some potential false positive cases are issued. Indeed, terms such as “acknowledgment” , “answer”, or “decision” are considered juridical jargon, since in legal writing they have a domain specific meaning. However, in procedure descriptions, such terms are not necessarily used with the juridical meaning. Hence, to limit the amount of warnings, the list of legal terms employed in our prototype has to be pruned from these cases.

**Difficult jargon** Concerning *difficult jargon* we have applied two rules. The first rule produces 76% defective sentences (41 sentences in average). We argue that this amount of warnings might be hardly tolerable for an editor. Moreover, considering that such rule has been defined by using the list of $5,000$ most frequent – and hence, *easy* – terms, which is not specific for the PA domain, we argue that most

of the cases can be perceived as false positives. Instead, the second rule produces 38% defective sentences (21 sentences in average). This is still a considerable amount of warnings. However, the list of terms used in this case comes from the Plain English Initiative, which is specifically focused in making official writing easier to read. Hence, such list is specific for the PA domain, and, though several warnings are issued, we argue that all such warnings are reasonable, and can help in improving the text.

### 5.8.2. Non-ambiguity

**Lexical ambiguity** *Lexical ambiguity* leads to 22% vagueness defects (12 sentences in average), and below 1% subjectivity and optionality defects (less than one sentence in average). These differences are mainly due to the larger amount of vague terms employed with respect to the other two categories. Given that the list of terms are well established from the literature, and given that 12 warnings can be considered acceptable, we argue that the three rule employed are suitable for Learn PAd.

**Syntactic ambiguity** Concerning *syntactic ambiguity*, we have 311 defects in total (25%, 13 sentences in average). As for the other indicators, we have defined our rules to identify the majority of the potential ambiguity cases, and, hence, many of these defects are actually false positives. However, as described in [72], the identification of those cases that are *nocuous* syntactic ambiguities – i.e., that are likely to lead to multiple interpretations – require appropriate machine learning algorithm, and is hard to be addressed with rule-based approaches, which might in principle lead to several false positive cases. Nevertheless, we argue that an editor can easily identify and discard those cases that are found not to be ambiguous. Moreover, this specific indicator is highly relevant according to our questionnaires. Event assuming that the majority of the cases found are false positive, the benefit given from the potential identification of a syntactic ambiguity is higher than the effort required to discard false positives.

**Pragmatic ambiguity** Our algorithm for *pragmatic ambiguity* detection, leads to 48% defective sentences (26 for each document in average). We have reviewed the output of one representative document of 50 sentences (“14 - sba become a cdc”, see Sect. 6). The document has 41 ambiguous names, and 22 sentences with ambiguous names (44%). So its length and degree of defects is close to the average of the other documents. We have checked the amount of actual pragmatic ambiguities in the file, and found that only 3 out of 22 sentences where actually false positive cases (14%). The amount of false positives increases if we look at the occurrences: 17 out of the 41 cases (41%). These false positive cases are mainly associated to cases where names are coupled with specifiers that have a numeric form (e.g., “504 program”), are coupled with an acronym (e.g., “CDC application”), or use the Saxon genitive (e.g., CDC’s compliance). These cases can be easily addressed by providing modifications to our rules. Other cases are associated to errors of the POS tagger, since some verbs are identified as plural names (e.g., “outlines”, “prints”). A more effective POS tagger can address there problems. The remaining false positive cases include terms such as “letter” or “copy”, which might be actually ambiguous from the pragmatic point of view – i.e., more clear specifications might be needed –, but, in the context of the document, it appears acceptable to leave them without further specification. Of course, developing an automated approach that distinguishes when it is acceptable to leave terms without specifiers requires further research. Another improvement that we foresee for a more effective pragmatic ambiguity detection, would be discarding from the ambiguous names all the cases where the name is defined in the glossary of the NL Content Template (see 5.6) of Learn PAd.

### 5.8.3. Content Clarity

**Actor unclear** Our approach for detecting sentences with an *unclear actor* leads to 27% defective sentences (about 15 sentences in average). We argue that this is an acceptable amount of warning for

| Quality Attribute | Indicator | Rule/Algorithm | Defective Sentences |
|---|---|---|---|
| Simplicity | Excessive length | RULE 1 | 335 |
| | Juridical jargon | RULE 1 | 272 |
| | Difficult jargon | RULE 1 | 934 |
| | | RULE 2 | 471 |
| Non-Ambiguity | Lexical ambiguity | RULE 1 (Vagueness) | 275 |
| | | RULE 2 (Subjectivity) | 11 |
| | | RULE 3 (Optionality) | 2 |
| | Syntactic ambiguity | RULE 1 (Coordination) | 100 |
| | | RULE 2 (Coordination) | 25 |
| | | RULE 3 (Anaphoric) | 147 |
| | | RULE 4 (Anaphoric) | 39 |
| | Pragmatic ambiguity | Algorithm | 598 |
| Content Clarity | Actor unclear | RULE 1 | 330 |
| | Unclear acronym | Algorithm | 208 |
| Correctness | Grammatical error | Language tool | 171 |

**Table 5.1: Preliminary evaluation of linguistic quality evaluation strategies.**

an editor, also considering that the usage of passive voice – which is considered in our rule for checking this indicator – is normally discouraged by public guidelines. False positive cases mainly occur with verbs coupled with past participle in adjective form (e.g, “The extent is limited”). However, we argue that an editor can easily discard these cases.

**Unclear acronym** Our algorithm for *unclear acronym* detection leads to 17% defective sentences (about 9 sentences in average). Overall, only 5 acronyms appeared to be defined, while 75 acronyms were undefined. As shown in Table 5.1 these acronyms appear in 208 sentences. Most of the undefined acronyms are expected to be known to the reader of the document. However, this cannot be established in advance, and, in addition, Learn PAd is oriented to *learners*, who may be new to the language and acronyms used in PA documents, and, in turn, in Learn PAd wiki pages. Therefore, we argue that our algorithm is suitable to be plugged in our quality checker.

### 5.8.4. Correctness

14% of the sentences included grammatical errors, according to Language Tools, which is the tool that we have been using for the *correctness* quality attribute and corresponding indicator. More specifically, the tool identifies 228 different errors in 171 sentences. Although this is a quite large amount of errors, especially if we consider that the evaluated documents are official documents, we argue that a grammar checker is indispensable to increase the quality of Learn PAd wiki pages.

