Improving Medical Coding with Case-Based Reasoning — Evaluation for Cancer Registries*

Michael Schnell[1,2][0000−0002−5728−2358], Sophie Couffignal[1], Jean Lieber[2][0000−0002−5547−6466], and Nicolas Jay[2,3][0000−0002−5790−6933]

[1] Department of Population Health, Luxembourg Institute of Health, 1A-B, rue Thomas Edison, L-1445 Strassen, Luxembourg, firstname.lastname@lih.lu
[2] UL, CNRS, Inria, Loria, F-54000 Nancy, firstname.lastname@loria.fr
[3] Service d'évaluation et d'information médicales, Centre Hospitalier Régional Universitaire de Nancy, Nancy, France, n.jay@chru-nancy.fr

Abstract. Providing a comparable and consistent description of any entity is a difficult task. Vocabulary, semantics and objectives need to be very clearly defined and followed. The same can be said about cancer registries. They are an essential element in the fight against cancer. Among the main tasks of these registries is the data collection and coding process of cancer cases. To ensure comparable and consistent data, complex international standards and numerous best coding practices have been defined. Unfortunately this complexity can easily overwhelm operators, which are the people in charge of data collection and coding. While coding experts can help operators in their job, this represents a great burden on their precious time. To assist operators in their task and reduce the burden on coding experts, a coding assistant relying on arguments was designed and implemented. This system provides answers and a partial explanation, using arguments in favor and against answers. In this paper, a first evaluation of this system is presented, testing the system on real topography questions asked by operators.

Keywords: interpretation of best practices · interpretive case-based reasoning · coding standards · cancer registries · user assistance · decision support.

1 Introduction and Context

The world is a complex entity. People have ever since tried to quantify it, describing it using carefully selected features and metrics. This gargantuan task presents many challenges, which we also find in medical coding. The latter is needed when it comes to understanding and evaluating the public health of the

* Supported by the Fondation Cancer (http://www.cancer.lu). The authors wish to thank the anonymous reviewers for their helpful comments.

overall population, but also for health care of individual patients. In order to have comparable and high quality data, international standards have been developed (e.g International Classification of Diseases, 10^{th} edition or Systematized Nomenclature of Medicine (SNOMED)). These standards define the codes used to describe medical data and how to select the correct coding given a patient's medical data. Despite complex and extensive coding standards, there will always be cases that simply do not fit in the described situations and coding decisions. In those situations, coding experts need to make decisions, based on their knowledge and understanding.

This work is done in the context of the Luxembourg National Cancer Registry (NCR). The goal of the NCR is to assess the incidence and the treatment of cancer in Luxembourg. To achieve this objective, data on all cancer cases diagnosed and/or treated in Luxembourg is collected. Among the retrieved information, we find the cancer type, which is mainly defined by a topography and a morphology. The topography of a tumor is the location where the tumor originated, i.e. where the first tumor cells developed. The morphology of a tumor describes the tumor cell type and behavior (e.g. aggressiveness).

As an illustrating example of the topography coding of a tumor, let us consider the following case. An operator is confronted with a tumor that spans between the middle lung lobe and the pulmonary pleura. The pulmonary pleurae are membranes that envelop the lungs. In the patient record, the operator finds one imaging that describes the tumor as originating from the pulmonary pleura and then invading the lung tissue. Later that tumor was surgically removed and the surgery report confirms this finding, i.e. a tumor originating in the pulmonary pleura. The removed tissue was sent to a laboratory for histological analysis. In their report, the pathologist contradicts the presumed origin, stating that given the tumor cell type, this tumor actually originated in the lung tissue and later spread to the pulmonary pleura. For the treatment of this patient, the exact origin of the tumor is not relevant and any later mention of the tumor just references it as a lung cancer, where lung designates the general type of cancer, but not the exact origin. For both origins, we talk about "lung" cancer. The NCR uses the ICD-O 3^{rd} edition [11] to describe tumor topography. For our example, two codes are considered, `C34.2` for middle lung lobe and `C38.4` for pulmonary pleura. For an operator this is a difficult choice, as it relies on both medical knowledge and coding knowledge to determine which opinion should be preferred. In this situation, coding experts decided to follow the pathologist's point of view, given his very strong argumentation.

