A set of Nooj Grammar to verify laboratory data correctness

Francesca Parisi[1] and Maria Teresa Chiaravalloti[1]

[1] Institute of Informatics and Telematics, National Council of Research, Cosenza, Italy
francesca.parisi@iit.cnr.it
maria.chiaravalloti@iit.cnr.it

Abstract. Semantic interoperability in clinical processes is necessary to exchange meaningful information among healthcare facilities. Standardized classification and coding systems allow for meaningful information exchange. This paper aims to support the accuracy validation of mappings between local and standardized clinical content, through the construction of NooJ syntactical grammars for recognition of local linguistic forms and detection of data correctness level. In particular, this work deals with laboratory observations, which are identified by idiosyncratic codes and names by different facilities, thus creating issues in data exchange. The Logical Observation Identifiers Names and Codes (LOINC) is an international standard for uniquely identifying laboratory and clinical observations. Mapping local concepts to LOINC allows to create links among health data systems, even though it is a cost and time-consuming process. Beyond this, in Italy LOINC experts use to manually double check all the performed mappings to validate them. This has over time become a non-trivial task because of the dimension of laboratory catalogues and the growing adoption of LOINC. The aim of this work is realizing a NooJ grammar system to support LOINC experts in validating mappings between local tests and LOINC codes. We constructed syntactical grammars to recognize local linguistic forms and determine data accuracy, and the NooJ contextual constraints to identify the threshold of correctness of each mapping. The grammars created help LOINC experts in reducing the time required for mappings validation.

Keywords: NooJ, Semantic annotation, LOINC.

1 Introduction

Nowadays electronic healthcare systems provide advanced technological infrastructures to support data exchange and pooling. Nonetheless, an effective communication between systems is not yet fully realized as still many healthcare settings use idiosyncratic local codes and names to identify the same clinical concepts. Clinical classification and coding systems describe concepts through standardized labels identifying them through unique codes. Among them, LOINC [1] is the main standard for coding laboratory and clinical observations. Its use requires that local codes are mapped to standardized codes so that equivalent concepts are aligned and can be easily under-

stood and reused by other systems. This mapping process is cost and time-consuming as involves for many hours domain experts to find the right association between local and standard codes. Trying to make easy and speed up this task, many mapping support tools and methodologies had been developed overtime. A lot of attention is focused on the mapping phase, as it is fundamental to foster the standard adoption, likewise mappings double checking is important to verify their accuracy.

The work described in this paper arises from the need to help LOINC Italia experts to manage the great amount of mappings they have to double check to validate them. After the Prime Minister Decree n. 178/2015 entered into force, a lot of public and private laboratories started the process of aligning their local tests to LOINC, often requiring assistance to LOINC Italia for training and validation. This have burdened LOINC Italia experts with a great amount of work, as their validation work follows a bottom-up approach, starting from each local test and the information associated with it, and using them to gradually narrow the research, without looking at the code already identified by the laboratorian, to check then if the LOINC code autonomously identified corresponds to the one chosen by the mapper. This method proved to be advantageous because it allowed to retrace the path carried out by the mapper to identify the most appropriate LOINC code and, therefore, to check for any issues that might have come across. Nonetheless this work is manually performed and hardly quantifiable in terms of time (and consequently costs) as it depends on the completeness of the information available in the local catalogue. If tests are descripted in detail, it is easy to refine the results until arriving to the LOINC code candidate for the mapping. Nevertheless, the reality is that laboratory catalogues are often incomplete and full of acronyms and abbreviations. It requires that experts dedicate more time to fully understand the exact clinical object of the test to determine if it was correctly mapped. Having a tool for automatic recognition of local linguistic forms would be an important step in helping LOINC experts involved in the mappings validation. In this paper we describe the creation of a NooJ grammar system to allow the automatic recognition of Italian local linguistic forms and, starting from them, the detection of specific correctness levels of the mappings between local tests and LOINC codes. By using NooJ contextual constraints we identified thresholds of correctness to determine the level of accuracy of each mapping and decide, on the basis of the assigned score, if the mapping has to be reviewed by LOINC experts or can be considered surely correct.