# 6 Feedback-based Quality Assessment Strategies

## 6.1. Overview

This chapter describes the experiments performed to evaluate to which extent machine learning can be applied in Learn PAd to automatically check the quality of Wiki pages. To perform our evaluation we have first identified an appropriate group of documents describing procedures of the PA. Such documents can be regarded as potential content of the Wiki pages. Then, we have defined a set of defects to be checked by human assessors (referred in the following as annotators). Annotators have been selected within the consortium, and have been asked to read the documents and annotate them according to the defined defects. In practice, annotators had to highlight the part of the text that was considered defective, and to state the type of defect found. The annotated documents have been first used to train and evaluate a machine learning algorithm, namely Naive Bayes, whose goal was to evaluate if a sentence in a document was defective, and which was the defect found.

Given the poor results obtained with this approach, we have focused on a single defect, namely sentences expressing unclear time or deadline. We have evaluated all the time-related sentences in the original documents, and we have annotated them as defective/not defective. Then, we have trained and evaluated a Decision Tree algorithm. In this case, results were slightly better. However, we argue that focused research is still needed to profitably employ machine learning to automatically evaluate defects of PA procedures.

## 6.2. Experimental setup

### 6.2.1. Data-set Definition

We have selected a set of documents describing procedures of the PA. Such selection has been performed according to two steps: (1) a preliminary step, where we have surfed the web to identify websites that were including pointers to procedures; (2) a selection step, where we have selected a set of 23 documents – i.e., our data-set – from the websites.

In the preliminary step, we have selected a first set of 21 websites that were including procedure descriptions. Then, by inspecting the content of such websites, we have classified them according to two attributes, namely Degree and Level. Degree indicates the level of expertise required to understand the procedures in the websites, and Level indicates the level of detail of the procedure. The attributes were associated to the following values.

**Degree:**

- Legal (L): if the procedure requires the level of expertise of a lawyer
- Domain (D): if the procedure can be understood by a person belonging to the domain of the procedure
- Citizen (C): if the procedure can be understood by any type of citizen

**Level:**

| ID | Name | Source |
|---|---|---|
| 1 | fsd major event protocol | Austin Police Dept. |
| 2 | fsd purchase requests | Austin Police Dept. |
| 3 | fsd request for analysis | Austin Police Dept. |
| 4 | hdec review | NZ Health and Disability Ethics Committees |
| 5 | hse formal second opinion | UK Health & Safety Executive |
| 6 | hse intervention process | UK Health & Safety Executive |
| 7 | hse review of decisions | UK Health & Safety Executive |
| 8 | interreg project application | EU Interreg |
| 9 | leap enrollment | LEAP Academy |
| 10 | leap payroll | LEAP Academy |
| 11 | leap student withdrawal | LEAP Academy |
| 12 | ohra new information | Harvard T.H. Chan School |
| 13 | sba audit | US Small Business Administration |
| 14 | sba become a cdc | US Small Business Administration |
| 15 | sba computer matching procedure | US Small Business Administration |
| 16 | sba privacy act appellate procedure | US Small Business Administration |
| 17 | scot-gov bidding | Scottish Government |
| 18 | scot-gov construction procurement | Scottish Government |
| 19 | scot-gov risk assessment | Scottish Government |
| 20 | uk-gov get a divorce | UK Government |
| 21 | uk-gov legislative process | UK Government |
| 22 | us-gov legislative process | US Government |
| 23 | us-nuclear commission voting | US Nuclear Regulatory Commission |

**Table 6.1: List of documents composing our data-set.**

- Regulation (R): if the procedure is a high-level regulation
- Implementation (I): if the procedure is the translation of the regulation into a set of high-level steps and guidelines
- Manual (M): if the procedure is a detailed step-by-step manual

From such websites, we selected those that were classified with Degree = C or D, and Level = I or M. Indeed, we did not want to include in the documents that were requiring the expertise of a lawyer, as well as high-level regulations. Then, we have selected a set of 23 documents from the selected websites. The set of documents, together with the authority that released the document, is reported in Table 6.1. The 23 documents counts 1234 sentences in total, according to the definition of sentence given in Sect. 5.2.

The 23 documents have been used both as a base documents-set to be tagged by annotators, as explained in the next section, and as a reference data-set – without annotations – to be evaluated with rule-based approaches, as described in Sect. 5.8.

### 6.2.2. Annotations and Annotators

We selected a tagset of defects to be annotated based on two main criteria: firstly, we included defects that cannot be easily identified by manually crafted rules and that hence justify the use of machine learning. Secondly, we concentrated mainly on defects that are specific for descriptions of business processes – the rational being that other defects appearing in other types of documents have already been researched elsewhere. The defects checked with machine learning are those tagged with the Purple flag in our quality model of Fig. 4.1.

The tagset had two layers: we allowed annotators to either annotate a term or small phrase with a top-level tag “term unclear” or to annotate a whole sentence with the tag “sentence unclear”. On a second level, the nature of the defect could be specified. The following sub-categories could be annotated both for terms *and* sentences:

- “(condition of) action unclear”: it is unclear how or under which circumstances an activity should be performed
- “actor unclear”: the text does not specify who (e.g. which role) is responsible for executing a task
- “deadline / time interval unclear”: it is not specified when an activity may start or until when it needs to be finished
- “recipient unclear”: the text does not explain to whom exactly the output of a task should be handed over

In addition, the following two defects could be annotated on the sentence level:

- “instruction/rule unclear”: the sentence defines an unclear business rule that should guide the way an activity is performed
- “multiple instructions/rules per sentence”

There were several additional tags that are less specific for business processes. For terms, this comprised “undefined term/acronym” and “other unclear term”. For sentences, annotators could use the tags “syntax ambiguous”, “syntax too complex” and “other unclear sentence”. Annotators were also allowed to enter their own sub-categories.

For each defect, we provided a short explanation and an example from our data set as a guidance for annotators. 17 Annotators were recruited from the whole Learn PAd consortium and all documents from the data set were annotated by at least two annotators, some by three or four. In total, the annotators detected 993 defects, some of which are, however, overlapping. Table 6.2 shows how many defects were identified for the different tags.

| **Defect** | **Frequency** |
|---|---|
| (condition of) action unclear | 45 |
| actor unclear | 56 |
| deadline/time interval unclear | 47 |
| recipient unclear | 18 |
| instruction/rule unclear | 36 |
| multiple instructions/rules per sentence | 10 |
| unknown/undefined term/acronym/abbreviation | 649 |
| other unclear term | 52 |
| syntax ambiguous | 6 |
| syntax too complex | 7 |
| other unclear sentence | 10 |
| others | 57 |

**Table 6.2: Number of annotations per defect.**

The defects above the horizontal line are specific to business processes. The frequency of the defects is another criterion for focusing our research: defects that are too rare can hardly be identified by machine learning since training data will be too scarce. Thus, we decided to exclude “recipient unclear” and “multiple instructions/rules per sentence” from our further research.