In this situation, the information provided was contradictory, but clear. In other situations, the provided information might be more vague or simply missing. In those situations, operators cannot easily decide what to code and have to rely heavily on coding experts. For the NCR, operators can ask questions when faced with difficult cases, but this process takes up a lot of time for the coding experts of the NCR. This project aims at designing a method to reduce the burden on coding experts and to facilitate coding.

In this paper, we summarize and explain the designed question solving method. We then present the evaluation of the coding assistance method, before concluding and highlighting possible future work.

2 Coding assistant

To tackle the issue of coding, we developed a coding assistant using case-based reasoning [10]. This choice was done after carefully analyzing the reasoning process of the coding experts when confronted with coding questions from operators. The designed solving method was implemented in a web portal, intended to centralize communications between operators (asking questions) and coding experts (answering questions). Figure 1 provides an overview of the interactions with our system. Figure 2 shows an example of a case viewed in the implemented system.

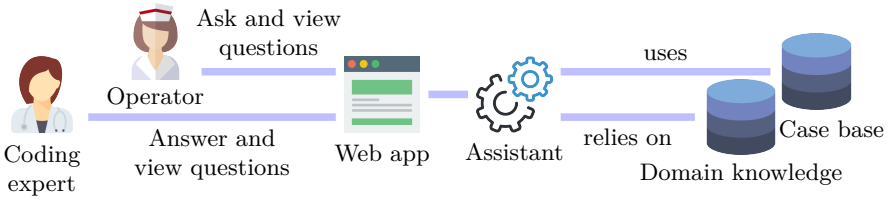

Fig. 1: Application overview and interactions.

In the following paragraphs, we will introduce some of the techniques and methods used for our system, before summarizing the argument-based retrieval design for our project.

The Semantic Web [3] is an extension of the World Wide Web, aiming at facilitating the access and reuse of all available information. The World Wide Web has grown immensely since its creation, yet most of the information is only accessible in unstructured textual form. To make this information machine-usable, a new knowledge representation language was defined, namely Resource Description Framework (RDF) [8]. Data is represented using triples (`subject predicate object`) that can be seen as a sentence (`predicate` as the verb). RDF Schema (RDFS) [4] is an extension of RDF for data-modeling. An RDFS base is a set of triples and can be assimilated to a graph, where nodes are subjects or objects and edges are labeled with predicates. The RDFS base shown in figure 3 partially describes the illustrating example from the introduction and expresses that John has an imaging with one finding of a tumoral lesion in the pulmonary pleura.

SPARQL (SPARQL Protocol and RDF Query Language) [6] is a query language used to retrieve and manipulate RDF (or RDFS) data stores. There are several types of queries, some to manage the underlying data (insert, update, delete) and others to retrieve it (select, construct, ask). `ASK` queries are