This paper is structured as follows. Section 2 offers an overview of the LOINC standard and presents a review of the related works; Section 3 describes the NooJ application realized; Sections 4 and 5 are dedicated to present results and conclusions.

2 Background

LOINC codes are uniquely assigned numbers and LOINC names are defined *fully specified* because they contain all the information needed to certainly identify a test, distinguishing it from others who might apparently seem identical. These names are given by the concatenation of six fundamental axis:

1. Component: the substance that is measured (e.g., sodium, glucose, etc.);
2. Property: the measurement type;
3. Timing: it distinguishes measurements made at a given time by those covering a time interval;
4. System: the type of sample on which the observation is performed;
5. Scale: the scale of measurement;
6. Method: the method used in test performing.

The LOINC database has more than 80,000 codes (v. 2.64 - June 2018) and it is translated in 13 languages and 21 linguistic variants. The first LOINC Italian translation was released in 2010 following the part based translational approach [2]. Since then, refinements, based on the analysis of the outcomes of the automatic translation process, produced other releases (the last one in December 2017).

To use LOINC, domain experts are required to map their laboratory tests to the codes of the standard, which convey the same semantic meaning. It is not only an informatics matter, but it requires a deep knowledge of both the destination terminology structure and the way in which the tests are actually realized. This is a non-trivial task because local catalogues contains idiosyncratic names and codes and they are not always complete of all the details required by the standard. Even if some studies [3] propose an incremental approach to mapping, it remains a time consuming and, consequently, cost effective process. Because of this, a lot of supporting tools and techniques had been developed overtime. Regenstrief Institute (the LOINC Standard Development Organization) releases the free program RELMA (REgenstrief LOINC Mapping Assistant), which offers different functionalities to help users in finding the right LOINC code, such as a test frequency rank [4], the Intelligent Mapper [5] and the Community Mappings [6] repository. The goodness of the Intelligent Mapper was tested in a comparative study [7] with a vector space model-based program for mapping local diagnostic radiology terms to LOINC, and in a German study [8] aimed at proving its validity also on non-English languages. Zollo and Huff [9] proposed a mapping approach that aims to map local catalogues to LOINC by cross-referencing the matching codes. They demonstrated that if two local tests match in all the information they carried, and one of them is correctly mapped to LOINC, then this mapping could be inherited by the other. Fidahussein and Vreeman [10] approach is based on supervised machine learning and information retrieval using Apache's Maxent and Lucene to show that the collective knowledge contained in a complete dataset of local terms mapped to LOINC can be used to support mapping new local terms. Lau et al [11] presented a methodology study for automatically mapping local terms to LOINC by using parsing, logic rules, synonyms, attribute relationships and the frequency of mapping to a specific LOINC code. One of the rare study [12] that evaluates mapping correctness, focuses on the analysis of the mappings performed by three large healthcare facilities in order to individuate common errors that occur during mapping operations and define possible mapping improving approaches. Nonetheless it does not address the issue of the time required to double check performed mappings and verify their accuracy before input them into an healthcare system for data exchange.

3 The NooJ application

Following paragraphs describe the development of the NooJ grammar system to support LOINC Italia experts in evaluating the mapping accuracy. It is presented the methodology used for corpus construction and grammars definition and the strategy for automatize the data accuracy control, through contextual constraints.