A first analysis revealed that annotator agreement on the data was very poor, i.e. the annotations made by different persons on the same document were barely overlapping. After talking to some of the annotators, we hypothesised that this was not caused by actual disagreement, but simply due to the fact that annotators tended to overlook defects that others spotted. That is, we came to believe that the annotations were actually complementary.

To test our hypothesis, we performed a simple experiment: we selected three documents and asked the two annotators that were responsible for each document to meet and discuss about each annotated defect that only one of them had annotated. The task was to decide whether a) the two annotators actually disagreed about the annotation or b) they agreed to remove the annotation (i.e. one convinced the other that it was not a defect) or c) both agreed to keep the annotation (i.e. one convinced the other that it was indeed a defect).

In the cases where annotators eventually disagreed, it seems necessary to keep the annotation since the disagreement proves that there is at least one person that still recognises a defect and hence a misunderstanding can result when applying the process description in PAs. Table 6.3 shows the frequency of the above-mentioned cases and summarises how many of the non-overlapping annotations would finally have to be removed – based on this rationale.

| Annotator pair | non-shared annotations | disagree | agree to remove | agree to keep | kept | %kept |
|---|---|---|---|---|---|---|
| Pair 1 | 22 | 4 | 3 | 15 | 19 | 84% |
| Pair 2 | 21 | 0 | 0 | 21 | 21 | 100% |

**Table 6.3: Number of (dis-)agreements resulting from annotator discussions.**

The numbers rather strongly confirm our hypothesis. Thus, we created a final annotated document set by simply uniting the sets of all annotations made by all of our annotators. In that final document set, we expanded “term unclear” annotations to cover the whole sentence that was affected, such that all annotations of the final document set were on the sentence level. We call this final document set our *gold standard*.

### 6.2.3. Evaluation Measures

Given the final annotated document set, our goal was to find a classifier that would be able to predict whether a sentence was defective. In order to evaluate the effectiveness of a classifier, several standard measures exist in the literature. The notions of true/false positives and true/false negatives play an important role for most measures: a “positive” is a sentence that was predicted by the classifier to be a defect, whereas a “negative” is a sentence that the classifier considered non-defective. Whether the positive or negative predictions are “true” or “false” is assessed based on the human annotations in the gold standard. For instance, a false positive is a sentence that annotators did not consider defective whereas the classifier predicted a defect. Figure 6.1 depicts this situation.

Given these definitions, we can enumerate the standard evaluation measures:

- Accuracy is the percentage of all predictions that were correct
- Precision describes how many of the predicted defects (positives) are actual defects: $P = \frac{\#TP}{\#TP+\#FP}$
- Recall describes how many of the actual defects were predicted: $R = \frac{\#TP}{\#TP+\#FN}$
- The F-measure is the harmonic mean of precision and recall: $F = \frac{2PR}{P+R}$

For the experiments described below, we will report these standard measures. However, the standard measures give equal weight to both types of mistakes – i.e. false positives and false negatives – that

| | Actual Class | | |
|---|---|---|---|
| Predicted Class | | Defect | No defect |
| | Defect | TP | FP |
| | No Defect | FN | TN |

**Figure 6.1: A confusion matrix with true/false positives/negatives.**

a classifier can make. In practice, the impact of a false positive can be substantially different from the impact of a false negative.

In our case, a false positive will result in an unnecessary warning that a responsible person (who is in charge of writing the process description) has to inspect and dismiss. A false negative, on the other hand, results in unclear descriptions – in the best case, the defect is spotted during execution and a civil servant has to clarify how to proceed. In the worst case, wrong assumptions are made regarding the unclear part and tasks are consequently executed in an incorrect way. Hence, a false negative has a more severe impact than a false positive.

One way to take this difference into account is to use a version of the F-measure that places greater emphasis on recall (recall grows when the number of false negatives decreases). Another option is a cost-based evaluation where one estimates the negative impact (cost) caused be each classifier decision. Figure 6.2 depicts the costs that we estimate to arise for our scenario.

| | Actual Class | | |
|---|---|---|---|
| Predicted Class | | Defect | No Defect |
| | Defect | C | C |
| | No Defect | NC | 0 |

**Figure 6.2: A cost matrix for prediction of defects.**

Whenever the classifier predicts a defect, a warning is raised and a responsible person has to inspect the corresponding sentence. We assume that this causes an average loss of time (i.e. cost) of $C$ minutes – the responsible has to re-consider the formulation of the sentence and sometimes possibly to clarify the situation.

When the classifier fails to identify a defective sentence (i.e. when a false negative occurs), the sentence remains in the final process description. We optimistically assume that civil servants who read the sentence, will not work on false assumptions, but will always spot the unclarity and attempt to clarify. Such clarification – that may involve speaking to colleagues or consulting other sources – causes a loss of time (cost) that we estimate to be at least as high as the above-mentioned cost $C$

for handling raised warnings. Process descriptions usually have to be read – at leat once – by all civil servants in a PA that are regularly involved in process execution. If the number of civil servants is $N$, then – based on the above arguments – the cost of a false negative is at least $NC$. In our evaluation runs, we used $C = 1$ and $N = 10$, i.e. we assumed a situation with 10 civil servants working based on a common process description.

## 6.3. Results

### 6.3.1. Experimental Evaluation - Multiple Indicators

In a first series of experiments, we attempted to get a first impression of how far one can get with a standard text categorisation approach. The standard approach we used consists in using all words that occurred in at least two sentences of the data set as features to describe a sentence. Other features are not used in this approach and order of words is ignored – this is called a "bag of words" representation. More precisely, each sentence $s$ is transformed into a vector $\vec{s} = (w_1, ..., w_n)$ where the above-mentioned features (words) form the basis of a vector space and the weight $w_i$ in vector $\vec{s}$ indicates to what degree the word $i$ represents the content of sentence $s$. We used a tf.idf term weighting (see e.g. [58]).

We performed two experiments: one with a data set where all annotated defects were treated the same and tagged with "defect" whereas all other sentences were tagged with "none". We will refer to this as the "two-class data set" or "two-class problem". In the other experiment, we worked with the original annotations, resulting in 12 different classes that the classifier should learn to distinguish.