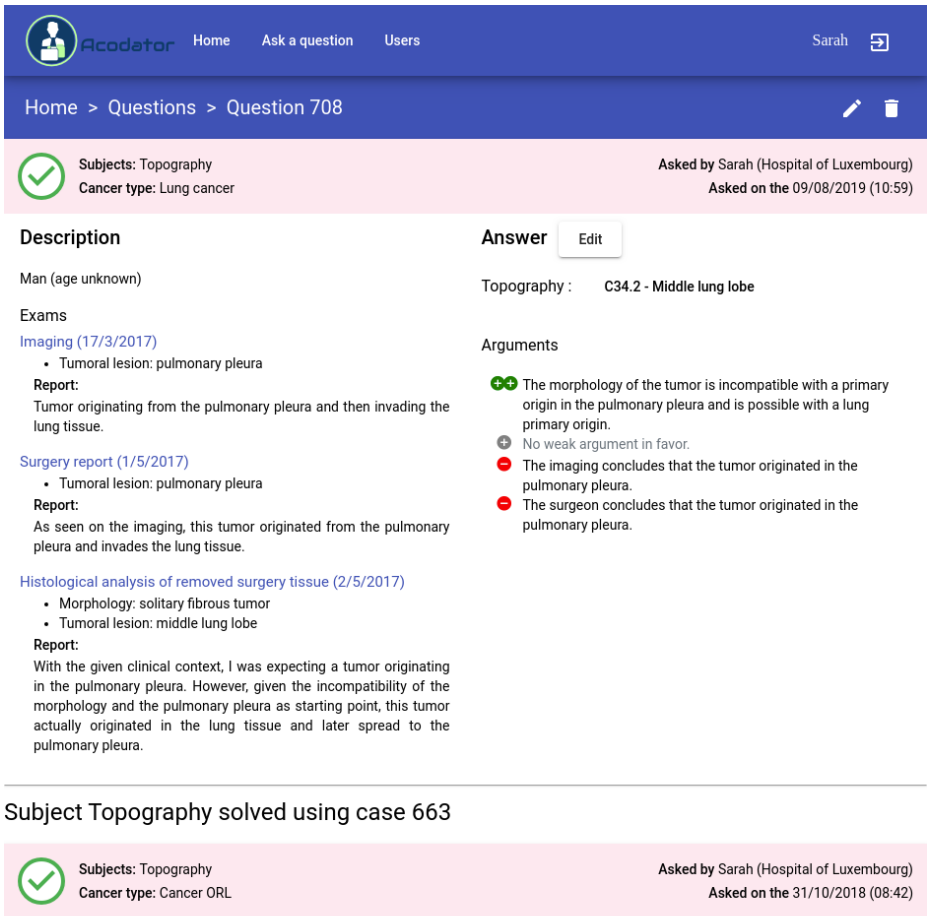

Fig. 2: Summary of the described problem and the proposed solution. The most appropriate source cases are shown similarly to the target problem (partially visible on this screenshot).

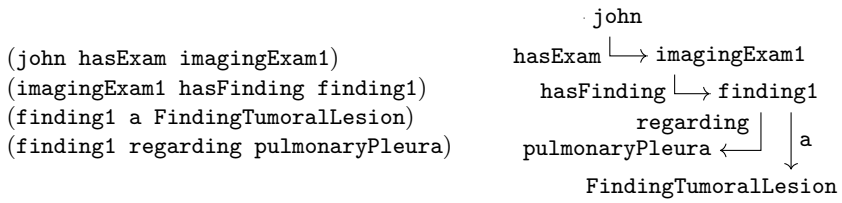

Fig. 3: RDFS base example (left) and matching graph (right) describing the imaging found in the illustrating example in the introduction.

used to check if a given pattern can be found in the underlying data store. `ASK {?person hasExam ?exam}` is a valid query, where `?person` and `?exam` are variables (name starting with a `?`), that can match any subject, property or object. Given the RDFS base shown in figure 3, the previous query will evaluate to true, as there is a set of triples (e.g. {(`john hasExam imagingExam1`)}) that matches the described pattern.

Case based-reasoning is a problem solving method where, for a given domain, previously solved problems are used to solve new problems. A case is defined as a problem-solving episode, typically represented by a pair (`pb`, `sol(pb)`), where `pb` is a problem from the given application domain and `sol(pb)` is a solution of `pb`. The new problem to solve is called target problem, denoted by `tgt`. Using the description of `tgt` and domain knowledge, a suitable case from the case base, i.e. the set of all previously solved cases, is identified. This case is called source case, denoted by (`srce`, `sol(srce)`). Then `sol(srce)` together with domain knowledge are used to solve the target problem. This new case (`tgt`, `sol(tgt)`) is then revised, e.g. to correct the solution or update the problem description. This revised case (`tgt'`, `sol(tgt')`) is then added to the case base, enabling the system to potentially solve new problems.