3.1 The corpus construction

The corpus used as reference for training the system is composed of 8 laboratory test datasets mapped to LOINC. It includes 375 local tests, which were structured on the model of the six main LOINC axes. It was not performed a normalization phase as the intent of the NooJ application we want to develop is properly to recognize local linguistic forms, thus avoiding the request to pre-process laboratory catalogues before the mappings and so relieving LOINC experts from the effort of interpreting local acronyms, abbreviations and conventions. We decided to focus on 33 different types of laboratory tests, so to better control the dictionaries and grammars definition and then eventually extend the developed methodology to the others. We choose some of the tests that usually have in our experience an high level of variability in the name description. We selected all the tests rows belonging to the following categories: *Acido Urico*, *Acido Vanilmandelico*, *Alanina*, *Albumina*, *Azoto*, *Bilirubina*, *Calcio*, *Cloro*, *Colesterolo*, *Creatinina*, *Emoglobina*, *Epinefrina*, *Estradiolo*, *Estrone*, *Follitropina*, *Fosfatasi alcalina*, *Fosfato*, *Glucosio*, *Aspartato transaminasi*, *Alanina transaminasi*, *Fattore di crescita insulino simile*, *Insulina*, *Lattato deidrogenasi*, *Potassio*, *Pregnenolone*, *Progesterone*, *Prolattina*, *Proteina*, *Sodio*, *Trigliceridi*, *Urobilinogeno*, *Vitamina D*. Starting from them, we constructed dictionaries in which the morphological proprieties of local linguistic forms were described and flexional grammars associated to them. Therefore, the corpus is composed by a set of textual data records in which the local test parameters (test name, sample, method, time, scale and type of result) are linked with the mapped LOINC code. Each test is considered such a sentence (See Fig. 1). The order of the elements is predetermined: each sentence has in first position the test name and in the last one the mapped LOINC code. The absence of textual context gave the possibility to apply the formal grammars recognition rules exactly to the correct category, being sure to find it in a specific position. The textual context could, indeed, create a lot of ambiguity especially in specialist domain applications, on the contrary a structured text avoids this problem and makes the results more accurate.

3.2 The NooJ dictionaries and flexional grammars

For recognizing all local linguistic forms in terms of simple lexical units or compound terms, we constructed a set of dictionaries to identify specific terminology used for each clinical test parameter [13]. In particular, in the dictionaries we considered all the local linguistic forms associated to the same LOINC axis. Each lexical unit was described considering its morphologic, syntactic and semantic characteristics as well

as its flexions. The added value given by NooJ is the possibility to associate local linguistic forms and standardized tags from LOINC, by using the semantic properties associated to the lexical units. In particular, for each lexical unit in the dictionaries, we defined a semantic property containing the value of the corresponding LOINC parameter. This allows also the domain experts' knowledge formalization.

The "Analyte.dic" dictionary contains Atomic Lexical Units (ALUs), which are all local linguistic forms placed at the beginning of the sentences and reporting the local idiosyncratic names corresponding to the LOINC Component. Each linguistic form was described using morphologic and semantic proprieties. In particular, the properties "+Analita" and "+CORE" were associated for identifying the specific category and the semantic property "+Component" for assigning the corresponding value for the LOINC Component (See Fig. 2).