After some experimentation with available implementations, we concluded that the Naive Bayes classifier delivered the best results. Table 6.4 shows the results for the two-class problem obtained with a 10-fold cross-validation. As a baseline, we used the classifier that always predicts "none", i.e. it will not identify a single defect. This baseline is reasonable since it describes the current situation, i.e. the situation where there is no quality assurance algorithm and hence all sentences remain (unchecked) in the process description.

| Measure | Baseline | Naive Bayes |
|---|---|---|
| Accuracy | 72.8% | 69.1% |
| Precision (of class "Defect") | 1 | 0.45 |
| Recall (of class "Defect") | 0 | 0.60 |
| F1 (for class "Defect") | 0 | 0.51 |
| ROC area | 0.50 | 0.71 |
| Total cost | 3520 | 1678 |

**Table 6.4: Classification results for the two-class problem.**

Although Naive Bayes clearly outperforms the baseline in terms of all measures except accuracy, the absolute numbers are not satisfactory: a recall of 60% means that 40% of defects are not identified, causing still a substantial cost of almost 1700.

The picture is – as one might expect – not much better for the mult-class problem, see Table 6.5.

We performed a qualitative analysis of the results by inspecting the model that the classifier learned from the two-class data. In particular, for Naive Bayes, the model comprises the conditional probabilities $P(w|c)$ of a word $w$ occurring within sentences of class $c$. We sorted words by the ratio $\frac{P(w|Defect)}{P(w|none)}$ in order to filter out the words – from the top of the sorted list – that have a much higher probability of occurring in a defective sentence than of occurring in non-defective ones.

The analysis was mostly inconclusive: it revealed that some acronyms are causing unclarity – the other words on top of the list often originated from one particular document. This leads us to believe

| Measure | Baseline | Naive Bayes |
|---|---|---|
| Accuracy | 72.8% | 47.1% |
| Precision (average) | 0.53 | 0.66 |
| Recall (average) | 0.73 | 0.47 |
| F1 (average) | 0.61 | 0.54 |
| ROC area | 0.67 | 0.71 |
| Total cost | 3520 | 1505 |

**Table 6.5: Classification results for the multi-class problem. Precision, Recall and F are averaged over all classes, including "none".**

that the Naive Bayes approach has very likely been overtrained on the data, i.e. the model is fitted to closely to the specific characteristics of certain documents in the training data.

We therefore concluded that further investigation was needed, in particular in terms of feature engineering, i.e. to find better linguistic features than a bag of words.

### 6.3.2. Experimental Evaluation - Single Indicator

We decided to concentrate on the defect "deadline/time interval unclear". In theory, a sentence can exhibit such a defect even when it does not contain any time-related information – the defect might be precisely that such information is missing. In practice, however, most annotated defects of this kind (35 out of 47) were sentences that did contain time-related information. In addition, we consider it a very hard task to learn automatically which characteristics of a sentence might indicate that time-related information is needed.

We therefore propose to apply a two-step procedure as depicted in Figure 6.3: first, we apply a rule-based approach – based on certain keywords – for deciding whether a sentence contains time-related information (we call this a "time sentence" from now on), then we apply a classifier that decides whether a given time sentence has a "deadline/time interval unclear" defect or not.

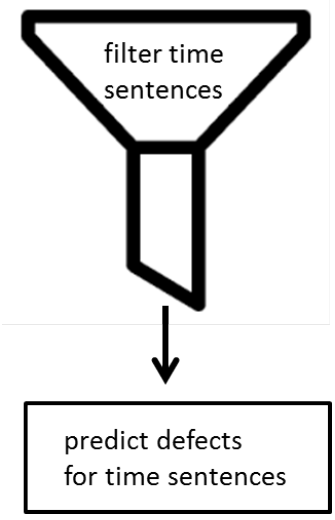



**Figure 6.3: Two-step approach for identifying time-related defects.**

Since our annotators identifed only 35 defects among time sentences, we suspected that they had overlooked some defects. We therefore applied the following approach to construct a new training set, specifically for analysis of time-related defects:

1) We ran the filter over the whole data set, resulting in 207 time sentences from our corpus.

2) for each identified time sentence, the two researchers involved in this study annotated the whole sentence as defective, non-defective or not time-related

3) we then discussed and reached an agreement on each sentence. This was not an easy process and it revealed that a better model of what really constitutes a defect might be needed. In the end, we kept 49 non-defective sentences and 85 defective ones.

After a qualitative analysis of the defective sentences – and based on the experiences gained during the discussions over the annotations – we hypothesised the following features to be useful in identifying defects:

- **Length of the sentence**: Longer sentences are likely to include complex time reference, or to create unclarity in the interpretation of a specific time. Indeed, unclear sentences in our set had an average of 21 words per sentence (95 chars), while clear sentences are around 25 words (125 chars).
- **Presence of vague terms**: When these terms appear in a sentence with dates, the time interval is likely to be unclear. We compiled a list of vague time-related terms and expressions (e.g. “shortly before”, “in advance” etc.) and used a lookup step to annotate them in the text.
- **Presence of coordination elements**, such as OR and AND: When coordination is involved, this sometimes leads to expressions that define multiple deadlines, which may result unclear. Example: “The FSO procedure should be completed within two weeks and in any case by the date by which an acceptance decision has to be made.”
- **Number of time-related adverbs** in the sentence: The unclear time-related terms are representative only for our data-set. However, they are all adverbs (e.g. “earlier”, “regularly” or “eventually”), and a high number of them might indicate that a sentence identifies a time with some discretion of the reader.

We these features, we attempted to learn a decision tree. Figure 6.4 shows the simple decision tree that resulted from the analysis (here, the class label “no” means that the sentence is defective). When expressed in words, it means that a sentence will be classified as a defect, either when it contains a vague term or if it contains a coordination.

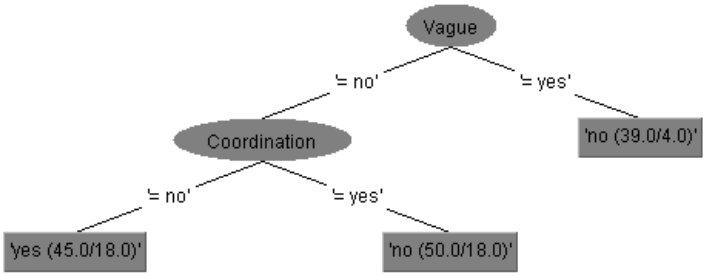



**Figure 6.4: The simple decision tree learned with the initial set of features.**

In a next step, we analysed the results in a qualitative way: we looked at both false positive and false negatives to understand how we could improve the simple decision tree. We observed mainly two things:

- Many false positives were generated because a coordination or vague term occurred in a part of the sentence that was not time-related. For example, in the sentence “You can get a divorce if you have been married at least a year **and** your relationship has permanently broken down.”, the conjunction “and” connects two proper sentences, but not two time-related pieces of information. The time-related information (“at least a year”) is actually clear.
- Some false negatives resulted from the use of terms that describe a periodicity by means of a longer period of time. For instance, in a sentence starting with “**Every year**, the payroll is checked to ensure...”, unclarity results because the sentence does not say exactly at which time within the year the payroll should be checked.