Our method, summarized in figure 4, uses case-based reasoning and a 4-R cycle [1] and two knowledge containers [9]. The domain knowledge container consists mainly of medical knowledge (e.g. hierarchical definition of body parts, exam types. etc.).

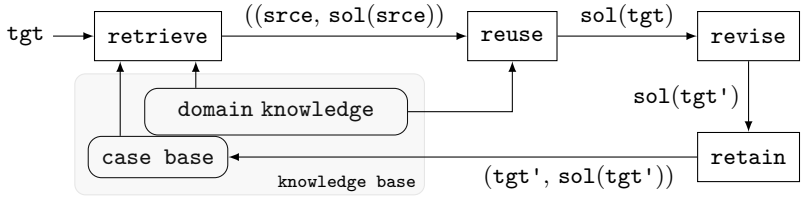

Fig. 4: Adapted 4-R cycle and knowledge containers of our approach.

For this project, the application domain is medical coding for a cancer registry. A problem is defined as a question, a patient record and a reference for the applied coding standards. The question consists of a subject, i.e. the variable or information to code (e.g. incidence date, topography), and a general cancer type, e.g. breast cancer, prostate cancer, lung cancer, etc. The patient record consists mainly of a summary description of the medical exams (e.g. date of exam, findings, etc.) and main history of the patient (e.g. cancer antecedents). Figure 5 shows a partial RDFS graph for the illustrating example. The coding standard references are used to track which version was followed for this case (e.g TNM version 7 or TNM version 8). A solution of a problem consists of an answer for the given question and arguments explaining the given answer. An argument is a piece of domain knowledge which is used by a coding expert to support or attack

a specific answer. We distinguish between three types of arguments, strong pro, weak pro and weak con. A strong pro is an argument in favor that leaves no doubt for the given answer. A weak pro is also an argument in favor, however by itself it is not sufficient evidence to conclude. Indeed there might be situations where despite this argument, a different answer is chosen. Similarly, a weak con is an argument against, but it is not sufficient to exclude a given answer.

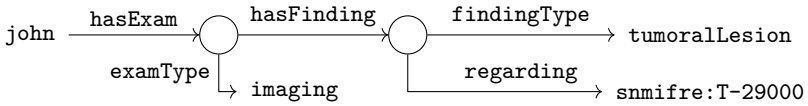

Fig. 5: Partial RDFS graph for the illustrating example, describing the imaging findings. `snmifre:T-29000` describes the pulmonary pleura in SNOMED.

Let's reconsider the example described in the introduction. For this problem, the subject is the topography, the cancer type is lung cancer, and the patient record description contains the imaging findings, the surgery report and the histological surgery report. The solution of this problem contains the answer to the topography question, i.e. C34.2 (middle lung lobe), and arguments for this answer. For this problem, there are two weak cons and one strong pro. The latter states *The morphology of the tumor is incompatible with a primary origin in the pulmonary pleura and is possible with a lung primary origin.* The weak cons state *The surgeon concludes that the tumor originated in the pulmonary pleura*, and *The imaging concludes that the tumor originated in the pulmonary pleura.*

To compare cases, we propose a method using arguments. In the retrieve step, we look for the most suited source case among all other cases in the case base which have the same subject as the target problem. These cases are ranked using three criteria. The first two criteria take the applicable arguments into account and the last, default criterion compares the patient records. An argument is considered to be applicable for a given problem if it is considered true for the described patient. Formally, an argument `arg` is a function that takes a case and returns a boolean. If the argument is applicable, then `arg(case) = TRUE`. Arguments are formalized using SPARQL `ASK` queries.