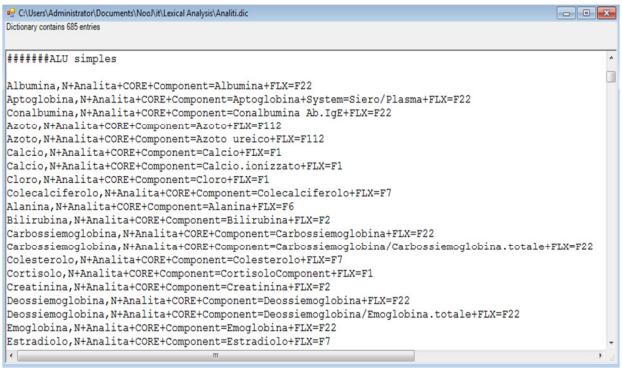

Fig. 1. Example of simple ALUs structure in the Analyte.dic

The definition «Conalbumina,N+Analita+CORE+Component=Conalbumina Ab.IgE» means that the local form *Albumina* is associated to the LOINC Component *Conalbumina Ab.IgE*. Going beyond this simple case, this domain is full of ambiguities, i.e. the definitions «Calcio,N+Analita+CORE+Component=Calcio» and «Calcio,N+Analita+CORE+Component=Calcio.ionizzato» report the same ALU that could be associated to different Component. In this case, there is not a rule that determines if the local form *Calcio* refers to *Calcio* or *Calcio.ionizzato*. Other simple categories such as acids, acronyms and synonyms were represented in the corpus respectively by the properties "+AC", "+Acronimi", and "+SIN". The corpus contains also compound ALUs as multi-words and morphemes. For example, the ALU *Emoglobina A1* refers to the LOINC Component *Emoglobina A1/Emoglobina.totale* reporting a ratio («Emoglobina A1,N+Analita+CORE+Component=Emoglobina A1/Emoglobina.totale »). The same is for the ALUs followed by specific symbols as %, i.e. the ALU *albumina%* refers to the ratio *Albumina/Proteina.totale* («Albumina%,Albumina/Proteina.totale,N+Analita+CORE+Component=Albumina/Proteina.totale»). For the morphemes category, we considered the suffixes *-emia* and *-uria*. These lexical units provide information about other LOINC axis. The suffix *-emia* suggests the value *Blood* (*Sangue* in Italian) of LOINC System and related sub specifications,

while the suffix *-uria* suggests the value *Urine* (*Urina* in Italian). Example of morphemes are: «Glicemia,Glucosio,N+Analita+Component=Glucosio+System=Sangue» and «Azoturia,Azoto,N+Analita+Component=Azoto+System=Urine». The annotation of local morphemes allows also managing ambiguities. In fact the LOINC database contains different sub specifications for *Blood*, often not detailed in the local test definition. For example *-emia* can refer to different kinds of *Blood* (i.e. *Venous blood*, *Arterial blood*, etc.) thus creating ambiguity. In this case, the semantic annotation rules created are able to consider only the element matching with the corresponding element in the associated LOINC code.

The dictionary describes also specific categories such as acids and acronyms. The first are always reported in LOINC definitions with the suffix *-ato*: «Acido metilippurico, Metilippurato, N+Analita+AC+Component=Metilippurato». Acronyms are associated to the official LOINC extended test definitions: «Fsh,Follitropina,N+Analita+Acronimo+Component=Follitropina».

The NooJ dictionaries allow managing the synonyms too. For example, the local form *Emoglobina glicata* is synonym of *Emoglobina A1c* used in LOINC definition, as «Emoglobina glicata,Emoglobina A1c,N+Analita+SIN+Component=Emoglobina A1c».

The "LOINC elements" dictionary contains values of the other LOINC axis. Following the same rules and structure of the Analyte dictionary, it defines the possible values for the parameters System, Scale, Property, Timing and Method. The description of these elements allows formalizing a lot of deductions starting from the values present in the local linguistic form. For example, the local unit of measure could suggest values for Scale, Property and Timing. This kind of formalizations allows deducing values of the LOINC axis even when they are not explicitly reported in local databases. Other important parts of speech considered for this purpose are the adjective in combination with the analyte definition. They are included in this dictionary and associated with the official LOINC axis value it refers to. For example, the adjective *urinario* suggests the value *urine* for the System.

In addition, LOINC codes are formalized by associating all the official elements they represent. This association allows to measure data accuracy and annotate only the correct parts without ambiguities. The realized formal grammars work on local data, transforming them in LOINC language and matching it with standard values associated to a specific LOINC code. The following example reports the formalization of a specific code: «21305-8,NUM+Code+Component=Glucosio+Property=MCnc+Timing=24h+System=Urine +Scale=Qn+Method=Absent». For each ALU we constructed a flexional grammar, according to the specific Syntax used in NooJ. In particular, flexional grammars allow recognizing all the morphologic variations of the lexical units. For example, the multi-words could have different variations (See Table 1).