To counter the first problem, we introduced the notion of a “time chunk”, which we defined as *the largest noun or prepositional phrase within a sentence that includes a time information*. We built a rule for annotating time chunks within sentences, based on the syntax trees generated by parsers that are available in GATE.

We then annotated sentences with two new features, namely *vague time chunk* – when the sentence contained a vague term within a time chunk – and *coordinated time chunk* – when the sentence contained a conjunction OR or AND within a time chunk.

For the second problem, we created a new list of terms, namely those indicating longer periods of time (“every year”, “quarterly” etc.) and annotated sentences with the new feature *period* when they contained a word from the list.

We learned another decision tree model with the new features – interestingly, the resulting (pruned) tree does not use the features that are based on time chunks, see Figure 6.5. It does, however, use the feature *period*. We discovered rather quickly that the time chunk features were not useful simply because many of the syntax trees produced by the GATE parsers were not correct. Better parsers are available, but take too long to be reasonably used for such experiments.

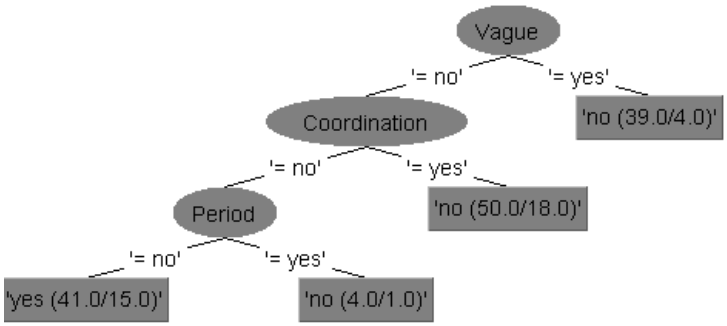



**Figure 6.5: The decision tree learned with the initial, plus the new set of features.**

Table 6.6 summarises the results that we obtained with the baseline (predicting no defect for each sentence), with the simple tree from Figure 6.4 and the tree we learned on the data with the new features (i.e. including the *period* feature).

One can observe that there is only marginal improvement of the “Period” tree over the simple tree. The cost is substantially lower for both trees when compared to the baseline and the recall figure indicates that these simple decision trees are able to find 80% of the annotated defects.

## 6.4. Conclusions and Future Work

In our machine learning experiments, we came to the following conclusions:

| Measure | Baseline | Simple Tree | Period Tree |
|---|---|---|---|
| Accuracy | 36.6% | 70.1% | 70.1% |
| Precision (of class “defect”) | 1 | 0.70 | 0.75 |
| Recall (of class “defect”) | 0 | 0.79 | 0.8 |
| F1 (of class “defect”) | 0 | 0.77 | 0.77 |
| ROC area | 0.5 | 0.68 | 0.67 |
| Total cost | 899 | 269 | 261 |

**Table 6.6: Classification results for deadline/time defects, including the baseline, a simple tree and a tree learned with additional features, especially the new feature *period*.**

- In the second set of experiments, when tagging sentences manually in order to create a gold standard, we discovered that we might need a better model of what constitutes a time-related defect. Having very clear guidelines for annotation will also make the feature engineering for machine learning easier. We have started working on a model of time defects, but this is still work in progress and will have to be deepened in the future.
- With a standard text categorisation approach applied to a data set with all defects annotated, one can reach classification results that are substantially better than the current baseline (in terms of cost). But on an absolute scale the results are rather poor (discovering only 60% of all defects) and the learned models suggest that there is a high risk of overtraining.
- When putting more effort into feature engineering for particular defects, the results can be improved. For instance, our learned decision trees for detection of time-related defects discovered 80% of all manually annotated defects. Results can be possibly further improved with the use of better syntax parsers – something to be evaluated in the future. The trees contain simple features such as the presence of a coordination (AND, OR) or the presence of vague time-related terms. Eventually, we may just implement a simple rule in GATE as a result of our learnings.

# 7 Architectural View on Content Analysis Component

In this chapter we provide a description of the architecture of the Content Analysis component of the Learn PAd platform. The component includes the quality assessment strategies defined in Chapter 5.

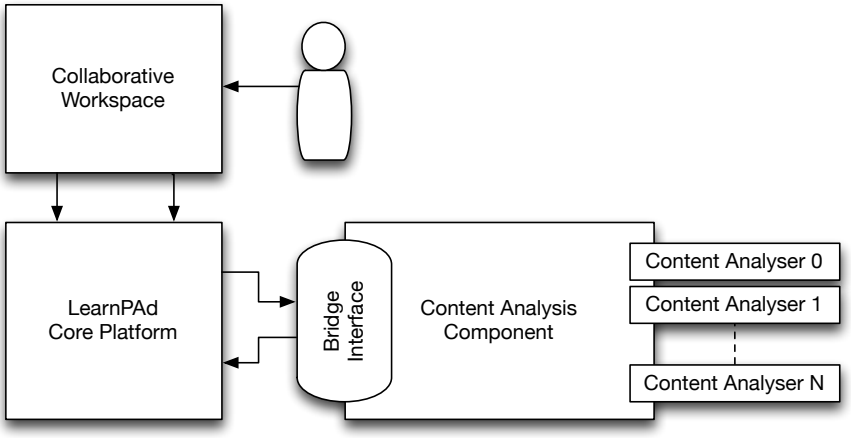



**Figure 7.1: Architecture of the Content Analysis component in the context of the Learn PAd platform.**

Fig. 7.1 depicts the Content Analysis component in the context of the Learn PAd platform. Such component interacts with the Learn PAd Core Platform through a unique interface, named Bridge Interface. The Learn PAd Core Platform mediates the interaction with the Collaborative Workspace component, which includes the wiki where the pages to be analysed are stored.

From a dynamic point of view, as already outlined in Fig. 2.4, Chapter 2, the user (i.e., the Content Manager) asks the system to validate the NL content of a wiki page, by interacting with the Collaborative Workspace component. The Content Analysis component receives the request through the mediation of the Learn PAd Core Platform. Together with the request, the component receives an XML file including the content to be analysed. Then, the component performs the different analysis for the different quality attributes (i.e., Simplicity, Non-ambiguity, Content Clarity, Presentation Clarity, Completeness, Correctness), according to the different indicators. A Content Analyser is defined for each indicator, which computes the value of the indicator as outlined in the different sections of Chapter 5. Then, the Content Analysis component aggregates the values of each indicator into quality measures – one for each quality attribute –, and stores the quality measure and the annotations in an XML file. For each quality attribute, a different XML file is sent back to the Collaborative Workspace.