The first comparison criterion, denoted by $\mathcal{C}_{strong}$, takes only strong pros into account. Given two source cases, the source case with the most applicable strong pros for the target problem is preferred. The second criterion, denoted by $\mathcal{C}_{weak}$, uses only weak arguments (both pro and con). Given two source cases, the case with the most applicable weak pros and the least weak cons for the target problem is preferred. The last criterion, denoted by $\mathcal{C}_{dist}$, relies only on the patient records. Given two source cases, the case with the patient record that is closest to the patient record of the target problem is preferred. The distance between the patient records is computed using a graph edit distance [5].

In order to find the most suitable source case, all the cases forming the case base are compared to each with regards to the target problem using the three criteria mentioned above, in a lexicographical way, that is first $\mathcal{C}_{strong}$, then $\mathcal{C}_{weak}$ and finally $\mathcal{C}_{dist}$. Formally, to solve a new problem `tgt`, the source case will be the case `srce` such that for all $\texttt{case} \in \texttt{CaseBase}, \texttt{srce} \preccurlyeq_{tgt} \texttt{case}$, where $\preccurlyeq_{tgt}$ is a preorder such that, given two source cases $\texttt{srce}_i$ and $\texttt{srce}_j$, $\texttt{srce}_i \preccurlyeq_{tgt} \texttt{srce}_j$ if

$$\Delta^s_{i,j} > 0 \texttt{ or } (\Delta^s_{i,j} = 0 \texttt{ and } (\Delta^w_{i,j} > 0 \texttt{ or } (\Delta^w_{i,j} = 0 \texttt{ and } \Delta^d_{i,j} \geq 0)))$$

where $\Delta^s_{i,j}$ is defined as

$$\Delta^s_{i,j} = \mathcal{N}^{\texttt{sp}}(\texttt{srce}_i, \texttt{tgt}) - \mathcal{N}^{\texttt{sp}}(\texttt{srce}_j, \texttt{tgt})$$

and $\Delta^w_{i,j}$ is defined as

$$\begin{aligned}\Delta^w_{i,j} = \quad & \lambda_{\texttt{p}} \cdot (\mathcal{N}^{\texttt{wp}}(\texttt{srce}_{\texttt{i}}, \texttt{tgt}) - \mathcal{N}^{\texttt{wp}}(\texttt{srce}_{\texttt{j}}, \texttt{tgt})) \\ & - \lambda_{\texttt{c}} \cdot (\mathcal{N}^{\texttt{wc}}(\texttt{srce}_{\texttt{i}}, \texttt{tgt}) - \mathcal{N}^{\texttt{wc}}(\texttt{srce}_{\texttt{j}}, \texttt{tgt}))\end{aligned}$$

and λ_p and λ_c are two non-negative coefficients currently fixed to λ_p `= 3` and λ_c `= 2` and $\Delta^d_{i,j}$ is defined as

$$\Delta^d_{i,j} = \texttt{dist}(\texttt{srce}_j, \texttt{tgt}) - \texttt{dist}(\texttt{srce}_i, \texttt{tgt})$$

and $\mathcal{N}^{\texttt{args}}(\texttt{srce}_s, \texttt{tgt})$ denotes the number of arguments of type `args` of the source case $\texttt{srce}_s$ which are applicable for the problem `tgt` and is defined as

$$\mathcal{N}^{\texttt{a}}(\texttt{srce}_s, \texttt{t}) = |\{\texttt{a} \in \texttt{args}(\texttt{srce}_s) \mid \texttt{a}(\texttt{tgt}) = \texttt{TRUE}\}|$$

and $\texttt{args} \in \{\texttt{sp}, \texttt{wp}, \texttt{wc}\}$ is function which returns all arguments of a case of a given type (`sp` for strong pros, `wp` for weak pros and `wc` for weak cons) and `dist` is a function where `dist(x, y)` returns the graph edit distance between `x` and `y`.

In the reuse step, we compute the solution for the target problem. The answer is copied from the solution of the found source case. For arguments, only those from the source case which are applicable for the target problem are copied.