Table 1. An example of flexional grammars: the multi-words flexion.

Multi-word flexion

<E>/m+s|<B>i<P><B>i/m+p	<E>/m+s|<B>hi<P><B>i/m+p
<E>/m+s|<B>o/m+s|<B>i/m+p	<E>/f+s|<B>e<P><B>e/f+p

3.3 The context-free formal grammars

The context-free grammar is a regular grammar composed of a set of recursive rules used to generate patterns of strings. In this work the structure of hierarchical context-free grammars is described in NooJ to produce the semantic annotation rules sequence [14], [15]. The basic idea concerns the transformation of local terms set used by laboratories to describe their tests into the official LOINC language to control the mapping accuracy. Each local term is stored in a specific variable together with the value of the corresponding semantic property. For example, local test name is stored in the variable "V1" that contains its value and annotates the value of the corresponding semantic property "+Component". The "starting point" is represented by the terms in the local databases and formalized in the NooJ dictionaries. The prediction rules are defined with a specific NooJ syntax establishing which semantic properties values will appear in the final annotation, so the "ending point" is represented by the terms chain recognized by the fixed rules. Table 2 reports the values stored by each variable (with reference to the linguistic category associated in the dictionary), the production rules and the terms chain obtained. For example, in the first row "N+Analita" is a linguistic category described in the dictionary, "+Component=A" is the semantic property associated during the dictionary formalization, "V1$Component" is the production rule for semantic annotation, "A" is the term obtained starting from V1. Row number 3 presents a particular case of production rule as it produces the value *Present* or *Absent* depending on the presence or absence of the value of the Method axis.

Table 2. Semantic annotation variables and annotation rules.

Number	Semantic annotation Variables	Production rules	Language obtained
1	V1=(N+Analita+Component=A)	V1$Component	**A**
2	V2=(N+UnMis+Property=B+Timing=C)	V2$Property V2$Timing	**B+C**
3	V3=(N+Metodo+Method=D)	+Method=Present +Method=Absent	**D**
4	V4=(N+Campione+System=E)	V4$System	**E**
5	V5=(N+Risultato+Scale=F)	V5$Scale	**F**

The final annotated chain will have the form «LOINC+Component=A+Property=B+Timing=C+Method=D+System=E». The following figures (See Fig. 3, Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9) show the complete structure of the realized system in NooJ. The "LOINC_ELEMENTS" grammar is composed of 6 hierarchical sub grammars producing the final terms chain. Each

yellow highlighted element represents the higher level of the related specific hierarchical grammar.