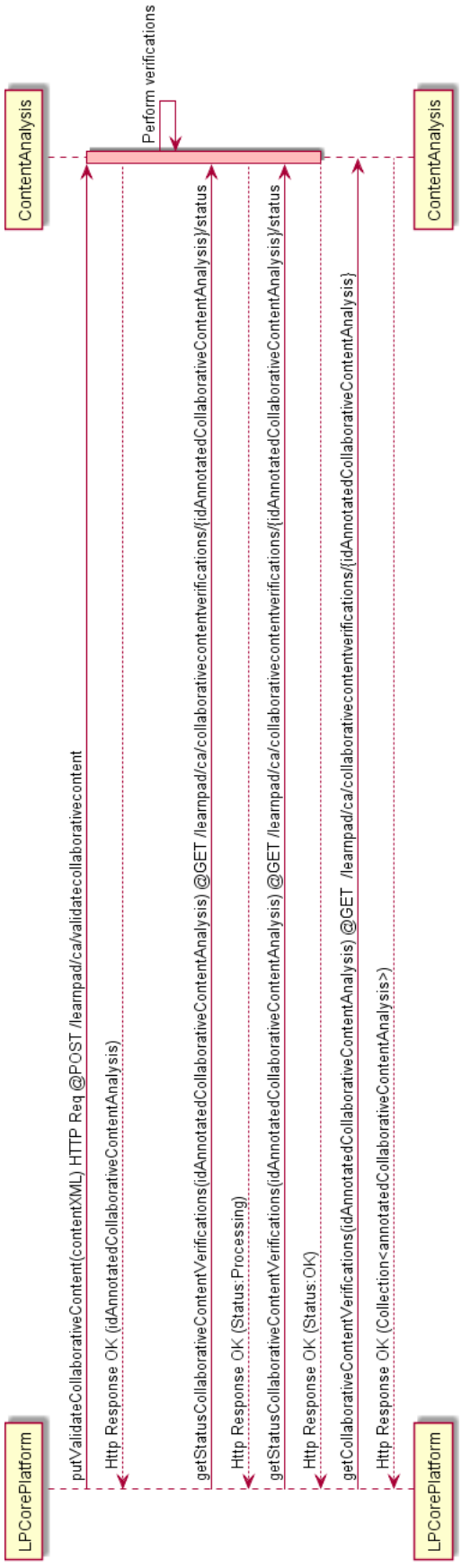



**Figure 7.2: Interaction between the Content Analysis component and the Learn PAd Core Platform.**

Fig. 7.2 depicts the interaction between the Content Analysis component and the Learn PAd Core Platform (LPCorePlatform in the figure). In particular, it depicts the interaction when the validation is required for the Collaborative Page (see Chapter 5). A similar interaction occur when validation is required for the Static Page.

The two components interact by means of REST interfaces. The Content Analysis component first receives the XML including the content to be analysed. Fig. 7.3 presents the structure of such content with an example. It is worth noting that the content is repeated both in its plain form, and in its HTML format. Indeed, the indicators of the Presentation Clarity quality attribute need the HTML format to be measured. All the other indicators require solely the plain text. We have decided to duplicate the information, since, after the analysis, most of the indicators are associated with *annotations* on the plain text. The same annotations performed on the HTML text would have required higher overhead to be computed – i.e., first we should have discarded the HTML, and then, after the analysis, annotations should have been projected on the HTML. At the same time, HTML was required for the Presentation Clarity indicators, and none of these indicators consider the textual content, but only the HTML tags. Hence, duplication was seen as the most effective choice to address the needs of the different Content Analysers, and, at the same time, reduce the overhead on the Content Analysis component.

```
<?xml version="1.0" encoding="UTF-8"?>
<CollaborativeContentAnalysis  language="english">
 <CollaborativeContent id="1213">
  <Title>Title of Documents</Title>
  <ContentPlain>
   The document shall be sent to the proper
   authorities as soon as possible
  </ContentPlain>
 <ContentHTML>
   <p>
   The <b>document </b> shall be sent to the proper
   authorities as soon as possible
   </p>
  </ContentHTML>
 </CollaborativeContent>
 <QualityCriteria simplicity="true" non_ambiguity="true"
  content_clarity="true" presentation_clarity="true" completeness="true"
  correctness="true" />
</CollaborativeContentAnalysis>
```

**Figure 7.3: Input XML for the Content Analysis component.**

After the different Content Analysers have performed their analysis, an XML is returned for each one of the quality attributes. The structure of such XML is presented in the example of Fig. 7.4. The XML is the one associated to the Non-ambiguity quality attribute. A node is associated to each annotation in the text, and such node is used as a reference to provide an appropriate recommendation. The overall quality and overall quality measure for the Non-ambiguity attribute are computed as outlined in Sect. 4.2, and Sect. 5.3, respectively, and stored in the XML file that is returned to the Collaborative Workspace through the Learn PAd Core Platform. The Collaborative Workspace will use the content of such XML file to display the content of the Quality Evaluation Page and Inspection Page, as described in Chapter 2.

```
<?xml version="1.0" encoding="UTF-8"?>
<annotatedCollaborativeContentAnalysis id="9324" type="Non Ambiguity">
 <CollaborativeContent id="1213">
  <Title>Title of Documents</Title>
  <Content>
  <Node id="147"/> The document shall be sent to
   the <Node id="30"/>proper<Node id="36"/> authorities
   <Node id="48"/>as soon as possible<Node id="66"/>
  </Content>
 </CollaborativeContent>
 <Annotations>
  <Annotation id="0"
   type="Non ambiguity"
   StartNode="30" EndNode="36"
   recommendation="The term proper is vague"/>
  <Annotation id="2"
   type="Non ambiguity"
   StartNode="48" EndNode="66"
   recommendation="The as soon as possible is vague"/>
 </Annotations>
 <OverallQuality>BAD</OverallQuality>
 <OverallQualityMeasure>50%</OverallQualityMeasure>
 <OverallRecommendations>
  Use more precise terms!
 </OverallRecommendations>
</annotatedCollaborativeContentAnalysis>
```

**Figure 7.4: Output XML for the Non-ambiguity quality attribute.**

# 8 Conclusions and Future Work

In this deliverable, we presented the first results of WP4 concerning the analysis of the NL content of the wiki pages that provide information about BP models. The contribution of this deliverable is manifold: (1) it provides an in-depth domain analysis on strategies for linguistic quality evaluation, and on peculiar qualities required by PA procedure descriptions. Such domain analysis includes a literature review, a set of interviews with civil servants, and a questionnaire delivered to civil servants (Chapter 3); (2) a set of guidelines for editing the content of Learn PAd wiki pages, and PA procedure descriptions in general, together with a quality model for PA procedures and wiki pages (Chapter 4); (3) a set of strategies to compute the quality of wiki pages and a preliminary evaluation of the applicability of such strategies in a real world data-set (Chapter 5); (4) an exploratory study on the application of machine learning techniques for the evaluation of the quality of PA procedure descriptions (Chapter 6); (5) a description of the Content Analysis component that will include the strategies for NL content quality evaluation, in the context of the Learn PAd platform (Chapter 7).