In the revise step, a coding expert can review the target problem `tgt` and the proposed solution `sol(tgt)`. If necessary, they will update the problem description, e.g. to remove unnecessary information, and/or correct the provided answer and/or arguments. In the retain step, coding experts decide if the new case (`tgt'`, `sol(tgt')`) is interesting for future use and thus should be inserted into the case base. Both revise and retain steps are currently manual and required for each new problem for the initial system use, but could be partially automated in the future.

3 Evaluation

The described method was developed to assist operators in the coding process. This first evaluation was designed to assess the strength of this approach on real cases. In a first step, it will only focus on comparing provided solutions with expected ones, to give us insights into the effectiveness of our approach and the quality of our evaluation set.

3.1 Dataset

The dataset consists of questions asked in the context of the NCR. We only considered questions about topography, as this subject is one of the recurring subjects and the amount of possible answers is reasonable, with only about three hundred possible values against over a thousand for morphology values. A topography code defines a location in the human body. It is composed of three digits, the first two define a global category and the third digit specifies a subcategory. For example, the code `C34.2` represents the middle long lube, from the category of bronchus and lung locations (`C34`). The subcategories `8` and `9` have a special meaning. Subcategory `8` means that multiple subcategories are concerned and that none can be specifically chosen. Subcategory `9` indicates that no information is available to use a more precise subcategory. Thus `C34.8` represents an overlapping region of bronchus and lung and `C34.9` represents the overall region, without more details.

The dataset consists of 37 cases and their solution. The cases were collected with the invaluable help of the coding experts of the NCR, which provided the necessary explanations. Out of the 333 existing topography codes, 27 were present in our dataset and 6 were used in at least two solutions. Out of the 70 provided arguments, 61 have been formalized into matching SPARQL ASK requests. Table 1 highlights how arguments were used in our dataset.

	Strong pro	**Weak pro**	**Weak con**	**All**
Total number	8	55	11	76
Average per case	0.2	1.5	0.3	2.0
Range per case	0–1	0–4	0–3	0–5

Table 1: Argument use in the evaluation set by argument type.

3.2 Experiments and Indicators

In our evaluation, both answer quality and explanation (i.e. argument) quality were evaluated using two experiments.

Firstly, we attempted to solve each case using the whole dataset (without removing the case to solve from the case base). This test was meant to evaluate if an identical new problem could be solved correctly. For this experiment, we had a dataset consisting of 37 cases.

Secondly, we used a leave-one-out cross-validation, i.e. we tried to solve each case from our dataset using all other cases. For this experiment, we only kept cases for which there was another case with the same answer (topography code), resulting in an evaluation set of 16 cases. This exclusion is linked to the copy reuse method, where a new target problem is given the same answer as the

selected source case. Thus if there is no source case with the expected answer, the target problem cannot be correctly answered.

To determine answer quality, we counted the number of correct solutions, for each experiment. For cases which failed to be resolved, we assessed if a source case leading to a correct solution was present in the top five of closest cases. To assess argument quality, we compared the suggested arguments to the expected ones. For additional arguments, coding experts were consulted to evaluate if those were relevant in these situations.

3.3 Results and Interpretation

For our first experiment, 35 out of 37 cases were correctly identified and solved. For the 2 remaining ones, the case was present in the top five of closest cases, but failed to reach the top due to missing formal arguments. There were other arguments which could be applied and thus these source cases were preferred.

In our second experiment, 10 out of 16 cases were correctly answered. There was 1 additional case for which a source case with the expected answer was in the top five closest answers. For the 5 remaining cases, the main cause for failure was the limited amount of arguments, with on average only 2 arguments per case. Some of the arguments might also have been too specific to solve these cases.

We also evaluated the provided arguments for each solved case. Only 3 out of 16 cases had exactly the same arguments in their new solution. For the other cases, there were always fewer arguments in the new solution compared to the one provided by coding experts. This difference is partially due to missing formalization of more complex arguments and partially to the sparse reuse of arguments in the cases of the evaluation set. Indeed, most arguments were only used in one case.