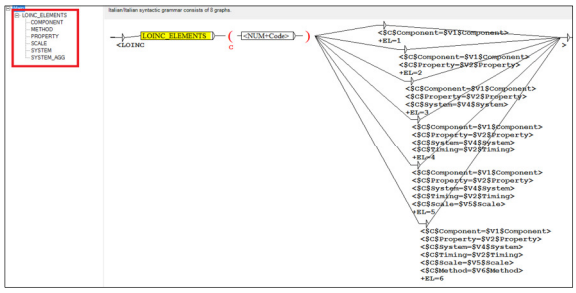

Fig. 2. The LOINC grammar

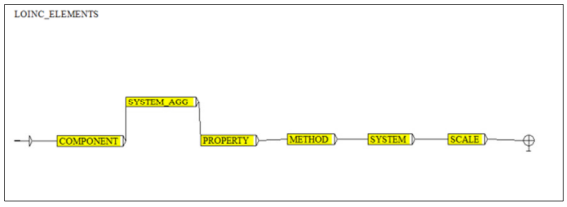

Fig. 3. The "LOINC_elements" grammar

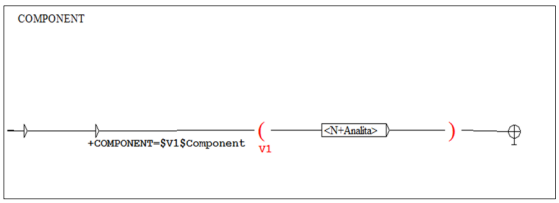

Fig. 4. The "Component" sub grammar

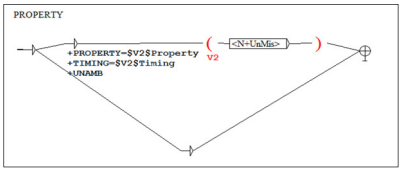

Fig. 5. The "Property" sub grammar

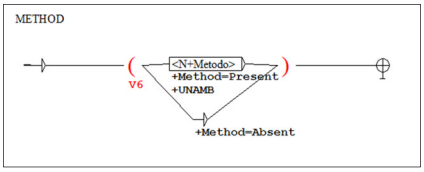

Fig. 6. The "Method" sub grammar

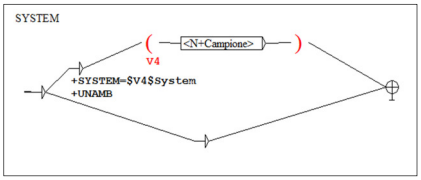

Fig. 7. The "System" sub grammar

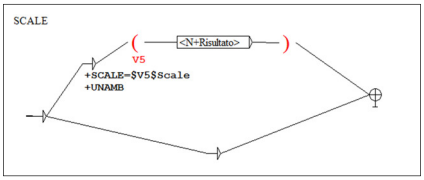

Fig. 8. The "Scale" sub grammar

The figure below (See Fig. 10) shows an example of semantic annotation of a record in the corpus.

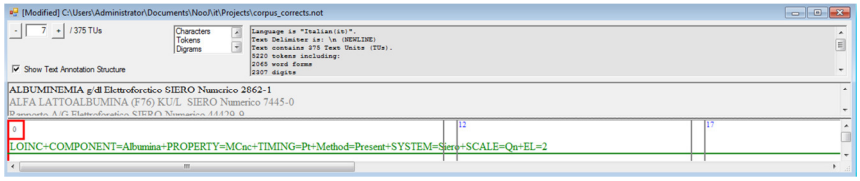

Fig. 10. Example of semantic annotation

3.4 The control of data accuracy: the contextual constraints

The language obtained after the semantic annotation rules application is compared with the values associated to the corresponding semantic properties described by the mapped LOINC code. This process allows identifying and storing the accuracy level of each local test. For example, considering the code description: «555,NUM+Code+**Component=A**+Property=Bb+Timing=Cc+**Method=D+System=E**» and the language obtained after semantic annotation application: «LOINC+**Component=A**+Property=B+Timing=C+**Method=D+System=E**», the contextual constraints produce a specific correctness level, in this case equal to "+EL=3" (matched elements are in bold). The final annotation will be: «LOINC+Component=A+Property=B+Timing=C+Method=D+System=E+EL=3».
For each record, the final part of the formal grammar stores in the variable C the value of the mapped LOINC code recognized in the text and allows comparing the obtained language with the values of the semantic property associated to the specific LOINC code. Table 3 shows the matching rules applied to control data accuracy. They allow to annotate for each test the number of elements that match between the local linguistic forms and the values of the axis of the mapped LOINC code. The semantic annotation is obtained if and only if the matching rules (constraints) are satisfied.

Table 3. The matching rules applied to control data accuracy levels.