Future work in the context of the Learn PAd project includes the implementation of the strategies defined in Chapter 5 as part of the Content Analysis component. Future work beyond the project include the development of additional quality checks to support the analysis of PA documents. Moreover, we aim to conduct further studies on the potential usage of machine learning techniques to detect defects in PA documents. To this end, after the deployment of the Learn PAd platform, we plan to leverage the feedback of the Learn PAd users on the NL content provided through the platform.

CC BY

# Bibliography


[1] Eneko Agirre and Philip Glenny Edmonds. *Word sense disambiguation: Algorithms and applications*, volume 33. Springer Science & Business Media, 2007.

[2] Sandra Aluisio, Lucia Specia, Caroline Gasperin, and Carolina Scarton. Readability assessment for text simplification. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9. Association for Computational Linguistics, 2010.

[3] Vincenzo Ambriola and Vincenzo Gervasi. On the systematic analysis of natural language requirements with Circe. *ASE*, 13, 2006.

[4] Satanjeev Banerjee and Ted Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *IJCAI*, volume 3, pages 805–810, 2003.

[5] Daniel Berry, Ricardo Gacitua, Pete Sawyer, and Sri Fatimah Tjong. The case for dumb requirements engineering tools. In *Requirements Engineering: Foundation for Software Quality*, pages 211–217. Springer, 2012.

[6] Daniel M. Berry, E. Kamsties, and Michael M. Krieger. From contract drafting to software specification: Linguistic sources of ambiguity, 2003.

[7] Daniel M. Berry and Erik Kamsties. The syntactically dangerous all and plural in specifications. *IEEE Software*, 22(1):55–57, 2005.

[8] Jeanne Sternlicht Chall and Edgar Dale. *Readability revisited: The new Dale-Chall readability formula*. Brookline Books, 1995.

[9] Francis Chantree, Bashar Nuseibeh, Anne N. De Roeck, and Alistair Willis. Identifying nocuous ambiguities in natural language requirements. In *Proc. of RE'06*, pages 56–65, 2006.

[10] Alessandro Cimatti, Marco Roveri, Angelo Susi, and Stefano Tonetta. Formalizing requirements with object models and temporal constraints. *SoSyM*, 10(2), 2011.

[11] Meri Coleman and Ta Lin Liau. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283, 1975.

[12] World Wide Web Consortium et al. Web content accessibility guidelines (wcag) 2.0. 2008.

[13] Michele A. Cortelazzo and Federica Pellegrino. 30 regole per scrivere testi amministrativi chiari. *Guida agli Enti Locali*, 20:XXV–XXXV, May 2002.

[14] M. Cutts. *The plain English guide*. Oxford University Press, 1996.

[15] Edgar Dale and Jeanne S Chall. The concept of readability. *Elementary English*, 26(1):19–26, 1949.

[16] Mark Davies. Word frequency data. http://www.wordfrequency.info/free.asp. Accessed: 1 Aug. 2015.

[17] Tullio De Mauro. *Il dizionario della lingua italiana*. Paravia, Torino, 2000.

[18] Felice Dell'Orletta, Simonetta Montemagni, and Giulia Venturi. Read-it: Assessing readability of italian texts with a view to text simplification. In *Proceedings of the second workshop on speech and language processing for assistive technologies*, pages 73–83. Association for Computational Linguistics, 2011.

[19] Felice DellOrletta, Simonetta Montemagni, and Giulia Venturi. Assessing document and sentence readability in less resourced languages and across textual genres. *International Journal of Applied Linguistics*, 165(2):163–193, 2014.

[20] Barbara DiCicco-Bloom and Benjamin F Crabtree. The qualitative research interview. *Medical education*, 40(4):314–321, 2006.

[21] Sergio Espana, Nelly Condori-Fernandez, Arturo Gonzalez, and Óscar Pastor. Evaluating the completeness and granularity of functional requirements specifications: a controlled experiment. In *Requirements Engineering Conference, 2009. RE'09. 17th IEEE International*, pages 161–170. IEEE, 2009.

[22] European Commission. How to write clearly. http://ec.europa.eu/translation/writing/clear_writing/how_to_write_clearly_en.pdf. Accessed: 18 Sept. 2015.

[23] Alessandro Fantechi, Stefania Gnesi, Gioia Ristori, Michele Carenini, Massimo Vanocchi, and Paolo Moreschini. Assisting requirement formalization by means of natural language translation. *Form Method Syst Des*, 4(3), 1994.

[24] Lijun Feng, Noémie Elhadad, and Matt Huenerfauth. Cognitively motivated features for readability assessment. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 229–237. Association for Computational Linguistics, 2009.

[25] Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noémie Elhadad. A comparison of features for automatic readability assessment. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 276–284. Association for Computational Linguistics, 2010.

[26] Alessio Ferrari, Felice dellOrletta, Giorgio Oronzo Spagnolo, and Stefania Gnesi. Measuring and improving the completeness of natural language requirements. In *Requirements Engineering: Foundation for Software Quality*, pages 23–38. Springer, 2014.

[27] Alessio Ferrari and Stefania Gnesi. Using collective intelligence to detect pragmatic ambiguities. In *Requirements Engineering Conference (RE), 2012 20th IEEE International*, pages 191–200. IEEE, 2012.

[28] Alessio Ferrari and Stefania Gnesi. Using collective intelligence to detect pragmatic ambiguities. In *Proc. of RE'12*, pages 191–200, 2012.

[29] Alessio Ferrari, Giuseppe Lipari, Stefania Gnesi, and Giorgio O Spagnolo. Pragmatic ambiguity detection in natural language requirements. In *Artificial Intelligence for Requirements Engineering (AIRE), 2014 IEEE 1st International Workshop on*, pages 1–8. IEEE, 2014.

[30] Alessio Ferrari, Paola Spoletini, and Stefania Gnesi. Ambiguity as a resource to disclose tacit knowledge. In *IEEE RE'15 Conference*, pages 26–35, 2015.

[31] Valerio Franchina and Roberto Vacca. Adaptation of flesh readability index on a bilingual text written by the same author both in italian and english languages. *Linguaggi*, 3:47–49, 1986.