There are several possibilities to improve the performance of our approach. Adding more cases and arguments is the most straightforward at this stage.

To increase the range of possible answers, a more complex reuse method and more sophisticated arguments could be used. Take for example two cases, $\texttt{srce}_1$ with answer `C34.2` (middle lung lobe) and a weak pro stating *An imaging concludes that there is a tumoral lesion in the middle lung lobe.*, and $\texttt{srce}_2$ with answer `C18.1` (appendix) and a weak pro stating *A CT scan concludes that there is a tumoral lesion in the appendix.*. Both arguments could be generalized into a single argument stating *An* `?imaging` *concludes that there is a tumoral lesion in the* `?location`., where `?imaging` would match any imaging exam types (CT scans, PET scan, etc.) and `?location` would match a location which is known to be coded with the topography code used in the case solution. While solving a new problem, when applicable, this argument could be used to provide new topography codes which have not yet previously been used in a source case. The mapping between the location and the topography codes is an example of domain knowledge needed to improve the system. For example, given a new problem with an exam indicating a tumoral lesion in the liver and knowing that

the liver is coded with the topography code `C22.0`, we could answer this case even if no source case exists in our case base with this answer.

Most other applications in medical coding focus on automatic coding, with little effort into explaining the generated codes [7]. There are other works in the area of argumentation, though they use cases as precedents [2] or focus on the interaction between arguments (arguments defeating each other).

4 Conclusion

This paper presented a first evaluation for the developed medical coding assistant. More cases and more subjects should allow for a more comprehensive evaluation. This evaluation focused mainly on the answering method, but in a second phase user acceptability for the coding assistant should also be assessed. The implemented tool has been launched in a pilot phase in the context of the NCR. A user survey to understand if users understand the solutions and explanations provided could provide insights into acceptability and trust. It would also be of interest to test this method in a different domain, to assess the generality of this approach.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI communications **7**(1), 39–59 (1994)
2. Aleven, V., Ashley, K.D.: Teaching case-based argumentation through a model and examples empirical evaluation of an intelligent learning environment. In: Artificial intelligence in education. vol. 39, pp. 87–94 (1997)
3. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific american **284**(5), 28–37 (2001)
4. Brickley, D., Guha, R.V.: RDF Schema 1.1, https://www.w3.org/TR/rdf-schema/, W3C recommendation, last consultation: June 2019 (2014)
5. Bunke, H., Messmer, B.T.: Similarity measures for structured representations. In: European Workshop on Case-Based Reasoning. pp. 106–118. Springer (1993)
6. Group, W.S.W.: SPARQL 1.1, https://www.w3.org/TR/sparql11-overview/, W3C recommendation, last consultation: June 2019 (2013)
7. Kavuluru, R., Hands, I., Durbin, E.B., Witt, L.: Automatic Extraction of ICD-O-3 Primary Sites from Cancer Pathology Reports. AMIA Summits on Translational Science Proceedings **2013**, 112–116 (2013), http://www.ncbi.nlm.nih.gov/pmc/papers/PMC3845766/
8. Klyne, G., Carroll, J.J., McBride, B.: RDF 1.1, https://www.w3.org/TR/rdf11-concepts/, W3C recommendation, last consultation: June 2019 (2014)
9. Richter, M.M., Weber, R.O.: Case-based reasoning: a textbook. Springer Science & Business Media (2013)
10. Schnell, M., Couffignal, S., Lieber, J., Saleh, S., Jay, N.: Case-Based Interpretation of Best Medical Coding Practices — Application to Data Collection for Cancer Registries. In: Conference Proceedings of ICCBR (2017), http://dx.doi.org/10.1007/978-3-319-61030-6_24
11. World Health Organisation: International classification of diseases for oncology (ICD-O) – 3rd edition (2013), http://www.who.int/iris/handle/10665/96612