Number	Constraints	Semantic annotation
1	CComponent=V1$Component	**+EL=1**
2	CComponent=V1$Component CProperty=V2$Property	**+EL=2**
3	CComponent=V1$Component CProperty=V2$Property CSystem=V4$System	**+EL=3**
4	CComponent=V1$Component CProperty=V2$Property CSystem=V4$System CTiming=V2$Timing	**+EL=4**

5	CComponent=V1$Component CProperty=V2$Property CSystem=V4$System CTiming=V2$Timing CScale=V5$Scale	**+EL=5**
6	CComponent=V1$Component CProperty=V2$Property CSystem=V2$System CTiming=V2$Timing CScale=V5$Scale CMethod=V6$Method	**+EL=6**

Considering for example the local test description «Proteinuria delle 24h mg/24h Urine Numerico 2889-4» and the associated variables:

- V1= Proteinuria,Proteina,N+Analita+Component=Proteina+System=Urine
- V2= mg/24h,N+UnMis+Property=MRat+Timing=24h+Scale=Qn
- V4= Urina,N+Campione+System=Urine
- C=2889-4,NUM+Code+Component=Proteina+Property=MRat+Timing=24h+System=Urine+Scale=Qn+Method=Absent

the obtained semantic annotation will be: *«LOINC+Component=Proteina+Property=MRat+Timing=24h+Method=Absent+System=Urine+Scale=Qn+EL=3»*.

Constraints make direct comparison between local and LOINC official elements to determine their matching. There are also Help constraints which operate on the deduced values, as they allow passing to the higher accuracy level by controlling the values deduced from different parameters. This kind of constraints frequently deals with the adjectives in the test definition, which could suggest values for the System. It allows to double check the correctness of the value associated to the System axis.

4 Results

The system for supporting mapping validation, based on semantic rules set, allows controlling the data accuracy referring to a specific domain. The work presented gives an important contribution in formalizing the experts knowledge in evaluating the associations between the standard linguistic forms used in LOINC and the local linguistic forms used by laboratories. This formalization deals also with the reasoning behind the deduction of specific values, which is proper of the human expertise. The formalization of the considered linguistic elements in NooJ allows having a lot of information about the analyzed datasets. It works as a question-answering system and allows querying the data. It is possible to get different kinds of information such as: “how many times laboratories define their tests only through the test name”, or “how many times they report the information about the used method”. These query operations are possible only through a precise recognition of all the different elements in the local databases. As the local linguistic forms cannot always be used as research strings because they are idiosyncratic, the semantic recognition of local linguistic

forms used by laboratories allows to easily find the subset of codes candidate for mapping. Furthermore, the semantic rules application reduces the set of possible correct codes as research parameters increment. The results obtained after constraints definition allow verifying the entire data trend in terms of accuracy and reducing the analysis time. It is possible to determine the level of correctness for each mapping and so refer back to the laboratories those that need to be improved. Table 4 shows the accuracy level scored by each test of the corpus.

Table 4. The accuracy level scored by each test of the corpus.

Level	Total Number	Stopped at level	Passed to higher level
1	346	146	
2	200	101	17 (level 3 HELP)
3	82	5	1 (level 4 HELP)
4	76		23 (level 5 HELP)
5	53	27	
6	26		

The chance to have an automatic selection of those mappings with an high correctness level (about the 60% out of the total) speeded up the entire validation process.

The realized NooJ application allows also constructing an automatic collection of local linguistic forms stored in a specific XML database and usable for other analysis. The XML file contains both the starting local language and the language obtained after the rules application. The content of each XML LOINC element is represented by the local test description associated to the LOINC code. Each element is associated to the accuracy level reported as the last attribute “EL”.

5 Discussion and conclusions

The work presented in this paper represents an important result in helping the validation process of local tests mapped to LOINC. The realized NooJ grammars allow reducing the time required for codes validation, while preventing potential human errors. As discussed in Section 2, prior studies presented automatic or semi-automatic mapping support tools and techniques. Despite the slightly different application domain, all of them require expert review to double check computer-generated results, while our approach reduces time required for mapping validation by automatically validating all the mappings that gained a top level accuracy score. The developed NooJ application allows to automatically identify one of the main common mapping errors, also described in [12]: a more specific test mapped to a more general concept. In fact, thanks to the use of the created dictionaries and grammars, it annotates all the elements of the local name that have to find correspondence into the LOINC name to be considered as correct. Differently from the cited mapping accuracy study, our system does not perform a validation based only on the names, but it automatically considers all the LOINC axis. Even if semantic annotation is not new in automated or semi-automated tools for language analysis and comparison in the LOINC mapping efforts, this is the first time it is applied to the mapping accuracy validation. Actually,