[32] Benedikt Gleich, Oliver Creighton, and Leonid Kof. Ambiguity detection: Towards a tool explaining ambiguity sources. In *Proc. of REFSQ'10*, volume 6182 of *LNCS*, pages 218–232. Springer, 2010.

[33] Stefania Gnesi, Giuseppe Lami, and Gianluca Trentanni. An automatic tool for the analysis of natural language requirements. *IJCSSE*, 20(1), 2005.

[34] R. Gunning. *The Technique of Clear Writing*. McGraw-Hill, 1952.

[35] Graeme Hirst. *Semantic interpretation and the resolution of ambiguity*. Cambridge University Press, 1992.

[36] Nancy Ide and Jean Véronis. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational linguistics*, 24(1):2–40, 1998.

[37] Kentaro Inui, Satomi Yamamoto, and Hiroko Inui. Corpus-based acquisition of sentence readability ranking models for deaf people. In *NLPRS*, pages 159–166, 2001.

[38] ITTIG-CNR and Accademia della Crusca. *Guida alla Redazione degli Atti Amministrativi: Regole e suggerimenti*. Istituto di teoria e tecniche dellinformazione giuridica del CNR, Firenze, 2011.

[39] J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, DTIC Document, 1975.

[40] Nadzeya Kiyavitskaya, Nicola Zeni, Luisa Mich, and Daniel M. Berry. Requirements for tools for ambiguity identification and measurement in natural language requirements specifications. In *Proc. of WER'07*, pages 197–206, 2007.

[41] Leonid Kof. From requirements documents to system models: A tool for interactive semi-automatic translation. In *Proc. of RE'10*, 2010.

[42] Yoong Keok Lee and Hwee Tou Ng. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 41–48. Association for Computational Linguistics, 2002.

[43] Raffaele Libertini, Pietro Mercatali, and Francesco Romano. *Come si scrive un atto amministrativo? Esperienze a confronto a partire da un questionario*. Istituto di teoria e tecniche dellinformazione giuridica del CNR, Firenze, 2015.

[44] P. Lucisano and M. E. Piemontese. Gulpease. una formula per la predizione della difficolta dei testi in lingua italiana. *Scuola e Citta*, 3:57–68, 1988.

[45] Yi Ma, Eric Fosler-Lussier, and Robert Lofthus. Ranking-based readability assessment for early primary children's literature. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 548–552. Association for Computational Linguistics, 2012.

[46] Aaron K Massey, Richard L Rutledge, Annie Anton, Peter P Swire, et al. Identifying and classifying ambiguity for regulatory requirements. In *Requirements Engineering Conference (RE), 2014 IEEE 22nd International*, pages 83–92. IEEE, 2014.

[47] G Harry McLaughlin. Smog grading: A new readability formula. *Journal of reading*, 12(8):639–646, 1969.

[48] L. Mich. NL-OOPS: from natural language to object oriented requirements using the natural language processing system LOLITA. *NLE*, 2(2):161–187, 1996.

[49] L. Mich and R. Garigliano. Ambiguity measures in requirements engineering. In *Proc. of ICS'00, 16th IFIP WCC*, pages 39–48, 2000.

[50] Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10, 2009.

[51] Roberto Navigli and Paola Velardi. Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(7):1075–1086, 2005.

[52] Ani Nenkova, Jieun Chae, Annie Louis, and Emily Pitler. Structural features for predicting the linguistic quality of text. In *Empirical methods in natural language generation*, pages 222–241. Springer, 2010.

[53] Hwee Tou Ng and Hian Beng Lee. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 40–47. Association for Computational Linguistics, 1996.

[54] Sarah E Petersen and Mari Ostendorf. A machine learning approach to reading level assessment. *Computer speech & language*, 23(1):89–106, 2009.

[55] Plain English Campaign. The A to Z of alternative words. http://www.plainenglish.co.uk/files/alternative.pdf.

[56] Dipartimento della Funzione Pubblica Presidenza del Consiglio Dei Ministri. Semplificazione del linguaggio dei testi amministrativi. *Gazzetta Ufficiale*, 141, June 2012.

[57] Janice Ginny Redish. *Letting Go of the Words: Writing Web Content that Works*. Elsevier, 2012.

[58] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[59] RJ Senter and EA Smith. Automated readability index. Technical report, DTIC Document, 1967.

[60] Luo Si and Jamie Callan. A statistical model for scientific readability. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 574–576. ACM, 2001.

[61] Mark Stevenson and Yorick Wilks. Word sense disambiguation. *The Oxford Handbook of Comp. Linguistics*, pages 249–265, 2003.

[62] Kumiko Tanaka-Ishii, Satoshi Tezuka, and Hiroshi Terada. Sorting texts by readability. *Computational Linguistics*, 36(2):203–227, 2010.

[63] Christopher R Trudeau. The public speaks: An empirical study of legal communication. *The Scribes J. Leg. Writing*, 14(2011-2012), 2012.

[64] UK Government. Content design: planning, writing and managing content. https://www.gov.uk/guidance/content-design/writing-for-gov-uk. Accessed: 1 Aug. 2015.

[65] UK Government. Content design: planning, writing and managing content. https://www.gov.uk/service-manual/user-centred-design/how-users-read.html. Accessed: 22 Sept. 2015.

[66] University of Sheffield,. Jape: Regular expressions over annotations. https://gate.ac.uk/sale/tao/splitch8.html. Accessed: 1 Aug. 2015.

[67] US Government. Federal plain language guidelines, revision 1. http://www.plainlanguage.gov/howto/guidelines/bigdoc/index.cfm, 2011.

[68] Jean Véronis. Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language*, 18(3):223–252, 2004.

[69] J.M. Williams and G.G. Colomb. *Style: Toward Clarity and Grace*. Chicago guides to writing, editing, and publishing. University of Chicago Press, 1995.

[70] Patrick Wilson. Situational relevance. *Information storage and retrieval*, 9(8):457–471, 1973.

[71] William M. Wilson, Linda H. Rosenberg, and Lawrence E. Hyatt. Automated analysis of requirement specifications. In *Proc. of ICSE'97*, pages 161–171, 1997.

[72] Hui Yang, Anne N. De Roeck, Vincenzo Gervasi, Alistair Willis, and Bashar Nuseibeh. Analysing anaphoric ambiguity in natural language requirements. *Requir. Eng.*, 16(3):163–189, 2011.

[73] Didar Zowghi and Vincenzo Gervasi. The three Cs of requirements: consistency, completeness, and correctness. In *International Workshop on Requirements Engineering: Foundations for Software Quality, Essen, Germany: Essener Informatik Beitiage*, pages 155–164, 2002.