for what is in our knowledge, the same LOINC producers do not have a system to control the clinical quality of the performed mapping around the world. Everything is left to the mappers competence or, as in Italy, to local organizations responsible for the standard implementation. The system presented is a first attempt to create a support tool for speeding up the mapping validation process, which is fundamental to avoid that erroneous mappings vehicle erroneous clinical concepts thus invalidating the semantic interoperability effort. In the future, the work could be extended to support laboratorians in mapping operations, but it would need some more adjustments of the defined rules as this activity is performed without a precise one to one comparison, as in mapping accuracy control, but having all the LOINC codes as potential reference. This system would be then helpful for both laboratorians involved in the mapping operations and LOINC experts for mappings validation.

References

1. McDonald, C.J., Huff, S.M., Suico, J.G., *et al*: LOINC, a universal standard for identifying laboratory observations: a 5-year update. Clinical Chemistry 49, 624–633 (2003).
2. Vreeman, D.J., Chiaravalloti, M.T., Hook, J., McDonald C.J.: Enabling international adoption of LOINC through translation. Journal of Biomedical Informatics 45(4), 667–673 (2012).
3. Vreeman D.J., Finnell J.T., Overhage, J.M.: A rationale for parsimonious laboratory term mapping by frequency. In: AMIA Annual Symposium Proceedings, pp. 771–775, 2007.
4. Regenstrief Institute, Inc., Common LOINC Laboratory Observation Codes, http://loinc.org/usage/obs, last accessed 2018/09/15.
5. Vreeman, D.J., McDonald, C.J.: Automated mapping of local radiology terms to LOINC. In: AMIA Annual Symposium Proceedings, pp. 769–773, 2005.
6. Dixon, B.E., Hook, J., Vreeman, D.J.: Learning from the crowd in terminology mapping: the LOINC experience. Lab Medicine 46(2), 168–174 (2015).
7. Vreeman, D.J., McDonald, C.J.: A comparison of Intelligent Mapper and document similarity scores for mapping local radiology terms to LOINC. In: AMIA Annual Symposium Proceedings, pp. 809–813, 2006.
8. Zunner, C., Bürkle, T., Prokosch H-U., Ganslandt, T.: Mapping local laboratory interface terms to LOINC at a German university hospital using RELMA V.5: a semi-automated approach. Journal of American Medical Informatics Association 20(2), 293–297 (2013).
9. Zollo, K.A., Huff, S.M.: Automated mapping of observation codes using extensional definitions. Journal of the American Medical Informatics Association 7(6), 586–592 (2000).
10. Fidahussein M., Vreeman D.J.:A corpus-based approach for automated LOINC mapping. Journal of the American Medical Informatics Association 21(1), 64–72 (2014).
11. Lau, L.M., Johnson, K., Monson, K., Lam, S.H., Huff, S.M: A method for the automated mapping of laboratory results to LOINC. In: AMIA Annual Symposium Proceedings, pp. 472–476, 2000.
12. Lin, M-C., Vreeman, D.J., McDonald, C.J., Huff, S.M.: Correctness of voluntary LOINC mapping for laboratory tests in three large institutions. In: AMIA Annual Symposium Proceedings, pp. 447–451, 2010.
13. Silberztein, M.: La formalisation des langues, l'approche de NooJ. ISTE, London (2015).
14. Chomsky, N.: Structures syntaxiques. Le seuil, Paris (1957).
15. Husser, R.: Foundations of Computational Linguistics. Springer, London (2014).